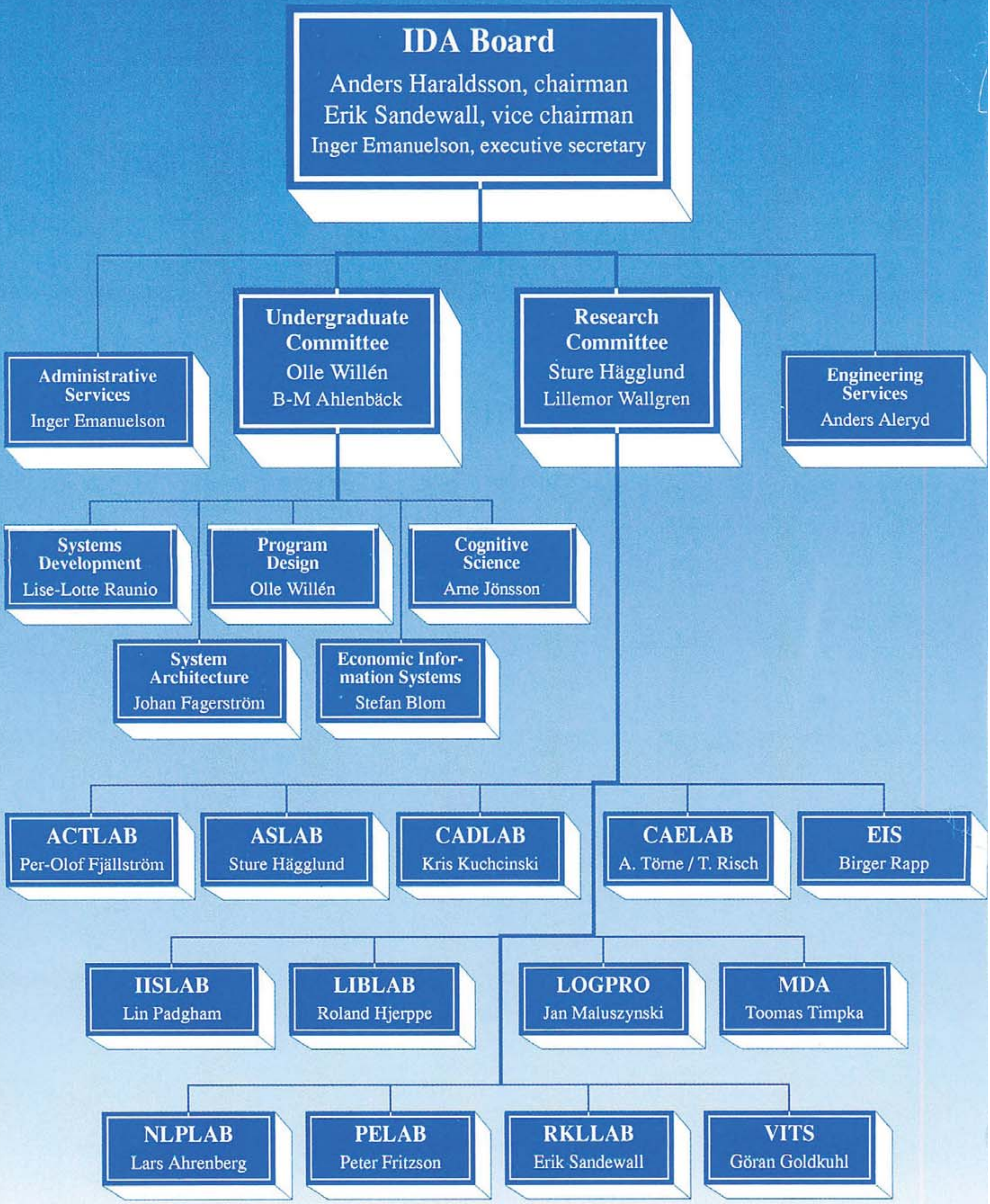
LINKÖPING
UNIVERSITY AND INSTITUTE OF TECHNOLOGY

Department of Computer and Information Science

ACTIVITY REPORT

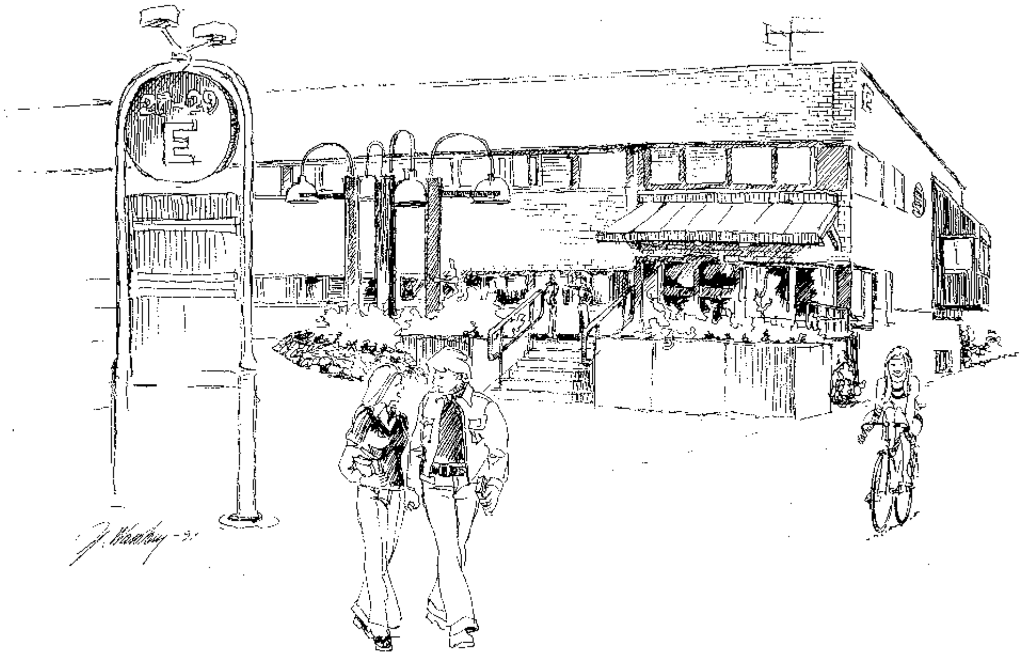
1991-92

The Department of Computer and Information Science



Department of Computer and Information Science

Activity Report 1991–92



Department of Computer and Information Science, main entrance.

ISBN 91-7871-066-9
ISSN 0281-4250
Printed in Sweden by LJ Foto & Montage AB, Linköping/Norrköping Tryckeri, Norrköping 1993

Acknowledgements

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

Figure. *The Editorial Board.*
From the left, Tommy Olsson, Lillemor Wallgren (chief editor), Lisbeth Linge, Björn Nilsson, and Sture Hägglund.

Figure. *The university campus with the E building to the left.*

Contents

Appendices

1 Introduction and Overview

The Department of Computer and Information Science (IDA) at Linköping University was formed 1983 as an independent department. Since then it has grown to the largest department in its area in Sweden with several internationally well-known and recognized groups in a variety of areas in the computer and information field. Several educational programmes with a large number of courses have been developed for undergraduate and graduate education, new research laboratories have been established, and an effective infrastructure in terms of computer systems, system support staff, and organization in general has been created.

Today IDA has about 170 employees, with a staff of teachers and researchers of about 60 persons (more than 50% Ph. D.'s) and administrative and technical staff of about 30 persons. The remaining employees are full-time graduate students. There are six full professors at the department. IDA has an annual budget of 65 MSEK ($9 million USD). About 35% of the money and effort are for undergraduate education, 5% for continuing education and technology transfer activities, and 60% for research. The computer facilities for IDA's employees and its students is a distributed network including more than 250 SUN workstations, an IBM AS/400, HP workstations, and about 140 Macintoshes and PC's.

At undergraduate level IDA gives courses in the subject areas *computer science* (datalogi), *computer systems* (datorsystem), *administrative data processing* (administrativ databehandling) and *economic information system*s (ekonomiska informationssystem) in both the School of Engineering (tekniska högskolan) and the School of Arts and Sciences (filosofiska fakulteten). We deliver more than about 130 undergraduate courses with about 7700 participants corresponding to a volume of 600 full-time students.

IDA has graduate programmes in the subject areas *computer science* (datalogi), *computer systems* (datorsystem), *library and information sciences* (informatik), *computational linguistics* (datorlingvistik), *engineering information systems* (tekniska informationssystem) and *economic information system*s (ekonomiska informationssystem). During the years 1991 and 1992 a total of 13 Ph. D.'s and 11 licentiate theses were defended from these programmes.

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 founders. Currently there are thirteen research laboratories. A lab is characterized by its long term commitment to development and maintenance of knowledge within a defined area, and by its long term responsibility for graduate students.

The current research laboratories are:

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

The most significant recent change in the research organization is the establishment of the VITS group as a permanent research laboratory.

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

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

1.1 IDA's organization in general

IDA's general organization is described in more detail in Appendix A. The department is led by a *Department Board* (institutionsstyrelse). From July 1, 1990, Anders Haraldsson has been Department Chairman (prefekt) and Erik Sandewall has been Vice Chairman (proprefekt).

The two main areas of activity are reflected in two subordinate committees.

- the *Undergraduate Teaching Committee* (undervisningsnämnden, IDUN), headed by Olle Willén;
- the *Research Committee* (forskningsnämnden), headed by Sture Hägglund.

The members of the Department Board are IDA personnel, elected from the staff of the department, and undergraduate students appointed by student interest organizations. Committee members are appointed by the Department Board and the Research Committee equals approximately the set of laboratory leaders.

The organizational groups within the department are:

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

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

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

1.2 Research background

Our research is located in the University's School of Engineering (Tekniska högskolan). The School of Arts and Sciences organizes its research and graduate education programmes in the Themes research organization, and normally not on the subject level. For that reason, IDA's research and graduate programmes are all organized in the School of Engineering.

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

Our main sources of research grants are from the Swedish Research Council for Engineering Sciences (TFR) and the Swedish National Board for Industrial and Technical Development (NUTEK). During the 80's the Swedish Board of Technical Development (STU) had 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 sponsors that research results should be transferred to applications in industry, commercial users of computer systems, public administration, or in other areas of research outside computer science. These goals are sometimes competing, and possibly even contradictory. In IDA we have tried to balance our efforts so that both the basic research goal's and the applied goal's should be achieved reasonably well. We also recognize the importance of continuous interaction between basic and applied research in our field.

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

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

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

mathematical logic for knowledge representation
logic programming
complexity theory
library science
information systems
human-computer interaction
industrial information technology
economic information systems
knowledge engineering
natural language processing
cognitive science
data bases

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 summarized below and described in more detail in subsequent chapters, attempts to make a reasonable trade-off between concentration and breadth.

1.3 Research organization

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

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

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

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

The recruiting situation is 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

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

ACTLAB 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.

ASLAB – Application Systems Laboratory

Sture Hägglund

ASLAB 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.

Associated with ASLAB is the *Industrial Software Technology project,* studying software engineering issues in an industrial context.

CADLAB – Laboratory for Computer-Aided Design of Digital Systems

Krzysztof Kuchcinski

CADLAB 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, Tore Risch

CAELAB conducts research regarding information technology to be used in industrial processes and products, in particular computer support for automation and for engineering data bases. The work is centered around two implementation frameworks - AMOS, Active Mediator Object System and ARAMIS, A Robotics and Manufacturing Instruction System. Application projects in cooperation with Swedish industry are also being done.

EIS – Economic Information Systems

Birger Rapp

EIS covers communication of information from people to/from systems or between people and the design of information systems supporting this communication. Research projects concern information support, agency theory, IT and organizational solutions, computer simulation for management training and decision support, business control and accounting and auditing.

IISLAB – Laboratory for Intelligent Information Systems

Lin Padgham

IISLAB studies theory and methods for advanced information systems, including object-orientation, taxonomic reasoning, composite objects and temporal issues. In a major implementation project a multi-user information system has been developed which supports parallel development of objects, historical information and automatic maintenance of the database via editing of structured objects.

LIBLAB – Laboratory for Library and Information Science

Roland Hjerppe

LIBLAB studies methods for access to documents and the information contained in the documents, concentrating on catalogues and bibliographic representations, and on the human factors of library use. Current interests are focused on i.a. document architecture issues, the merging of information from libraries, archives and museums, spatio-temporal information and Geographic Information Systems, and formal approaches to the analysis of qualitative data.

LOGPRO – Laboratory for Logic Programming

Jan Maluszynski

LOGPRO has its research concentrated on foundations of logic programming, relations to other programming paradigms and methodology.

MDA – People, Computers and Work

Toomas Timpka

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

NLPLAB – Natural Language Processing Laboratory

Lars Ahrenberg

NLPLAB 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

PELAB conducts research in the area of tools and programming languages for software development and maintenance. Current projects include tools for semi-automatic bug location, debuggers for parallel languages, dependence analysis of programs, generation of efficient compilers from denotational semantic specifications, very high level languages and programming environments for scientific computing, and generation of parallel code for mathematical models.

RKLLAB – Laboratory for Representation of Knowledge in Logic

Erik Sandewall

RKLLAB conducts research on logic-based principles for the design of intelligent autonomous agents. This includes research on non-monotonic logics, logics for reasoning about action and change, fuzzy logic, algorithms for planning and temporal prediction, and related topics. It also includes research on methods for the systematic description of physical systems on a discrete level, and on architectures and tools for complex real-time systems.

VITS - Development of Information Systems and Work Contexts

Göran Goldkuhl

VITS is a research group studying information systems development in relation to organisational aspects. Special research interest/projects on: Methods for change analysis, information requirements analysis and informations systems evaluation. Strategies for information systems architecture. Relations between methods and CASE tools (CASE shells).

1.3.2 CENIIT – The Center for Industrial Information Technology

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

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

- *Algorithms for Geometrical Contact Detection*, (Per-Olof Fjällström)
- *Engineering Databases*, (Tore Risch)
- *Computer Support for Autonomous Manufacturing,* (Anders Törne)
- *Autonomous Real Time Systems,* (Jacek Malec, Erik Sandewall)
- *Conceptual Text Representation for Automatic Generation and Translation*, (Lars Ahrenberg)

CENIIT projects at IDA are described in greater detail in chapter 15 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 16 of this report, with additional details given in Appendix B - Graduate Study Programme and Appendix C - Courses and Teachers in the Undergraduate Education 1991/92.

1.4.1 Undergraduate education

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.

The *computer science and technology programme* (Datatekniklinjen), which started 1975, is a 4.5 year programme with an annual admittance of 120 students leading to a

Masters degree in Engineering (civilingenjörsexamen). This study programme has an equal emphasis on software and hardware related courses. At the end of the study line a number of specializations are offered. One such specialization of special importance is in software engineering, where students do assignments in development and maintenance of large software systems in an industrial environment. Important issues in addition to software methodologies and programming support are project management, quality assurance, economic and legal aspects.

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

IDA also contributes to other study programmes with a computer science profile. In the *industrial engineering and management programme* a new specialization has been introduced towards economic information systems. The *mechanical engineering programme* has a specialization that combines mechanical and computer engineering. We believe that especially research in artificial intelligence with application in robotics will be significant within this specialization.

A new engineering programme in *computer engineering* has been developed during the last year. These engineering programmes are two years and give an engineering exam (ingenjörsexamen) and the new programme accepts 30 students every year. The focus is on operation and maintenance of computer systems with courses in C and UNIX, operating systems, computer networks and distributed data bases.

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

IDA also gives a number of single subject courses. Some new courses can be mentioned: *Electronic Media, Information System for Health Care* and *Information Resources in Society.*

1.4.2 Graduate education

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

It will naturally take a number of years before improvements in the educational activities show up in the form of increased production of Ph. D.'s. The current rate of degrees awarded is 6-7 Ph.D's and 8–9 Licentiate degrees per year.

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

1.4.3 Continuing education

Continued eduction in form of course programs has decreased in the last few years, compared with what was done from the middle to the end of the 80's with large programs specially designed for people from the industry. We are ready to start such programs again, as soon as demand arises.

The department is also arranging shorter courses and seminars for the purpose of technology transfer. A series of software tutorials, SOFT, has been in operation for a number of years. The tutorials, SOFT-7 to SOFT-11 have been organized during the last two years on the subjects: Object-oriented engineering databases, Intelligent handbooks for technical documentation, Introduction to cognitive engineering, Software communication skills, Reliable software production.

1.5 Technology Transfer Activities

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

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

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

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

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

Since the start in 1984 a number of companies have participated in the knowledge transfer programme: SE-Banken, Ericsson Information Systems, ASEA, Alfa-Laval, Philips (now NobelTech), Sandvik Coromant, Volvo Cars, Volvo Data, Pharmacia LKB Biotechnology, Ericsson Telecom and TeleSoft (formerly TeleLogic Programsystem). The programme also involves cooperation with research institutes, such as The Defense Research Establishment and The Road Traffic Institute.

The programme includes activities such as project cooperation, courses and exchange of personnel. Each company participant is assigned a faculty supervisor and is associated with one of the research laboratories. Courses and other training activities are organized and coordinated at the department level.

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

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 commercial software, hardware, or designs to be delivered. In many cases we can instead refer to existing spin-off companies, and we may also encourage IDA employees to form new spin-off companies in order to catch an opportunity. However, we strongly promote the transfer of research results into products to be used outside the university. Thus, for instance, the MDA group is now transferring their multimedia information system environment to a company, where it will be casted into a commercial product and an engineering information system built in CAELAB is introduced at ABB Stal.

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

1.6 International Cooperation

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

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

- the Prometheus project, which is part of the European EUREKA programme. Prometheus is concerned with future traffic and automobile systems and is a joint effort by the European car manufacturers. RKLLAB has made significant contributions to PROART, the AI-oriented subproject of Prometheus.
- ESPRIT projects, where for instance the Laboratory for Logic Programming, the People, Computers and Work group, the Application Systems Laboratory, the Programming Environments Laboratory and the Laboratory for Representation of Knowledge in Logic are involved in projects.
- the AIM programme, where the People, Computers and Work group participate in the DILEMMA project.
- COST projects. Current involvement in for instance in the CoTech project, studying computer-supported cooperative work. A previous effort was a European project on Computer Architectures for A.I., together with the Free University of Brussels (Belgium), the University of Rome ‘La Sapienza’ (Italy), and Delphi S.A. in Pisa (Italy).

IDA also has extensive contacts with university and industrial research laboratories, primarily in the USA but also in Europe and to a certain degree also in the Far East.

1.7 Computer Facilities

The main part of IDA is situated in the university’s E building. About one third of the department’s personnel are located in a new temporary building close to the E building, called the E++ building. The premises are in general well planned and very nicely designed, but unfortunately at present heavily overcrowded. Some extra space will be available for the department in the former computer centers area in the B building.

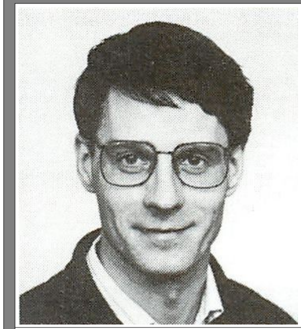
With respect to equipment, IDA has gone through a transition phase where now about 250 networked SUN workstations provide the main computing service. Further, there are about 140 Macintoshes, PCs and some HP and VAX workstations in the department, as well as some special-purpose computers such as an IBM AS400.

More details on equipment are given in Appendix D.

ACTLAB

2 The Laboratory for Complexity of Algorithms

Computational geometry
Analysis of algorithms
Data structures



Per-Olof Fjällström
director

Some of the members in the laboratory for Complexity of Algorithms.
Back row: Jan Petersson, Per-Olof Fjällström
Front row: David Andersson, Bodil Mattsson-Kihlström.

Figure 1-1. *The E building of the Department of Computer and Information Science.*

2.1 Introduction

The Laboratory for Complexity of Algorithms is concerned with the design and analysis of efficient algorithms and data structures for combinatorial and geometric problems, and the study of the inherent complexity of these problems in simple models of computation. Members of the laboratory believe that work on efficiency of algorithms and data structures is no less important than the development of new programming methodologies, or new faster computers[1].

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

The major funding 1991 is from CENIIT (the Center for Industrial Information Technology), a special programme for interdisciplinary research within Linköping University. Moreover, STUF is funding a project concerning parallel algorithms for certain geometric problems.

2.2 Laboratory Members

Laboratory leadership and administrative staff:

Per-Olof Fjällström, Ph. D., associate professor
Bodil Mattsson-Kihlström, secretary

Graduate students and research assistants:

David Andersson (up to July, 1992)
Jan Petersson

2.3 CENIIT project: Geometrical Algorithms

The topic of this project is efficient algorithms for geometric problems, in particular *three-dimensional problems* and geometric problems occurring in *motion planning*. The project ended in June 1992. The various activities within the project can be summarized as follows.

Motion Planning:

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

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

Parallel Algorithms for Geometric Problems:

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

Polyhedral Approximation of Bivariate Functions:

In many applications we need to approximate bivariate functions, where the function value is initially known only at irregular locations. To do this it is common to construct a function which interpolates *all* the given data. However, if we have large amounts of data it would be better if we could find a more compact but still accurate representation of the data. A method to achieve this was developed, see [Fjä91]. Moreover, in [Fjä89] the corresponding univariate problem was investigated.

Algorithms for the All-Nearest-Neighbours problem:

The all-nearest-neigbours problem can be stated as follows. Given a set of points, find for each point its nearest neighbour in the set. This problem (and its variations) occurs in a number of applications, for example, statistical analysis, pattern recognition, collision detection, etc.. In [FPK91] a number of algorithms (both sequential and parallel) for this problem are described.

2.4 CENIIT project: Algorithms for Contact-Detection

This project started in July 1992 and is done in collaboration with Prof. Larsgunnar Nilsson and Zhong Zhi-Hua at IKP. The need to detect contact between geometrical models of physical objects occurs in applications such as computer-aided design and simulation of systems consisting of moving or deforming parts. For example, in the simulation of a car collision a substantial part of the calculations is concerned with detecting contact or penetration. Since the geometric model of a car can consist of up to 50 000 surface elements, it is also clear that sophisticated contact-detecting algorithms are required. The goal of the project is to develop and evaluate experimentally algorithms for contact-detection.

2.5 Parallel Algorithms for Geometric Problems

This project started in July 1990. In the following we give a short introduction to the problem area, and describe the problems of specific interested to us.

During the past fifteen years a multitude of sequential algorithms for geometric problems have been developed. However, geometric problems often occur in on-line applications, where the demand for short response times may rule out sequential computation. Therefore the design of parallel geometric algorithms has received growing attention.

Parallel algorithms have been developed for a number of different computational models, each of which differs with respect to coordination of and communication between processors. In parallel computational geometry the *parallel random access machine* (PRAM) has become predominant. In this model a number of synchronized identical processors perform the same instructions on different data, and communicate by exchanging data through a global shared memory. Although this model cannot be considered to be *physically* realistic, it has turned out to be useful for studying both the *parallel complexity* of geometric problems (i.e. what can be parallelized) and the *logical structure* of parallel computation as such (i.e. how it can be parallelized).

Most research on parallel geometric algorithms has focused on problems in two-dimensional space. However, many important problems are of higher dimensionality, and it is often hard to generalize two-dimensional algorithms to cater for the corresponding multidimensional problems. For this reason we think that it is important to develop parallel algorithms for problems in multidimensional space.

In this project we are focusing on the *convex hull* problem. The computation of convex hulls is central to many applications, such as *pattern recognition* and *image processing*. Through appropriate transformations, algorithms for this problem can also be used to solve other geometric problems. In particular three-dimensional *Delaunay triangulations* can be computed by transformation from four-dimensional convex hulls. The problem of computing such triangulations occurs in applications such as *approximation of trivariate functions* and *volume visualization*.

Delaunay triangulations belong to a class of problems called *proximity problems*. In this project we have also investigated parallel algorithms for another proximity problem, the *all-nearest-neighbours problem*. This work is presented in [FPK91].

2.6 External contacts

Per-Olof Fjällström: Participated in the *Third Canadian Conference on Computational Geometry,* Vancouver, Canada, August, 1991.

Per-Olof Fjällström: Participated in the *Second Workshop on Algorithms and Data Structures,* Ottawa, Canada, August, 1991.

Per-Olof Fjällström: Participated in the *ALCOM Summer School on Efficient Algorithms Design,* Aarhus, Denmark, August, 1991.

Courses for Graduate Students

An important task of the group is to spread knowledge of algorithm design and analysis among graduate students within the department. The following graduate courses have been offered for the academic year 1991-92:

Design and Analysis of Algorithms.
Introduction to Parallel Algorithms.

The course on parallel algorithms was given in collaboration with Ola Petersson, Department of Computer Science, Lund University.

2.7 References

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

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

[Fjä91] **P-O. Fjällström:** Polyhedral Approximation of Bivariate Functions. *LiTH-IDA-R-91-24.* Also in *Proc. Third Canadian Conference on Computational Geometry,* Vancouver, Canada, August 6-10, 1991. Accepted for publication in *Computer-Aided Design.*

[FPK91] **P.-O. Fjällström, J. Petersson, J. Katajainen:** Algorithms for the All-Nearest-Neighbors Problem. *LiTH-IDA-R-91-25.* Presented at the *15th Conference on System Modelling and Optimization* (IFIP), Zurich, Switzerland, September 2-6, 1991.

Figure 2-1. *The E++ building.*

ASLAB

3 The Laboratory for Application Systems

Knowledge-based systems
Human-computer interaction
Industrial software technology

Sture Hägglund
director

Some of the members in the laboratory for Application Systems. .
From the left: Rego Granlund, Birgitta Franzén, Pär Carlshamre, Mikael Lindvall, Jonas Löwgren, Brant Cheikes, Sture Hägglund, Kristian Sandahl, Eva Lundberg-Ragnemalm, Göran Forslund.

Figure 3-1. *David L. Parnas lecturing at the SOFT-11 "Reliable Software Production" seminar at IDA, October 19-20, 1992.*

3.1 Introduction

The research[1] in the Application Systems Laboratory (ASLAB) has traditionally been 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 software engineering technology. Projects usually take an experimental approach and emphasize participation in application-oriented projects with industry and the public sector.

3.2 Summary of current research

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

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

In addition, the laboratory currently hosts a group dealing with

- *industrial software technology*, the IST project, in particular object-oriented approaches to development and maintenance of large-scale industrial software systems and empirical studies of software engineering experiences in industry.

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

The major activity concerning development environments for generic knowledge systems, including intelligent tutoring support, during the last year has been based on

1. The work in ASLAB is mainly supported by NUTEK, and the university. Additional funding comes from CENIIT, HSFR, FOA and others.

a concentrated effort in an application domain defined together with Pharmacia Biosystems AB. Several prototype systems have been developed concerning planning of protein purification experiments. (See Ph.D. dissertations by Sandahl and Eriksson.)

Another focus of research is on human-computer interaction, especially with respect to knowledge-based consultation systems. A generalized model for a UIMS (User Interface Management System) architecture which supports dialogue interfaces to expert systems has been developed (Löwgren). Our approach identifies two different structures in an expert system user interface: the *surface dialogue* and the *session discourse*. The management of the session discourse provides a possibility to make the dialogue structure independent of the underlying reasoning process of the system, by using a planning formalism based on operators defined by speech acts. Current work focuses on methods for supporting the design of human-computer interfaces with a special emphasis on explicit usability criteria and the design rationale (Carlshamre).

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

In related work, a critiquing/negotiating approach is applied to the problem of designing multiple inheritance hierarchies in terms of their logical consequences. (Eklund).

The bordering area between knowledge systems, database technology and computer based decision support is also of interest. We cooperate with other groups in the deparment on these subjects, for instance IISLAB/LINCKS (Padgham) concerning diagnosis reasoning by encoding classification knowledge in type hierarchies with inheritance (Zhang). Another topic of study is information management in command and control applications, where we cooperate with the Defense Research Establishment (FOA) on evaluation methods in iterative systems development (Näslund) and training simulators (Granlund). In the area of intelligent tutoring, we are also involved in process control applications (Ragnemalm).

3.3 ASLAB personnel

The following list presents the current staff and persons contributing to project activities in ASLAB during 1991–1992.

Laboratory leadership, administrative and technical staff:

Sture Hägglund, Ph. D., professor.

Birgitta Franzén, administrative assistant.
Mats S. Andersson, senior research engineer.

Researchers and graduate students:

Pär Carlshamre, M. Sc. (starting 1992)
Brant Cheikes, Ph. D. (visiting researcher 1992/93)
Henrik Eriksson, Ph. D. (at Stanford 1991-93)
Peter Eklund, Ph. D. (until fall 1991)
Göran Forslund, M. Sc., (starting 1992)
Rego Granlund, M. Sc., (starting 1992)
Tim Hansen, Tekn. Lic (until spring 1992)
Jonni Harrius, M. Sc.
Mikael Lindvall, M. Sc.
Jonas Löwgren, Ph. D.
Torbjörn Näslund, Fil. Lic.
Eva L. Ragnemalm, M. Sc.
Ivan Rankin, B.A., Fil. Lic.
Kristian Sandahl, Ph. D.
Tingting Zhang, M. Sc.

IST project personnel:

Bengt Lennartsson, Ph. D., 1991/92
Robert Glass, visiting professor
Iris Vessey, Ph. D., visiting professor

Other project personnel:

Jon Brewer, M. Sc., 1991/92
Christopher Cowperthwait, M. Sc., 1991/92
Thomas Grönwall, M. Sc., 1991/92

Associated persons:

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

Marianne Almesåker, FMV
John Brewer, Pharmacia Biosystems AB.
Ulf Cederling, University College of Växjö
Östen Dahl, Dept. of Linguistics, Stockholm University.
Christina Hellman, Dept. of Linguistics, Stockholm University.
Ulf Idesten, Ericsson Programatic AB
Karin Mårdsjö, Institute of Tema Research
Tommy Nordqvist, FOA 5.
Östen Oskarsson, consultant
Thomas Padron-McCarthy (on leave from graduate studies)
Hans-Erik Pettersson, VTI
Martin Rantzer, Ericsson Programatic AB
Greger Sahlberg, Pharmacia Biosystems AB.

Rein Schandersson, VTI
Anne Steinemann, Stanford University.
Yvonne Waern, Dept. of Psychology, Stockholm Univ. / Inst. of Tema Research.
Bengt Österlund, Pharmacia Biosystems AB.

3.4 Engineering Environments for Generic Knowledge Systems

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

The most important reference domain is advisory systems for the planning of biotechnical experiments, with practical applications developed in cooperation with Pharmacia Biosystems AB. These experiments involve the use of complicated laboratory equipment and domain experts are regularly consulted both for initial advice and if experiments fail. Other areas we have explored include model-based diagnosis (Hansen) and advice on traffic safety management (Forslund).

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

3.5 Knowledge system design with active expert participation

Kristian Sandahl

Beginning in the mid-80's, techniques for knowledge acquisition based on interviews have been taken more or less for granted. With those techniques, the knowledge engineer interviews the expert, who either takes a passive role or becomes a tutor. After having gained an initial understanding of the expert's knowledge, the knowledge engineer formalizes and encodes it into a knowledge base. Using the knowledge base, a prototype system is demonstrated to the expert, who validates the outcome. This process is iterated until expert acceptance is reached.

Based on this approach, we performed and systematically documented three real-world case-studies, A3, SOPRANO and RUFX (Sandahl 1992). A common factor of these otherwise substantially different projects was that the work was organized in a way which partially changed the roles of the expert and the knowledge engineer, making the experts more active in the design of the knowledge base. The success of the three projects' ability to cope with several problems was explained in terms of an

Active Expert methodology for development of knowledge-based systems. (Sandahl 1990, 91)

At a high level, we divide the Active Expert methodology in three phases:

- The preparation phase. This includes problem analysis, specification, and creation of means for building and maintaining the knowledge base. Evaluation criteria for entering the subsequent blocks should be specified.
- The knowledge entering phase. In this block, the expert enters the knowledge assisted by the knowledge engineer.
- The delivery phase. The delivery includes physical migration of the system to the end-users' site, as well as validation, marketing and maintenance.

Our work focuses on the methodological problem of transferring knowledge from domain experts to non-expert end users. Experiences from a number of previous expert system projects, form the basis for the current design of the Knowledge-Linker support environment as described below.

A description of the active expert approach to knowledge engineering and an analysis of experiences from a series of applied projects is found in the Ph.D. dissertation by Sandahl (Sandahl 1992).The research undertaken was organized in the Knowledge-Linker project, where a generic knowledge engineering environment was designed. The work contains both an architectural and a methodological component. Knowledge acquisition is strongly supported by a knowledge acquisition tool, which in turn is maintained with a meta-tool. The migration is accomplished by separating the environments for development and utilization of the knowledge base. The environments are bridged with a migration tool which customizes a delivery version of the knowledge base. The end-user accesses the knowledge base flexibly through various interpreters depending on the problem he wants to solve and his own qualifications.

The Active Expert development methodology is formulated in three versions depending on the degree of the tool support provided, namely an unsupported version, a knowledge-acquisition tool supported version, and a meta-tool supported version. The intention is that the Active Expert methodology should support the creation of knowledge acquisition tools and/or meta-tools.

3.5.1 Meta-level tools for knowledge acquisition

Henrik Eriksson

Developers can use tools that support the design of domain-oriented knowledge-acquisition tools to create an environment in which domain specialists can enter and edit knowledge bases. Such *meta-tools* take as input high-level definitions of knowledge-acquisition tools, and produce as output descriptions of the target knowledge-acquisition tools that can be compiled to executable programs. Next, these target knowledge-acquisition tools can be used to support the development of the final application systems. This development approach assumes that domain-oriented knowledge-acquisition tools provide better support than do generic knowledge-acquisition tools, and that such specialized knowledge-acquisition tools can be developed at low cost.

The *meta-view* supported by a meta-tool is the way in which target knowledge-acquisition tools are specified in the meta-tool. We have developed a meta-tool, Dots, that supports an *abstract-architecture* view of the target tools. In Dots, the developer specifies the target knowledge-acquisition in terms of its architecture. In essence, the developer designs the target tool from a library of user-interface, knowledge-representation, and knowledge-transformation components. Dots is task and domain independent in the sense that it does not assume any task or domain for the knowledge-acquisition tools it generates. However, Dots makes assumptions on the architecture of the target knowledge-acquisition tools.

During 1991/92, Dr. Eriksson has been a visiting researcher at the Knowledge Systems Laboratory (KSL), Section on Medical Informatics at Stanford University, Stanford, California. During the visit, he has participated in an ongoing research project, Protégé-II, at the KSL. The interest of the Protégé-II group is very close to the research on meta-tools for knowledge acquisition at the ASLAB. The Protégé-II project aims at developing computer-based tools that support software developers in designing knowledge-based systems. The development of a knowledge-based system can be broken down into the design of a problem solver that accomplishes domain tasks and the definition of a knowledge base that represents the expertise required by the problem solver for various inferences in the problem-solving process. The Protégé-II project addresses both of these aspects. In Protégé-II, problem solvers are composed from primitive problem-solving methods, which can be reused in several projects. These primitive problem-solving methods can be configured to specific tasks by providing additional information about the tasks. In addition to configuring problem-solving methods for tasks, developers must acquire and operationalize in a knowledge base the expertise required to accomplish the tasks. In the Protégé-II approach, domain-oriented knowledge-acquisition tools are used to acquire domain expertise for the knowledge base. Protégé-II provides meta-tool support for the design of these domain-oriented knowledge-acquisition tools.

3.6 Knowledge-based human-computer dialogue models

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

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

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

3.6.1 User interface management

Jonas Löwgren, Pär Carlshamre

Researchers have argued at length for the benefits of separating the implementation of user interfaces from their underlying functionality. Numerous tools and architectures have been developed to support such separation. UIMS tools, for example, are intended to support the design as well as the execution of user interfaces. When it comes to expert systems, however, the separation models found in existing UIMS work falls short of the mark. The reason is that expert system architectures typically regard the structure of the dialogue between system and user as a side effect of the reasoning required to solve the domain problem. This holds in particular for backward-chaining consultation systems: the system has a goal to deduce a certain proposition, and searches through its rulebase for rules that could contribute to the goal. The conditions of rules found are posted as new goals. The system asks the user only about information that it cannot deduce from its rulebase. The resulting dialogue is thus shaped both by the structure of the rulebase and the inference strategy. This in turn means that updates to the knowledge base can have unwanted effects on the dialogue structure. Moreover, if the system designer wants to achieve a certain dialogue structure, editing the rulebase is not the most convenient way of doing it.

To address the problem of user-interface management in expert systems, we have developed an expert system UIMS called Ignatius (Löwgren 1992). It consists of two modules: a surface interaction manager, responsible for the interaction techniques of the user interface, and a session dialogue manager which addresses the dialogue structure (in the sense discussed above). The surface interaction manager has a graphical editor, where the designer constructs the surface of the user interface. For the dialogue structure, the designer specifies plan schemas in a declarative textual language. These plan schemas can be seen as parameterized dialogue fragments. When the interface is executed together with the underlying expert system, the dialogue manager uses planning techniques to construct a dialogue structure according to the designer's specifications. The reason for using techniques as powerful as planning is to achieve the necessary flexibility; in expert system dialogues, the next question that the system is going to ask the user typically depends on the answers given to previous questions. The planning techniques of the dialogue manager also take care of the cases where the user asks for clarification of a domain concept or volunteers a piece of information before the system intends to ask for it.

The plan schema representation used in Ignatius for dialogue structure specification and execution is not limited to consultation systems. For example, Sokolnicki (Sokolnicki 1990) used it as a part of his proposed architecture for a tutoring system. We have also discussed elsewhere (Löwgren 1991, 1992) how it can be used for critiquing systems. Finally, the Ignatius architecture is an important part of our work on knowledge communication. In this work (Waern et al., 1992), we have collaborated with the Department of Psychology at Stockholm University in analyzing knowledge-based systems as media for communication of knowledge from expert to user. The architectural requirements on knowledge-based systems that came out of this analysis are addressed suitably by Ignatius.

3.6.2 Expert critiquing and text generation

Ivan Rankin, Jonni Harrius

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

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

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

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

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

Supporting an expert critiquing mode is important not only as a means to improve user acceptance, but also in situations where we cannot 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 development and validation of an expert system. More precisely, the usage may be restricted to critiquing services during early development stages with a still imperfect knowledge base. When confidence in the system is established, users can be allowed to rely more on the system's advice.

The work by Rankin demonstrates how text generation based on general-purpose linguistic principles can be achieved in the context of a critiquing system. This project has investigated the utility of more straightforward approaches in a medical application (the Crime system). In the generalized approach, a system called Arest, (Harrius 1989) was implemented based on Mann's *rhetorical structure theory.* Rankin emphasizes the problem of deep generation, rather than concern with the surface text realization. Understanding how to establish the goal the text is to achieve, determining the appropriate level of detail, planning for the presentation and finally organizing a cohesive message presents a stimulating challenge for research. Continued work in co-operation with Stockholm University (Dahl, Hellman) will further develop a theory for rhetorical relations, suitable for generation of critiquing and explanatory texts.

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

On a more fundamental level, the comments produced by a critiquing system can be generated in different ways. One sometimes distinguishes between differential and analytical critiquing. Differential critiquing is achieved by having the system produce its own solution to the problem at hand, compare it to the one proposed by the user and identify the differences. The knowledge base of an analytical critiquing system, on the other hand, may not be capable of solving the problem by itself. It is instead designed to analyze the proposed solution, looking for flaws, errors or suboptimal solutions. We are interested in studying these foundational mechanisms.

The systems Crime and Arest (Harrius 1992), which we built as illustrations of the dynamic text generation described above, are examples of differential critiquing systems. In this context, we have developed a knowledge representation where domain knowledge is integrated with rhetorical structure. Together with an enhanced inference algorithm, it enables us to perform dynamic critiquing tasks such as text generation in a computationally feasible way.

In contrast, the Kri systems (to be described below) are analytical. Our conclusion from these experiments is that the nature of the problem domain is a major factor in selecting the underlying mechanism. Differential critiquing allows for greater flexibility and more informative representations, but cannot be applied in domains where the problem is open-ended or ill-defined. The reason is simply that it is not feasible to make the system generate its own solution to compare with the user's.

3.6.3 Critiquing as user-interface design support

Jonas Löwgren et. al.

Our interests in user-interface development and critiquing systems were brought together in the Kri projects, where the overall idea was to develop critiquing systems containing knowledge about user-interface design. We expected these systems to be useful support tools for user-interface designers. The Kri projects were carried out in collaboration with the National Defense Research Establishment (FOA 53).

The first project addressed the domain of pseudo-graphical user interfaces such as the ones designed using the simulated graphics of a VT100 terminal. Such interfaces are typically form- and menu-based and the keyboard is used for navigation as well as data input. We used traditional knowledge acquisition techniques (expert interviews, etc.) to develop an analytical critiquing system called Kri (Löwgren 1990). In the preliminary validation of Kri, we found that it was important to support the credibility of system comments on the proposed design. To justify the comments, we integrated textual design guidelines from standard literature in the area into the system.

The goals of our next project in this area were (1) to address a commercially relevant design domain, and (2) to investigate the use of knowledge sources other than human experts. We decided to concentrate on graphical user interfaces developed using the OSF/Motif tool-kit. Knowledge acquisition for the new system was done by analyzing the Motif Style Guide and the collection of design guidelines by Smith and Mosier (Smith and Mosier 1986). The new system, called Kri/AG, was integrated with the commercial user-interface development tool TeleUSE. The designer could create a design prototype in TeleUSE and then submit it for evaluation by Kri/AG. User Interface Management Systems (UIMSs) are today widely recognized as bringing several advantages to the process of software development. By separating the user interface from the functionality of the application being developed, maintainability, reusability and prototyping is greatly enhanced. Unfortunately, UIMSs of today are not directly applicable to expert systems development. The reason for this is that in expert systems, the application functionality has traditionally been regarded as the determining factor for the structure of the user interface. We are developing a user interface management architecture for expert systems, analogous to a UIMS, which gives the expert system developer the ability to design the user interface of the system as a separate entity without affecting the problem-solving capabilities of the expert system.

3.6.4 Negotiating Network Hierarchies

Peter Eklund

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

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

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

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

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

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

This work was reported in the Ph. D. dissertation by Eklund.

3.6.5 Intelligent tutoring and training simulators

Brant Cheikes, Eva Ragnemalm, Rego Granlund

A prime motivation 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 for a tutoring system, which can be used to train inexperienced personnel in decision making, especially for unfamiliar or unusual situations. (Sokolnicki 1991)

We believe the potential for re-use of knowledge in new applications or for different purposes, such as for instance 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 the system. The system's task is then to coach and supervise the user and give hints and advice whenever needed.

In a current project, where the aim is to use simulations to train process-control operators, we are studying the possibilities to improve the resulting understanding of the involved processes by extending the traditional mathematical simulation models with more heuristic knowledge (Ragnemalm, Cheikes). Of particular interest is the possibility to relate strategies for explanations to the coaching exercised by the system in a training situation. Such training simulators should have a great potential for improving decision making and for developing the knowledge and competence for people professionally involved in dynamic decision making. The critiquing approach to knowledge-based consultations, as discussed above, is especially interesting when combined with intelligent tutoring. General issues in the context of developing and using simulators for training purposes in command and control applications are also pursued in a joint project with the Defense Research Establishment, FOA 53, (Granlund).

3.6.6 Evaluation in iterative development of support systems

Torbjörn Näslund

When developing systems intended to support activities such as management of processes, companies, and decision making, not only technical issues need to be considered. For such a system to be effective and provide useful support, it is necessary to achieve and maintain a proper interplay between the system, the user, the task to be performed, and the organization where the system will be used.

All these factors can seldom be handled simultaneously by the same developer. We have thus outlined an approach for development, where the support system is developed in an iterative process. In this process, responsibility for development and evaluation is divided between two roles; the developer role and the evaluator role. In our studies, we have placed the main emphasis on evaluation activities in this process, and have found that the outcome of these evaluations can be important injections of knowledge into the development process. This leads to a continuous shift of focus between technical aspects of the support system and issues such as usability, understandability, and effectiveness. Such a repeated shift of focus will increase the chances that the system in the end will be able to give effective support to its user.

Studies of methodologies and work practices in systems development should preferably be performed in real-world projects. We have thus in cooperation with the National Defence Research Establishment (FOA 53) studied a number of projects where support systems for military use were developed. The results of these studies, which were reported in a licentiate thesis (Näslund, 1992), show that evaluations are useful for finding potential deficiencies in the given support. Consequently, the outlined approach for development is valuable. However, we also noted some problems with the use of the approach. Many of these problems had to do with lack of explicitness in formulation of objectives and ideas, and with resource-consuming communication between the developers and the evaluators in the projects.

In our continued studies of how to support the development process by emphasizing evaluation activities, we thus intend to include studies also of opportunities to support communication and explicit formulation of key issues during iterative development. This should result in an increased understanding of how to develop useful support systems.

3.7 Classification models for knowledge-based diagnosis

Tingting Zhang

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

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

This work is carried out in cooperation with IISLAB. In a first investigation, we have built a medical diagnosis system for urinary tract infections. Our system, LIC (A Lattice-based Inheritance Classifier), uses the lattice-based inheritance model developed by Lin Padgham. The purpose of the implementation is to evaluate the use of this representation, both from the point of view of the adequacy of the resulting system for diagnosis, and also from the point of view of organization of the knowledge base, ease of obtaining and maintaining the knowledge and use of the knowledge base for tasks other than diagnosis. The results of this study will be presented in a Ph.D. dissertation in the spring 1993.

3.8 Industrial Software Technology

Bengt Lennartsson (91/92), Kristian Sandahl (92-)

The objective in the project Industrial Software Technology (Industriell programvaruteknik, IST) is to study the programming-in-the-many and the programming-in-the-large aspects of software development and maintenance. Since the large scale effects cannot easily be studied in an academic environment, the approach is to start with observations and case studies in the software industry. The focus is on the development of embedded systems.

This project has a basic funding from CENIIT, with the funding period starting in July 1989. Initially the activity was conducted by Bengt Lennartsson and one graduate student, Ulf Cederling. Cederling's research is also supported by the University College of Växjö, where Cederling is employed, and the results are presented in a licentiate thesis, Industrial Software Development - A Case Study (Cederling 1992).

In this project the development process used at one particular Swedish company, NobelTech Systems AB, is studied. The focus is on reusability aspects of the developed software. Differences and similarities between their process and methodology and the corresponding approaches described in the literature are investigated.

In another project, maintainability and Modifiability aspects of software developed with object-oriented methods are studied (Lindvall). This project is funded by a grant from Ericsson Radio Systems, ERA, sponsoring one graduate student and carried out in cooperation with ERA.

Our research in this area is also supported by the Wallenberg foundation with a grant covering the cost for a guest professorship in software engineering. Under this grant, Robert Glass and Iris Vessey are regularly visiting our department and we have also been able to engage David L. Parnas for shorter visits. In this connection, several external courses and seminars (software tutorials, SOFT) have been arranged.

3.9 Joint projects and external cooperation

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

- Pharmacia, Uppsala. Cooperation on knowledge-based systems in support of experiment planning in bio-technology.
- The National Defence Research Establishment in the areas of cognitive engineering, simulation in command and control systems and techniques for evaluation of system designs.
- Programatic AB in the area of methods for integrated design of human-computer dialogues and user handbooks.
- The Road Traffic Institute in the area of knowledge-based systems for promoting safety in road traffic systems.
- Ericsson Radio Systems in the area of software engineering methods.

In joint projects with Psychology and Linguistics at Stockholm University (Yvonne Waern, now at the Tema Research Department, Linköping and, Östen Dahl), we study the cognitive and linguistic aspects of the communication that goes on in the interaction with knowledge-based systems. Within the department, cooperation is also extensive, e.g. with CAELAB, IISLAB, NLPLAB and MDA where joint projects are in progress.

Numerous seminars for industry have been given and we have also participated in the development of continuing education programmes in *knowledge engineering* and in *human-computer interaction*. International contacts are mainly oriented towards the US. We have regularly sent students to Xerox PARC as summer interns. Other important contacts are found at Stanford, University of Colorado at Boulder, George Washington University, University of Oregon and Pennsylvania State University. In Europe we have regular cooperation with e.g. Xerox EuroPARC and Univ. of

Limerick, Ireland (Kevin Ryan). We also participate in a COST project on Computer Supported Cooperative Work and in the ESPRIT project AMODEUS 2.

3.10 Publications

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

Ph. D. dissertations 1991-92:

Jonas Löwgren: Knowledge-Based Design Support and Discourse Management in User Interface Management Systems. 1991.

Henrik Eriksson: Meta-Tool Support for Knowledge Acquisition. 1991.

Peter Eklund: An Epistemic Approach to Interactive Design in Multiple Inheritance Hierarchies. 1991.

Kristian Sandahl: Developing Knowledge Management Systems with an Active Expert Methodology. 1992.

Licentiate theses 1991-92:

Ulf Cederling: Industrial Software Development - A Case Study, 1992.

Torbjörn Näslund: On the Role of Evaluations in Iterative Development of Managerial Support Systems, 1992

International publications 1991-92:

P. Eklund. Negotiating Inheritance Taxonomies in Conceptual Structures. In *Proceedings 3rd Scandinavian Conference on Artificial Intelligence*, Roskilde, pages 211–221, 1991.

P. Eklund. A Framework for Negotiating Conceptual Structures: Inheritance Hierarchies. In T. Nagle, N. Nagle, L. Gerholz, and P. Eklund, editors, *Current Directions in Conceptual Structure Research*. Ellis-Horwood, 1992.

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

H. Eriksson. Specialized Knowledge Acquisition Tool Support Compared With Manual Development - a Case Study. In *Proc. of the 7th IEEE Conference on AI Applications*, Miami, FL, February 1991.

H. Eriksson. Domain-oriented Knowledge Acquisition Tool for Protein Purification Planning. *Journal of Chemical Information and Computer Science*, 32(1):90–95, 1992.

H. Eriksson. A Survey of Knowledge Acquisition Techniques and Tools and Their Relationship to Software Engineering. *Journal of Systems and Software*, in press.

H. Eriksson and M. A. Musen. Conceptual Models for Automatic Generation of Knowledge-acquisition Tools. In T. Wetter, K.-D. Althoff, J. Boose, B. Gaines, M. Linster, and F. Schmalhofer, editors, *Current Developments in Knowledge Acquisition: EKAW'92*, pages 14–36. Springer-Verlag, Berlin, Germany, 1992.

H. Eriksson and M. A. Musen. Meta-Tools for software Development and Knowledge Acquisition. In *Proceedings of the AAAI Workshop on Automating Software Design*, pages 43–47, San Jose, CA, July 1992.

H. Eriksson, K. Sandahl, J. Brewer, and B. Österlund. Reactive Planning for Chromatography. *Chemometrics and Intelligent Laboratory Systems: Laboratory Information Management*, (13):185 – 194, 1991.

H. Eriksson, K. Sandahl, G. Forslund, and B. Österlund. Knowledge-Based Planning for Protein Purification. *Chemometrics and Intelligent Laboratory Systems: Laboratory Information Management*, (13):173 – 184, 1991.

H. Eriksson, Y. Shahar, S. W. Tu, A. R. Puerta, and M. A. Musen. Task Modeling With Reusable Problem-solving Methods. In J. H. Boose and B. R. Gaines, editors, *Proceedings of the Seventh*

Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Canada, October 1992.

L. Gerholz, T. Nagle, J. Nagle, and P. Eklund, editors. *Conceptual Structures: Current Research and Practice*. Ellis-Horwood Limited, 1992.

S. Hägglund. Investigating Consultation Paradigms for Knowledge-based Desicion Support. In *Proc. of the IFIP Working Conference on Support Functionality in the Office Environment*, Canterbury, 1991. North-Holland.

S. Hägglund. Office Applications Panel: Chairman's Summary. In *Proc. of the IFIP WG 9.1 Working Conference on Human Jobs and Computer Interfaces*, Tampere, Finland, 1991. North-Holland.

S. Hägglund. Expert Critiquing as a Paradigm for Giving Consultative Advice. In *Proc. of the ECAI'92 Workshop on Expert Judgement, Human Errors and Intelligent Systems*, Vienna, 1992.

S. Hägglund and I. Rankin. A Knowledge-based Critiquing Approach to Explanation Generation. In *Proc. of the AAAI Spring Symposium Workshop on Explanations*, Stanford, 1992.

J. Harrius. AREST, a System For Deep Generation of a Critique Using Aggregate Schemas. In *Proc. of the ECAI'92 Workshop on Expert Judgement, Human Errors and Intelligent Systems*, Vienna, 1992.

J. Harrius. Text Generation in Expert Critiquing Systems Using Rhetorical Structure Theory and Conceptual Graphs. In T. Nagle, N. Nagle, L. Gerholz, and P. Eklund, editors, *Current Directions in Conceptual Structure Research*. Ellis-Horwood, 1992.

J. Löwgren. Supporting Design and Development of expert System User Interfaces: The Ignatius Environment. *IEEE Expert*, 7(4):49–53, 1992.

J. Löwgren and T. Nordqvist. Knowledge-based Evaluation as Design Support for Graphical User Interfaces. In *Human Factors in Computing Systems (CHI'92 Proceedings)*, pages 181–188, Monterey, 1992. ACM Press.

T. Näslund. Injective Formative Evaluation for Enhancing Computer Support for Cooperative Staff Work. In *Proceedings of the Human Interface '92, Kawasaki*, October 1992.

L. Padgham and J. Löwgren. User interface management for an object oriented database system. In *Proceedings of the First International Conference on Information and Knowledge Management*, Baltimore, Maryland, 1992.

I. Rankin and J. Harrius. Rhetorical Aggregates in Text Planning. To appear in *Proceedings of the 3rd Nordic Conference on Text Comprehension in Man and Machine*, Linköping, Sweden, 1992.

K. Sandahl, H. Eriksson, T. Padron-McCarthy, T. Sokolnicki, and B. Österlund. Meeting the Requirements of Knowledge Management. *Expert Systems with Applications*, 3:259–267, 1991.

T. Sokolnicki. Towards Knowledge-based Tutors: A Survey and Appraisal of Intelligent Tutoring Systems. *Knowledge Engineering Review*, 6(2):59–95, 1991.

Y. Wærn, S. Hägglund, J. Löwgren, I. Rankin, T. Sokolnicki, and A. Steinemann. Communication Knowledge for Knowledge Communication. *Int. Journal of Man-Machine Studies*, 37(2):215–239, 1992.

Some other publications 1991/1992:

S. Hägglund and R. Hjerppe. (Eds). Selected Term Papers on Methodology of Research in Computer Science. Technical Report LiTH-IDA-R-91-30, IDA, Linköping University, 1991.

H. Eriksson. Dots - a Guided Tour. Memo 91-01, ASLAB, IDA, Linköping University, 1991**.**

T. Näslund. Evaluation as a Means for Improving Development of Decision Support Systems. Technical Report LiTH-IDA-R-92-01, IDA, Linköping University, 1992. (Presented at the Doctoral Consortium at the 11th Int. Conf. on Information Systems, Copenhagen.).

T. Nordqvist, S. Löf, and J. Löwgren. Crude: A Critiquing User Interface Design Environment. Video presented at *Human Interface '91*, Japan, 1991.

K. Sandahl and A. Brogren. Erfarenheter från användarstyrt kunskapssystemprojekt. Memo 92-02, ASLAB, IDA, Linköping University, July 1992. In Swedish.

K. Sandahl and B. Möller. Datalogisk problemanalys inom projektet utveckling av expertsystem för ökad trafiksäkerhet. Memo 92-01, ASLAB, IDA, Linköping University, June 1992. In Swedish.

CADLAB

4
The Laboratory for Computer-Aided Design of Digital Systems

Krzysztof Kuchcinski
director

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

Some of the members in the laboratory for Computer-Aided Design of Digital Systems.
Xinli Gu, Krzysztof Kuchcinski, Bodil Mattsson-Kihlström, Mats Larsson, Zebo Peng.

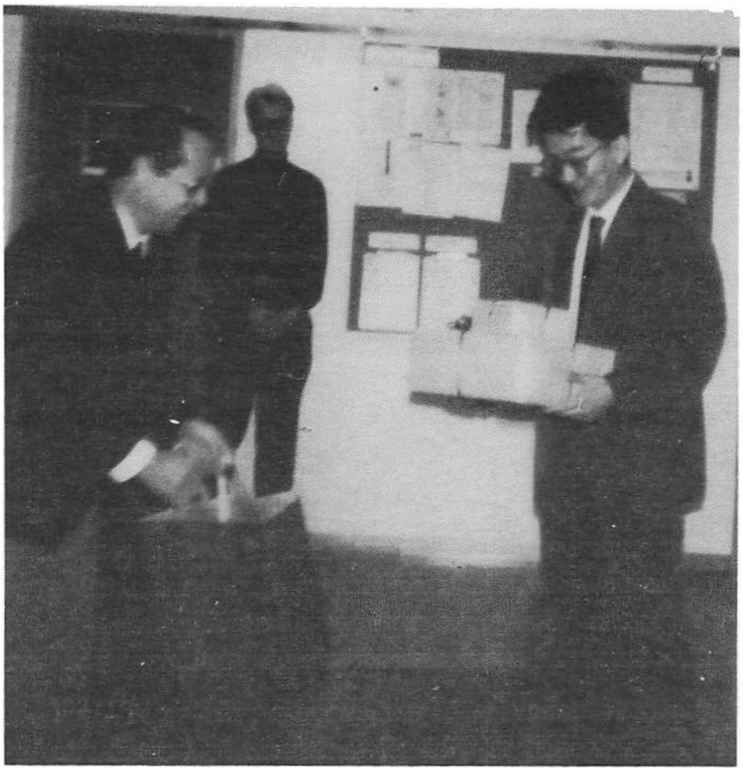


Figure 4-1. *Xinli Gu is congratulated after presenting his licentiate thesis by his advisor Kris Kuchcinski (left) .*

4.1 Introduction

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

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

Our research activities concentrate mainly on different methodologies and computer-aided design tools which constitute a complete design environment. Except high-level synthesis, mentioned above, we are also interesting in special design styles, like synthesis of pipeline structures and synthesis for testability. The complementary research is conducted on formal methods used for synthesis. There is also an ongoing project looking at hardware/software co-design problems which examines interaction between hardware and software in a design process.

4.2 Laboratory Members and Guests

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

Laboratory leadership and administrative staff:

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

Laboratory members having a Ph. D. degree:

Zebo Peng, Ph. D., assistant professor

Graduate students:

Xinli Gu, Lic.
Björn Fjellborg, Tech. Lic.

The work in CADLAB is mainly supported by NUTEK, The Swedish National Board for Industrial and Technical Development

Mats Larsson, M. Sc., (on leave at Cambridge University during 91/92)
Erik Stoy, M. Sc.

Visitors during extended periods:

Raimund Ubar, Ph. D., Tallinn Technical University, Estonia
Märt Saarepera, M. Sc., Tallinn Technical University, Estonia
Petru Eles, M. Sc., Technical University of Timisoara, Romania
Marius Minea, Technical University of Timisoara, Romania
Stanislaw Szejko, Ph. D., Gdansk Technical University, Poland

The following degrees were awarded to CADLAB members during 1991/92:

Björn Fjellborg completed his Ph. D. dissertation entitled *Pipeline Extraction for VLSI Data Path Synthesis*.

A total of 15 technical papers were accepted for publication or presented on local and international conferences by CADLAB members during 1990-92. The paper entitled “Compiling VHDL into a High-Level Synthesis Design Representation” by Petru Eles, Krzysztof Kuchcinski, Zebo Peng and Marius Minea got the “Best Paper Award” at EURO-DAC’92 conference on EURO-VHDL track.

4.3 The Objectives of the Present Research

High-Level Synthesis

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

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

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

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

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

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

To achieve the high-level synthesis goal, i.e., to generate a register-transfer implementation structure, a set of basic issues must be addressed. They include operation scheduling, resource allocation and binding as well as composition and decomposition of design modules. This basic activities can be considered as design transformations which change a design to achieve a goal or meet a constraint.

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. Different authors have reported a set of working algorithms addressing either an individual issue or several issues together. However, problems remain. One of the most difficult problems is that the different issues of high-level synthesis are closely interrelated and depend on each other. If they are solved separately as most of the existing algorithms do, the global optimization cannot be achieved. The main goal of the present research is to come closer to the global optimal solution for the high-level synthesis problem.

Formal Hardware Synthesis

Formal hardware verification is one of the methods which can formally prove the correctness of the designed hardware. Usually it is a proof of the correctness of the implementation with respect to the specification. Although many complex digital systems have been verified with this approach, it has failed to make any substantial impact on design practices in industry. We believe that there exist several reasons for this. Firstly, experience has shown that the effort of verifying unstructured digital designs is often a couple of orders of magnitude greater than the effort of creating the actual design. Secondly, theorem proving is inherently difficult and hardware designers have no background in formal methods. Finally, theorem proving methods are not integrated with existing CAD tools and design practices.

The CADLAB's research activities dealing with *formal hardware synthesis* issues are a step in the direction to overcome the previously stated problems. The formal synthesis, as defined in the framework of our project, can also be seen as complementary to the previously described high-level synthesis activities. It begins with a formal specification, and progresses by refining it, i.e. applying correctness-preserving transformations to the specifications. The process ends when the specification has been transformed into a related implementation. The result of the synthesis process is a proof that the implementation is correct with respect to the specification. The goal of this research is to create an interactive environment which will allow the designer to formally transform the design specifications into an implementation.

Hardware/Software Co-Design

Most application-specific systems in information technology consist of both hardware and software components. As the complexity of such systems is getting greater and greater, it is becoming more and more difficult to design them. The only way to handle the complexity is to equip the designers with efficient mechanisms to explore different design alternatives and make appropriate decisions. We also need a new view of system design and a new design methodology which treats the hardware parts and the software parts as integrated components of the designed systems.

Traditionally, the hardware and software design paths were separated early in the design cycle and once departed, they did not get together until it was almost time to ship a product. In such a situation, software design was dictated by hardware design decisions that were made during the first phase of the system design cycle. Software development did not track changes made during the hardware design cycles, nor did it influence hardware design. It was impossible to move functionality between the hardware domain and the software domain or to change the hardware/software interface to improve the overall design of a system. It has been shown by many investigations such a strict separation of design activities into two distinct domains does not work well.

The main goal of the research is to develop a design methodology which allows a total interaction of the software and hardware design activities. In particular, the goal is to develop a *hardware/software co-design* environment that will allow system designers to start the design process with a high-level behavioral specification and generate a hardware implementation at register-transfer level as well as a software program which together will implement the given specification.

4.4 Current Research Projects

Our research activities at CADLAB in the synthesis subject have so far been concentrating on how to improve the chance of getting a globally optimal solution during a synthesis process. The CAMAD system addressed this particular problem. The related work on pipeline extraction from ordinary algorithm descriptions forms a preprocessor for the CAMAD system. It resulted in a prototype system, PiX (Pipeline Extractor). The results of this work has been presented in Björn Fjellborg's Ph. D. dissertation. The research in the area of testability resulted in the project which integrates the testability evaluation and improvement into CAMAD system. This work becomes a licentiate topic for Xinli Gu.

Another part of our activities in the synthesis area include the work on formal methods which can be used for hardware synthesis. This work is a licentiate subject for Mats Larsson.

The natural extension of high-level synthesis project is research on system level design. By system level design we mean hardware/software co-design. Preliminary studies and some research have been initiated in the area.

4.4.1 Design Representation and Synthesis

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

To make this strategy possible we have developed a design representation, called the Extended Timed Petri Nets (ETPN), which can be used throughout the whole design process. The ETPN notation has been developed by Zebo Peng (see for example [1], [3]). It consists of separate but related models of control and data parts. Petri nets are used here to represent the control flow, which allows explicit expression of concurrency and parallelism. The data part of the design representation, on the other hand, is modelled as a directed graph. Nodes of the graph are used to represent operations on different levels of granularity. The arcs are used to model the data flow between the operations.

It is assumed that different specification languages can be used as an input for ETPN creation. Presently a Pascal-like language called ADDL and a VHDL synthesisable subset can be translated to the ETPN design representation [8].

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

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

The formulation of the ETPN design representation and the basic framework of CAMAD were described in Zebo Peng's Ph. D. dissertation. The current work on CAMAD is intended to integrate more design decisions into the optimization process. Examples of such design decisions deal with synchronicity/asynchronicity trade-offs, module-library component selection, design verification, and testability issues.

4.4.2 Pipeline Extraction

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

Current design tools for automatic pipeline synthesis exploit pipelining by analysing loops in the design. However, they lack the ability to automatically select the parts of the design that can benefit from pipelining. Pipeline extraction performs this task as a first step of pipeline synthesis. This research (see for example [7]) addresses the problem of pipeline extraction from a general perspective, in that the search for pipelines is based on detecting potential for hardware sharing and temporal overlap between the individual tasks in a design. Thus loops appear as an important special case, not as the central concept. A formalism for reasoning about the properties underlying pipelinability from this perspective has been developed. Using that, a series of results on exactly what mutual dependencies between operations that allow a pipelined schedule with static control sequence to be constructed are proven. Furthermore, an evaluation model for designs with mixed pipelined and non-pipelined parts has been formulated.

This model and the formalism’s concept of pipelinability form the basis for a heuristics-guided branch and bound algorithm that extracts an optimal set of pipelines from a high-level algorithmic design specification. This is implemented in the pipeline extraction tool PiX, which operates as a preprocessor to the CAMAD VLSI design system. The extraction is realized as transformations on CAMAD’s Petri net design representation. For this purpose, a new model for representing pipeline constraints by Petri nets has been developed. The results from PiX are competitive with those from existing pipeline synthesis tools and also verify a capability to extract cost-efficient pipelines from designs without apparent pipelining properties.

4.4.3 Design for Testability

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

The testability analysis is achieved by defining a quantitative measurement of testability for the ETPN design representation [2, 4]. The measurement gives both the controllability and observability of each connecting point in the data path. Both the controllability and observability are in turn defined by a combinational factor and a sequential factor. The combinational factor measures the complexity of the hardware circuits involved to control a specified data (for controllability) or to observe a specified output value (for observability) at some given point. The sequential factor, on the other hand, measures the corresponding time steps (e.g., number of clock cycles) needed to control or observe the given point. In the ETPN notation, the combinational factor is mainly determined by the structure of the data path. The sequential factor, on the other hand, is based on the properties of the Petri net. We have developed a set of heuristics for computing testability factors.

The testability measures are used in the early stage of the design process to improve testability of the design [9]. Two strategies for determining difficult-to-test parts have been used and evaluated. The first one, called worst part select first, selects the part with the worst testability, recalculates testability and iterates the process until a given constraints are satisfied. The second one, called sequential component selection, allows to find registers responsible for difficult-to-test parts. Later the known testability improvement transformations, such as scan methods, are applied to improve testability of the design.

4.4.4 Formal Hardware Synthesis

In the framework of the project we want to relate the designer's intuitive notion of design progress to a concept in formal logic. The relation often used to relate specification and implementation in formal verification is logical implication. The expression implementation $\supset$ specification can be read as the implementation achieves the specification and we believe that this captures the intuition behind refinement of specification.

Goal-oriented reasoning is the most popular method of proving theorems. This is reflected in the goal-oriented interfaces associated with many mechanised proof assistants. Reasoning with such systems begins by stating the goal, and progress by simplifying the goal, or by breaking it into several subgoals whose collective truth would establish the original one. This style of reasoning is well suited for developing correct hardware through formal verification but because goal-oriented reasoning relies on users knowing what they want to prove in advance, it is not well suited for formal synthesis. Window inference, on the other hand, is a style of reasoning where the user is invited to transform one expression into another one, while preserving some relationship between them. We conjecture that just as goal-oriented reasoning is a natural style of reasoning for developing provably correct hardware by formal verification, window inference is a natural style of reasoning for developing provably correct hardware by formal synthesis.

The project makes use of the window inference technique to develop a method for designing correct hardware by stepwise refinement of the specification. The idea is to automatically generate a proof of correctness (with respect to the specification) as a by-product of the design process itself. By this we will avoid a difficult problem of verifying the designed hardware which can be very often developed in a very unstructured way.

4.4.5 Hardware/Software Co-Design

To design hardware/software concurrently in an integrated environment leads to more possible design trade-offs, thus resulting in a higher degree of complexity of the designs as well as the design process [3]. In order to develop an integrated environment, we first need a uniform way of capturing the design in the intermediate design stages so that hardware/software trade-offs can be easily made and evaluated. We need also to partition the intermediate design representation into the hardware subsystem and the software sub-system and compile them into the corresponding implementations. Finally, we need a set of automatic or semi-automatic tools to help the designers quickly explore the different design solutions.

The main goal of the project is to develop a top-down design methodology for hardware/software co-design as well as supporting tools. As an input functional specification the VHDL language is assumed. A VHDL program is used to specify the high-level behavior of the designed system without presupposing the hardware/software boundary or implementation details. The front-end compiler translates the VHDL specification into an architecture-independent intermediate representation. This representation will capture parallel computations explicitly and allow the partitioning of hardware/software to be done in different ways. The main criteria to be used for the partitioning of hardware/software are the performance requirement and the execution profile of the design which will be obtained by simulating the intermediate representation. When the preliminary partitioning is done, a hardware synthesizer (high-level silicon compiler) can be used to generate the hardware structure at register-transfer level and a software compiler can be used to compile the object code. However, these two activities must be able to communicate their design decisions to each other either directly or via the intermediate representation.

In the current state the partitioning procedure has been implemented and tested on several examples [6]. The prototype VHDL compiler to the ETPN representation is also implemented and preliminary results are available [8]. However, the work on the integrated environment are still under development.

4.5 Related Activities and External Cooperation

CADLAB is involved in the graduate courses program of IDA. During the reported time our laboratory contributed to the program by giving two courses. During the spring 1991 the course “Hardware description and the VHDL Language” was given by Krzysztof Kuchcinski and Zebo Peng. The course “Parallel Computers: Architecture and Programming” was given by Krzysztof Kuchcinski and Zebo Peng during spring 1992. Additionally, a series of research seminars on digital systems design automation were given every week by members of CADLAB.

Krzysztof Kuchcinski organized the Fourth Swedish Workshop on Computer System Architecture held in Linköping, January 13-15.

In terms of international cooperation we have established contacts with Prof. Mario Stevens from Eindhoven Technical University in the subject of system level design. Regular contact are established with Tallinn Technical University. Prof. Raimund Ubar and his students cooperate with our group in the test generation and testability issue. Finally, a research project on VHDL compilation to the ETPN internal design representation has been started in cooperation with Technical University of Timisoara.

4.6 References

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

1. **Z. Peng, K. Kuchcinski:** A Multi-Level Optimization Strategy for Behavior-Entry Synthesis, *Proc. of the Fourth International Workshop on High-Level Synthesis, Kennebunkport, Maine, USA, October 15–18, 1989.*
2. **K. Kuchcinski, Z. Peng:** Testability Analysis in a VLSI High-Level Synthesis System, *Microprocessing and Microprogramming, The Euromicro Journal, vol. 28, no. 1-5, March 1990, pp. 295–300.*
3. **Peng, Z., Fagerström, J., Kuchcinski, K.**, A Unified Approach to Evaluation and Design of Hardware/Software Systems, *ICSE-13 Workshop Hardware/Software CoDesign, Austin, Texas, USA, May 13, 1991.*
4. **Gu, X., Kuchcinski, K., Peng, Z.**, Testability Measure with Reconvergent Fanout Analysis and Its Applications, *The Euromicro Journal, Microprocessing and Microprogramming, nrs 1-5, August, 1991.*
5. **Kuchcinski, K.**, Toward Automatic Test Pattern Generation for VHDL Descriptions, *Proceedings of the EURO-VHDL, Second European Conference on VHDL Methods, September 8-11, 1991, Stockholm, Sweden.*
6. **Peng, Z., Kuchcinski K.**, Architectural Partitioning of Application Specific Systems, *DSA-92, Fourth Swedish Workshop on Computer System Architecture, Linköping, January 13th-15th, 1992*
7. **B. Fjellborg:** Pipeline Extraction for VLSI Data Path Synthesis, *Linköping Studies in Science and technology. Dissertations. No. 273.* 1992.
8. **Eles, P., Kuchcinski, K., Peng, Z., Minea, M.**, Compiling VHDL into a High-Level Synthesis Design Representation, *EURO-DAC'92, September 7-10, 1992, Hamburg, Germany.*
9. **Gu, X., Kuchcinski, K., Peng, Z.**, An Approach to Testability Analysis and Improvements for VLSI Systems, *The Euromicro Journal, Microprocessing and Microprogramming, nrs 1-5, August, 1992.*

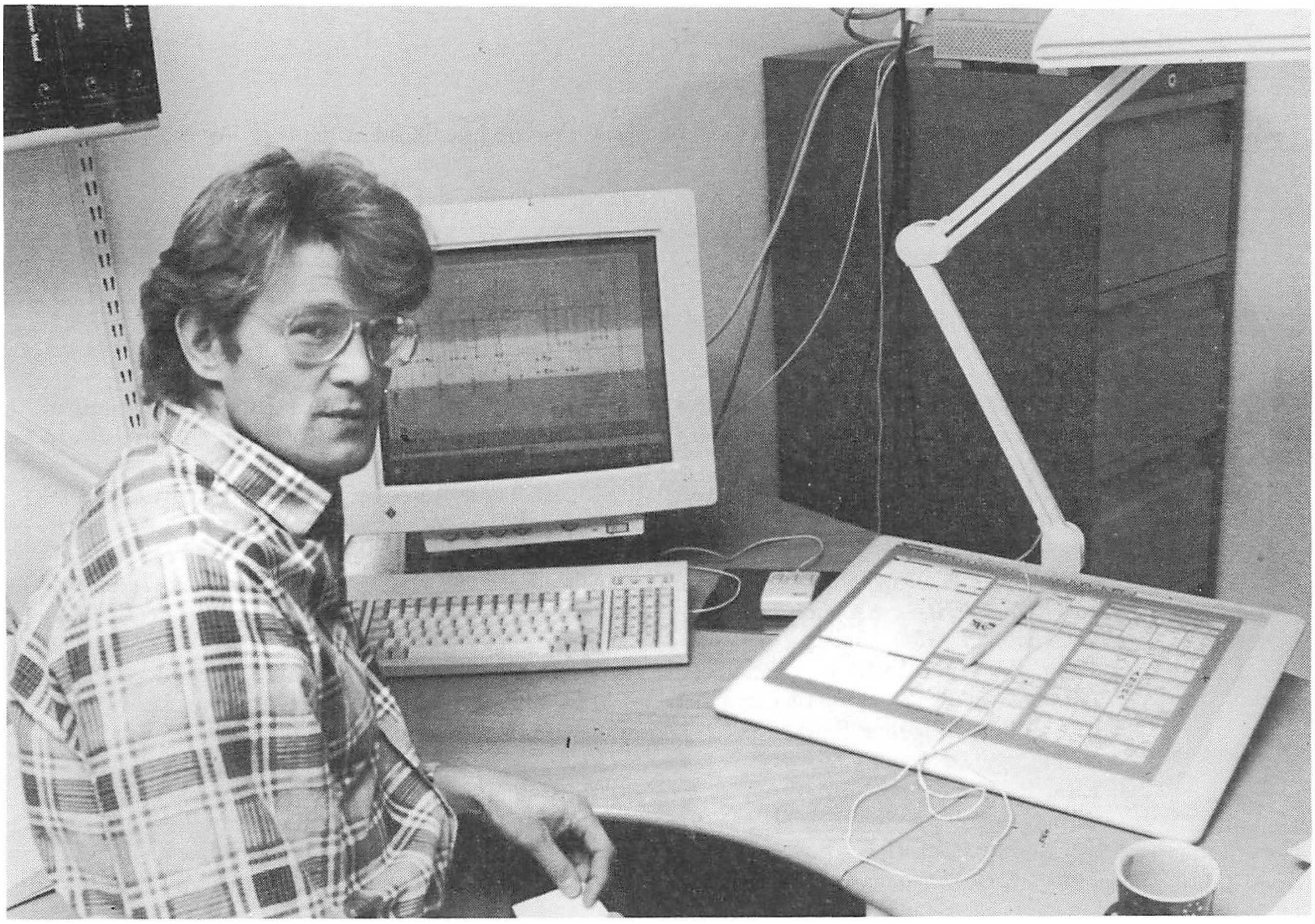
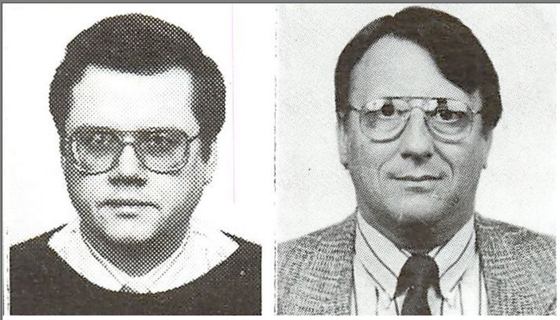


Figure 4-1. *Sun IPX workstation with digitizing pad.*

CAELAB

5
The Laboratory for Computer Assistance in Engineering



Anders Törne
director

Tore Risch
professor

Design Support
Engineering Databases
Automation
Real Time Systems

Some of the members in the laboratory for Computer Assistance in Engineering.
From the left: Magnus Werner, Kjell Orsborn, Tore Risch, Anders Törne, Olof Johansson, Per Bergman, Gustav Fahl, Zebo Peng, Martin Sköld, Peter Loborg.

5.1 Summary

The laboratory for Computer Assistance in Engineering was formally started in July 1989. The work in the laboratory is interdisciplinary between Computer Science and Engineering, i.e., mechanical, electrotechnical, or chemical process 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. The research consists of applying results from computer science research to assist this process and from the experience, initiate and inspire research in computer science. The laboratory has two directions of research:

- Engineering Information Management (Tore Risch)

  Incorporates basic technologies for engineering databases. Important concepts are distribution, heterogeneity, active databases, and databases in real time systems. Applicative projects are also conducted in cooperation with industry; one on object-oriented data modelling and one involving knowledge based techniques. Professor Tore Risch started his work in the laboratory in march 1992.

- Computer Support for Automation (Anders Törne)

  Incorporating task level programming, real time architectures for supervisory control, programming of autonomous manufacturing environments, and real time systems.

5.2 The laboratory research

5.2.1 General

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

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

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

The laboratory research is focused on the architectures and representations needed in systems for computer support in the engineering process or, more specifically, principles for software construction in this area of computer applications. In practice the work is done by case studies, the subsequent generic analysis of the useability of the architectures and representations, and the development of special architectures and

The work in CAELAB is mainly supported by CENIIT (The Center for Industrial Information Technology), Linköping University, by Nutek (The Swedish National Board for Industrial and Technical Development) and by TFR (The Swedish Research Council for the Engineering Sciences).

representations for this application domain. Two basic directions of computer science research are involved - one is the *management* of data repositories for this information and support for the *data modelling* needed - the second is information representation and generation for *automation and control* of the manufacturing process.

Professor Tore Risch came to the laboratory in march 1992 and has rapidly built up a group of graduate students and projects, centered around *engineering information management*. In autumn 1992 this research has grown to be the larger part of the CAELAB research.

5.2.2 Method

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

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

Both research directions have a framework project. Within the Engineering Information Management group the AMOS architecture serves as an umbrella - ‘vision’ - for several subprojects. Each subproject contributes pieces to the AMOS vision. In this work, the verification of theorethical results is emphasized by incorporating them into the AMOS architecture. In the Computer Support for Automation group the ARAMIS system serves a similar purpose.

Contacts with industry and other departments are encouraged and considered very valuable. However, the risk in this is acknowledged. Industrial needs are considered as a compass needle, which does not mean that the result of the research will satisfy the needs of today - tomorrow. Rather the needs of tomorrow are important as a guideline.

5.2.3 Laboratory members

Laboratory leadership and technical/administrative staff:

Anders Törne, Ph. D., associate professor, laboratory director
Tore Risch, Ph. D., professor
Gunilla Lingenhult, secretary

Graduate students:

Anders Ekman, M. Sc.
Gustav Fahl, M. Sc.

Olof Johansson, Tech. Lic.
Peter Loborg, M. Sc.
Kjell Orsborn, M. Sc.
Martin Sköld, M. Sc.
Magnus Werner, M. Sc.

Associated researchers and supervisors:

Sture Hägglund, Ph. D., professor, IDA, supervisor
Zebo Peng, Ph. D., CADLAB, IDA, associated researcher
Jerker Wilander, Softlab AB, industrial expert
Alexander Lauber, professor, IFM, supervisor
Per Holmbom, M. Sc., IFM, project participant
Ole Pedersen, M. Sc., IFM, project participant
Bo Torstenfeldt, Ph. D., IKP, supervisor

5.3 Engineering Information Management

Future computer supported engineering environments will have a large number of work stations connected by fast communication networks. Workstations will have their own powerful computation capacities to store, maintain, and make inferences over local engineering data/knowledge bases, or *information bases*. Each information base is maintained locally by a human operator and is autonomous from other information bases. Each one will need a set of DBMS (database management system) capabilities, e.g., a data storage, a data model, a query- and data-modelling language, transactions, external interfaces, etc. The classical relational data modelling languages are not powerful enough for the manipulations needed, e.g., to build advanced models to filter and extract interesting information. Tools are also needed to adapt to dynamic changes in the information bases.

The *Engineering Information Management* projects are focused around an architecture (called AMOS, Active Mediators Object System) to model, locate, search, combine, update, and monitor information in such a dynamic engineering information system. Our approach introduces an intermediate layer of software to *mediate* between databases and the use in applications and by users. We call the class of intermediate modules *active mediators*, since mediators support ‘active’ and real-time database facilities, as will be explained in the project descriptions.

The AMOS related research has received support from three sources; from CENIIT - “Engineering Databases”, from Nutek - “Mediators for Information Management” and from TFR - “Query Language Optimization for Autonomous Information Bases”. Support have also been received from Hewlett-Packard by a donation of three large ‘snake’ workstations and a 1 Gbyte file serving facility.

Two applicative projects are also connected to this direction of research. Both receive support from Nutek. One is a cooperation project with ABB Stal, Finspång - “Decision Support for Plant Designers”. In this project, techniques and methods are studied for building a support system for the design of steam turbine plants. It is a practical study of how to use todays and tomorrows computer based tools to build a CAE-system, that is adequate for the needs of the manufacturing industry. The other concerns software engineering for numerical calculation systems. The project - “Effectivization of the Development, Maintenance and Use of Software Systems for Technical Calculations with Application to FEM-systems” - will develop a prototype for a knowledge based FEM-system to increase the performance of the designers using it.

5.3.1 AMOS - an architecture for active mediators

Tore Risch

The previously mentioned active mediators approach introduces an intermediate *mediator* layer of software between databases and their applications and users. An architecture for active mediators is developed, which is called AMOS (Active Mediators Object System). The AMOS platform will support DBMS facilities, such as a local database, a data dictionary, a query processor, transaction processing, remote access to data sources, etc. The query language will support object-oriented (OO) abstractions but still allow for declarative queries. The language will be extensible to allow for easy integration with other systems. We envision that knowledge, now hidden within application programs as local data structures, will be extracted and stored in AMOS. The query processing should therefore be efficient enough to encourage the use of local embedded databases without significant performance penalty. Queries should be allowed to span more than one AMOS system or other data sources. An important capability of AMOS is the support for *active* queries, i.e. daemons that execute when certain complex conditions change.

Figure 5-1 illustrates how a set of application programs access a set of data sources through active mediators. The integrators combine data from several data sources or other mediators, to form a uniform view of combined data. The monitors model the detection of significant data changes in some data source and notifies (indicated by arrows) applications or mediators. The following subsections describe the work on each kind of mediator more in detail.

To support the initial work on AMOS, a main-memory object-oriented DBMS is used. It was developed (by Tore Risch) at Hewlett-Packard Laboratories, Palo Alto, CA and provides an object-oriented query language, OSQL, and fast execution. The system is open and easy to modify for the research intended.

5.3.2 Integrators

Gustav Fahl, Tore Risch

Data sources may store information in different ways, e.g., different data sources could use different DBMSs and data models, similar data is represented with different data formats, etc. Therefore integrators are needed that retrieve and combine results from many data sources, that check the consistency of fused data, and that present a higher level view to application, e.g., using object abstractions. Integrators decouple applications from the necessity to maintain multiple data models for all possible data sources.

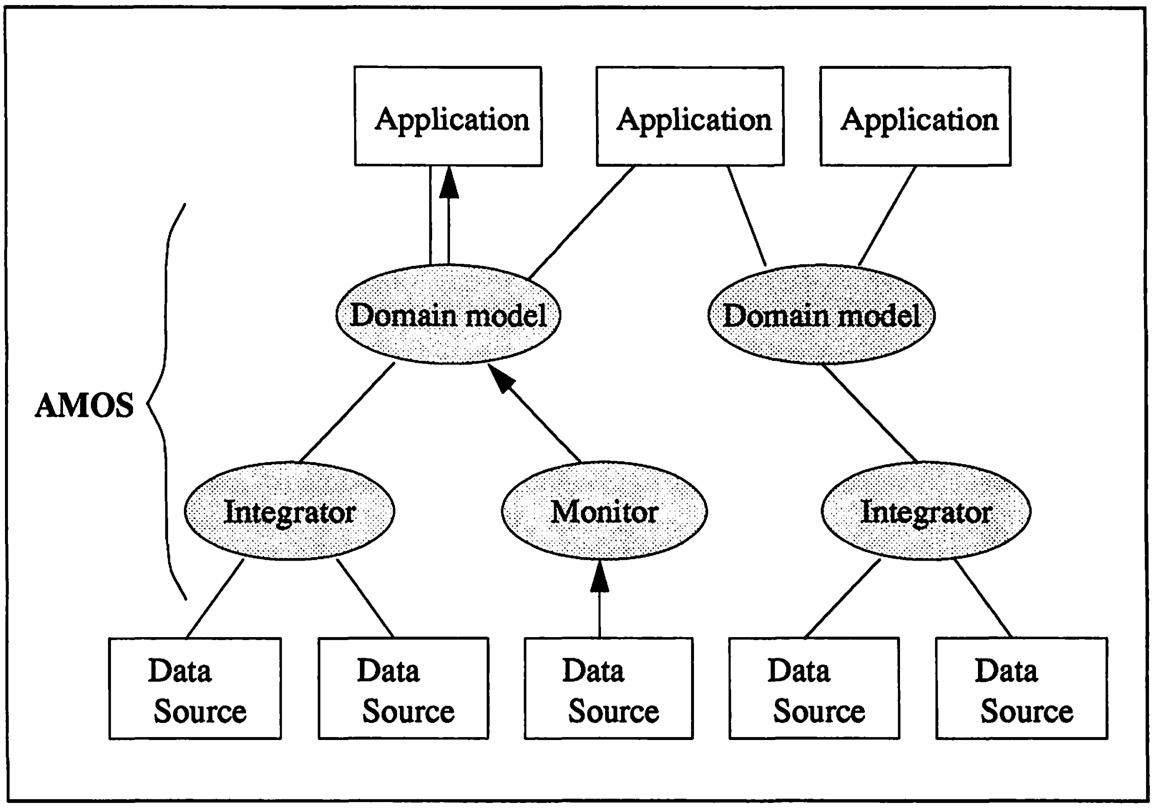


Figure 5-1. *Active mediators of different classes mediating between data sources and users/applications*

Gustav Fahl works on integration of relational databases and other data sources with AMOS. A particular problem with such an integration is to get object-oriented access to non object-oriented data sources. In the method chosen, the AMOS system will contain descriptions of how to generate object identifiers from each data source. The data source interfaces will dynamically generate object identifiers, which are maintained by AMOS. The method allows OO queries to be stated with transparent access to non OO data sources.

5.3.3 Monitor Models

Martin Sköld, Tore Risch

Some applications require a mechanism to handle the problem of dynamically changing contents and locations of data. For example, when a scheduling plan is initially made, the assumption is that the critical data for the execution of the plan is not changing. In practice these data are frequently changing. Therefore the planner should be notified when data is updated that was assumed to be constant when the plan was made, so that it can be adapted as well. Mediators should be provided that continuously monitors these invariant data and notify the planner when the invariants change to a significant extent. *Monitor models* allow the control to pass between cooperating application programs via AMOS. Of particular interest is to provide means to build monitor models that filter change in data sources, so that irrelevant changes are ignored.

Martin Sköld is working on adding active database extensions to OSQL. Rules are defined by pairs <condition, action>, where each condition is a declarative OSQL query and where each action is an OSQL database procedure block. An action is executed (i.e. the rule is triggered) when the condition becomes true. Unlike other active DBMSs, the condition queries may refer to derived functions (similar to views). Rules are furthermore parameterized and type overloaded, so that they can be instantiated for objects of different types. Data may be passed from condition to action of each rule, to allow for iterative action execution over sets of instances.

5.3.4 Distributed AMOS Systems

Magnus Werner, Tore Risch

It is desirable to state OO queries and to build OO models that span many AMOS systems. Therefore the system needs to contain means for intercommunication between AMOS systems as well as between AMOS and applications. The user will have access to a *multi-database OO query language*. The query language should have means for naming the individual databases accessed in a query. It should also be possible to define functions (methods, views) that access data in many AMOS databases. A query optimizer should optimize the complex distributed multi-database queries. Special optimization techniques are needed that choose between pipelined access to other AMOS servers and downloading of data.

Magnus Werner works on a transactional remote procedure call mechanism that handles low level message interfaces between AMOS servers. The query layer will be built on top of this mechanism. The mechanism will also support ‘callback’ so that many AMOS call each other interchangably. It is important to support transactional behavior so that each database can remain consistent after communication or software failures.

The work on multi-database query languages and the work on semantic differences between different AMOS databases connect distribution to the work on integrators. Furthermore, the means for monitor rule specification will be generalized to access more than one database, which connect this work and the work on monitor models.

5.3.5 Active Real-Time Databases

Martin Sköld, Tore Risch

Real-time databases are databases with added timeliness requirements such as transaction and operation deadlines, contingency plans, and timeliness trade-offs regarding data consistency, precision, and completeness. A real-time query language also needs ‘temporal database’ facilities that incorporate time, e.g., to access the clock or to timestamp and version data. The AMOS query language will be extended with such constructs. The active database facilities being incorporated in AMOS are important for real-time applications, e.g., to monitor combinations of sensor data and to perform actions when ‘interesting’ situations occur. The rule language will need to be complemented with timeliness constraints, e.g., for rule conditions and actions. Finally many real-time environments are distributed, where sensory data are collected and filtered on separate sites and where decisions and effectuations are made on other sites. This requires distributed execution of AMOS rules with timeliness constraints.

5.3.6 Domain Models

Vacant, Tore Risch

We also wish to extract domain knowledge now hidden in application programs and store them in mediators called *domain models*. Domain models facilitate the maintenance of large information bases by breaking them up into locally maintained information bases that are simple, inspectable, and autonomous. Examples of domain models are tax rule models, models to obtain a preferred part for a product, or models to describe properties of a user interface.

5.3.7 Locators

Vacant, Tore Risch

In large information bases, it is not trivial to know which data sources incorporate certain data. For this support, a class of mediators is needed, which given descriptions of the data to retrieve, locates the matching data sources. These mediators are called *locators*. In a simple environment the application will know exactly where the data sources are located, e.g., by knowing the exact locations of database tables. In a broadly distributed environment one may not have such direct 'handles' to the data sources, but rather give the locators descriptions of properties of what to look for. Locators provide a high level language for connecting data to application programs. The effect is to increase flexibility when information sources are changing.

Locators are similar to data dictionaries, but would not be centralized, would access autonomous databases, and would handle dynamic changes in the location of data sources. We currently put less emphasis on research on locators. It is presented here mostly for making the presentation complete.

5.3.8 Decision Support for Plant Designers

Olof Johansson, Sture Hägglund, Tore Risch

The design of large scale complex mechanical artifacts, such as steam turbine plants, will benefit tremendously if models of the plant can be stored in engineering databases. However, the application software for managing engineering data in terms of graphical user interfaces and application specific database storage structures is cumbersome to develop.

With a continuously changing market situation for manufacturing companies, and rapidly evolving software and hardware technology, the application specific system for plant modelling must be flexible and easy to adapt in order to meet new demands on plant design and changing computer technology. There is a strong need for efficient high-level support tools (Computer Aided Software Engineering tools) that ease the burden of maintaining and adapting this kind of systems.

Object-oriented data models can serve as compact specifications, i.e., of what information a steam turbine manufacturer wants to store in their engineering database.

These data models can be formalized and stored in a data dictionary. Knowledge-based techniques can be applied on the stored data models to automatically generate program code that implement a prototype of a plant modelling system for a particular database and windowing system.

In the project “Decision Support for Designers”, prototypes of design support systems are developed in cooperation with the industrial partner ABB STAL Finspång. They provide us with realistic requirements on the plant modelling system and expertise in plant design. We use object-oriented, and knowledge based techniques for developing CASE-tools that support development of plant modelling systems.

Our method is to apply advanced computer science technology on industrial problems. From practical experience with realistically sized prototype implementations generalizations are made, theories developed and software architectures are investigated which facilitate the implementation of CASE-tools.

This activity is directly funded via IT-4, Nutek to the same level as the costs for the industrial partner.

5.3.9 Effectivization of the Development, Maintenance and Use of Software Systems for Technical Calculations with Application to FEM-systems

Kjell Orsborn, Tore Risch, Anders Törne, Bo Torstenfelt

The project will use methods and techniques from computer science to achieve the goal implied by the title. A prototype system for a knowledge based approach to *scientific computing* (in particular FEM - finite element analysis) will be generated. The application area is initially confined to linear-elastic problems in two dimensions. An object-oriented database (preferably AMOS) will be used to store the product model and the information about the analysis process. Domain knowledge is integrated for example as rule structures.

The project approach is to try and combine different technologies to achieve the goal, i.e., conceptual modelling, object-oriented/knowledge based methods, object-oriented databases, graphical user interfaces, standards for product modelling and simultaneous engineering. An earlier project, 1987 - 1991, focused on the possibilities of using knowledge based techniques in software systems for damage tolerance design of aircraft structures at the Department of Mechanical Engineering. The experience is carried over to this project by Kjell Orsborn and Bo Torstenfelt.

The project received specific Nutek funding from June 1992.

5.4 Computer Support in Automation

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

- How to separate the environment independent information from the dependent and what restrictions must be put on the language for describing the information at different phases in the engineering process?
- How, when, and why information is incorrect, and how corrections are to be made, when the executing environment (the manufacturing environment) changes?

Particularly interesting areas are process and operations planning, factory scheduling, and control of the physical process at the manufacturing cell level.

To focus the research, three issues have been extracted:

- Tools for generation and monitoring of control in multi-machine environments.
- Representation and compilation for control in manufacturing.
- Distributed RT-databases for control in multi-machine environments.

This research is supported by CENIIT - “Computer Support for Autonomous Manufacturing Systems” and has also received Nutek support, from the program for “Adaptive Manufacturing Equipments”. The Nutek project is called “System Architecture and Sensor Integration in Flexible Manufacturing Systems”. The goal is to realize a task programmed industrial robot with sensory integration and is done in cooperation with professor Alexander Lauber at the department of Physics and Measurement Technology (IFM) at Linköping University.

5.4.1 ARAMIS – task oriented programming of manufacturing environments

Anders Törne

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

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

The goal for this project is to develop a programming and run-time environment for multi-machine manufacturing environments, including cooperating manufacturing cells. The project is used as a platform for other research activities within the Computer Support for Automation group. It corresponds to a subprocess in the engineering process which takes operation lists for the manufacturing task and generates an executable control program for the different cooperating machines. A change in the operation list, in the equipment or in the manufacturing requests imply changes to the program – the aim is to minimize or possibly automatize the effort needed to introduce these changes into the program.

5.4.2 ARAMIS Task Specification

Peter Loborg, Anders Törne

A forward-chaining rule based approach have been taken for the description of manufacturing tasks. The language combines *procedural abstraction* and *rules*. An action is a set of rules invoced procedurally. The temporal order between action invocations is a partial order allowing for parallel event flows. Determined by the rule condition, only a subset of the rules of an action are triggered for each invocation. The context for an action is determined by the state of the real world. The state is represented as the conjunction of object states in an object-oriented data model - the world model. A subset of the objects in this model is passed as arguments at invocation time to the action. Actions are recursive. Primitive actions changes the state of the objects in the world model.

A graphical editor and a run-time environment have been implemented on SUN Sparc Stations. The present work is concerned with exception handling, error recovery and resource handling.

5.4.3 Sensor/Actuator Specification

Martin Sköld, Anders Törne

Each object in the world model is represented as an *augmented finite state machine* where control algoritms are associated with each transition and state. When a state transition occurs in the Aramis world model (primitive in the language), the control algorithm is executed in the real-time control system and success or failure is acknowledged to the ARAMIS run-time system. Algoritms are also associated with each object state if this must be actively maintained (non-stable state). This work integrated with the task specification is presently tested.

Continued work will concern implementation of tools for sensor fusion and actuator fission specification. The sensor fusion (~ calculation of reliable data from redundant sources) behaviour is considered static for a specific sensor configuration and the same is assumed for actuator fission (~ effect achieved by redundant physical actuators). Se further 5.4.5.

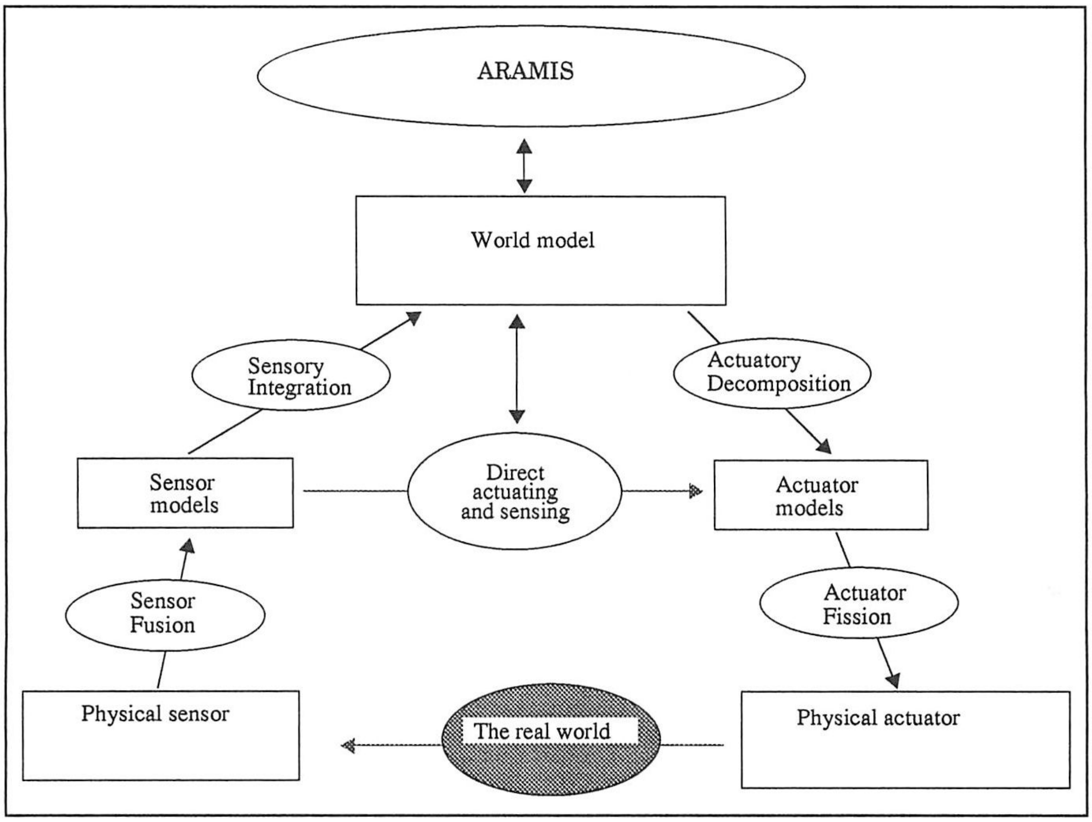


Figure 5-2 *A model of the dataflow and the different subprocesses when connecting ARAMIS to an executing environment*

5.4.4 Task Level Modelling

Per Holmbom, Ole Pedersen, Peter Loborg, Anders Törne

This project concerns developing benchmark tests and demonstrators for the ARAMIS-concept. The present benchmark is an assembly of four parts on a table. The initial position of the parts are unknown but visible from a vision system. The goal state is the ready assembly at a specific position. The task contains ambiguities in the identification of parts, alternative assembly plans depending on context and geometrical reasoning of assembly paths.

5.4.5 Sensor Integration

Anders Ekman, Anders Törne

Sensor integration deals with the dynamically changing relation between the sensor model data and the world model status and decisions made by ARAMIS. This involves for example *active sensing*, i.e., improving world state information reliability by execution of actions in the environment. It also involves the *context dependent* merge of data from different heterogeneous sensors - *sensor integration*.

5.4.6 Other Principles for Sensor/Actuator Data Flow

Ole Pedersen, Alexander Lauber

Instead of specifying the control algoritms in a programming language, neural nets may be used. The neural net maps state information to actuator signals. The assumption is that the necessary learning cases can be generated. The work includes using neural nets for sensor fusion and integration. A test has been made using neural nets to identify objects in the picture taken by a video camera.

5.4.7 Extended Petri Nets for Modelling and Analysis of Real-Time Manufacturing Systems

Zebo Peng, Anders Törne

The initial approach of this project is to use Extended Timed Petri nets (ETPNs) to describe real-time processes and manufacturing processes in particular. The idea is to partition the system into one controlling (embedded) system and one controlled system, similar to practice in control theory and discrete event systems. Both parts are described in ETPNs and the joint temporal behaviour can be analyzed. This behaviour is compared with a simpler specification in the same representation. This specification could then be used as a controlled system at a higher level of system description. A hierarchy of embedded control system may be described in this way, all in the same representation. This work has newly started.

5.4.8 Distributed RT-databases for Control of Manufacturing Processes

Martin Sköld, Tore Risch, Anders Törne

A distributed RT-database concept will enhance the communication and synchronization abilities between different manufacturing cells. Data may be extracted for monitoring of the production. This is a severe test on the distributed, real-time AMOS concept presented above (5.3.5). This work is presently initialized.

5.5 Other activities

A graduate course on “Real Time System Design” was carried out during autumn 1991 (Anders Törne). Participants came from Computer Science, Control Engineering and from University of Växjö. This autumn a graduate course on “Principles of Modern Database Systems” is given (Tore Risch). The laboratory seminar series included three seminars in spring 1992 on RT-databases for telecommunications (Jerker Wilander).

Professor Stanley B. Zdonik from department of Computer Science, Brown University, Rhode Island, USA, gave a seminar for IDA entitled “Object-Oriented Database Systems: Fact or Fiction” in october 1991. Dr. D. Tarabanis from IBM TJ Watson RC and Columbia University, visited the project in may 1992 and gave a seminar with the title “Sensor Planning and modelling for Machine Vision Tasks”. Professor Sharma

Chakravarthy from University of Florida visited the laboratory and gave a seminar on "Active Databases" in june 1992.

Within the university contacts are of course frequent with other groups within IDA - cooperation is established with CADLAB in the realtime systems area and is initiated with LOGPRO on deductive databases. IISLAB, RKLLAB and ASLAB are other laboratories which have close connection with the CAELAB activities.

From July 1992, Anders Törne is also commissioned as director of CENIIT - the Center of Industrial Information Technology - at Linköping University. This center has participants from most of the engineering departments at the university.

At other departments, the IFM-group lead by professor Alexander Lauber is participating in the Nutek-financed project on ARAMIS-related activities. CAELAB also have regular contacts with the department of Electrical Engineering and the department of Mechanical Engineering on workshops and CENIIT activities.

Industrial contacts are considered important for the activities in the laboratory. One project (Olof Johansson, IT4) is a joint project (50% financing from industry) with ABB Stal - an international swedish manufacturer of steam and gas turbines. Several joint seminars have been arranged for SAAB Military Aircraft in Linköping, concerning product modelling and the use of object-oriented databases in engineering support. Other contacts have been with Ellemtel, the joint research company of Ericsson and Swedish Telecom. We are presently investigating how a cooperation within the field RT-databases could be initiated.

5.6 Relevant publications since 1989

Tore Risch, Martin Sköld: Active Rules based on Object-Oriented Queries, LiTH-IDA-R-92-35, in special issue on Active Databases of *IEEE Data Engineering*, Jan 1993

Peter Loborg, Per Holmbom, Martin Sköld, Anders Törne: Programming Flexible Manufacturing Systems, in *Proceedings of Workshop on Adaptive Manufacturing Equipments*, Nutek 1992:53, Oct. 1992, CAELAB Memo 92-02.

Witold Litwin, Tore Risch: Main Memory Oriented Optimization of OO Queries using Typed Datalog with Foreign Predicates, LiTH-IDA-R-92-24, in special section on Main Memory Databases in *IEEE Transactions on Knowledge and Data Engineering*. Vol.4, No.6, Dec. 1992.

Gustav Fahl: Integration of Heterogeneous Databases in a Mediator Architecture, LiTH-IDA-R-92-23.

Peter Loborg, Martin Sköld, Anders Törne, Per Holmbom: A Model for the Execution of Task Level Specifications for Intelligent and Flexible Manufacturing Systems, LiTH-IDA-R-92-22, in *Proceedings of the Vth Int. Symposium on Artificial Intelligence, ISAI92*, Cancun, Mexico, Dec. 1992.

Peter Loborg, Anders Törne: A Hybrid Language for the Control of Multimachine Environments, LiTH-IDA-R-91-09, in *Proceedings of EIA/AIE-91*, Hawaii, June 1991.

Peter Loborg, Martin Sköld, Anders Törne: A Hierarchical Software Architecture for Control of Industrial Robots and Manufacturing Equipment, presented at *1st National Symposium on Real-Time Systems*, proc. in Tech.Rep. no30, Dept. of Computer Systems, Uppsala University, June 1991. CAELAB Memo 92-01.

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

Olof Johansson: Improving Implementation of Graphical User Interfaces for Object-Oriented Knowledge-Bases, Linköping Studies in Science and Technology, Lic. Thesis no 283.

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

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

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

Anders Törne: Planning for Assembly Tasks, FAMOS-seminar on “AI technology as a tool in automatic assembly, STU, Stockholm 1989, STU-info 720-1989.

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

Figure 5-3. *Students’ union building.*

EIS

Birger Rapp
director

6
Economic Information Systems

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

Some of the members in the group for Economic Information Systems.
From back: Anders G. Nilsson, Björn Helander, Rolf Rundfelt, Fredrik Nilsson, Bo Lagerström, Lars Mattsson, Bengt Savén, Peter Carlsson, Maria Olsson, Eva Elfinger, Lars Poignant, Thomas Bennet, Stefan Blom, Kristina Larsen, Camilla Sjöström, Jan Ollinen, Jörgen Andersson, Mehran Noghabai, Anette Stolt, Jaime Villegas, Anders Bäckström, Elisabeth Gudinge, Anna Moberg.

6.1 Overview

The subject area Economic Information Systems has become the focus of increasing interest throughout the last decade, reflecting the growing awareness of the value of information and the importance of adequately developed Information Systems. An expression of this new awareness was the foundation of a professorship in Economic Information Systems (EIS) in Linköping in 1987.

The evolution of society towards a “knowledge society” is also creating new conditions of employment and forms of organization, and allocating new roles to the individual. Information Manager and Division Controller are two examples of new kinds of employment which are relevant to Economic Information Systems. Research is going on, both internationally and in Sweden, in order to reach a better understanding of how information technology can be adapted to the condition of people and organizations. The role of information technology and its consequences have been given a new emphasis in connection with the strategic questions facing companies. New attention is being paid to the direction of companies and the control of the flow of goods and services, with the help of the new information technology.

Both technological and economic and management questions are related to Economic Information Systems. There is also a natural strong common area of interest with the field Computer and Information Science.

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

Today the area EIS offers a research education in three different ways. One possibility is the “traditional” way towards the licentiate or doctor degree. Another possibility is offered through the VITS lab under the leadership of docent Göran Goldkuhl. There exists also a special possibility to take a licentiate degree. In cooperation with five different accounting firms 3-5 researchers are offered this education each year. The firms are Bohlins-KPMG, Ernst & Young, SET, TRG-revision and Öhrlings-Reveko. The researchers are mostly graduated from Business Administration. The courses are offered in compact form during some research weeks. During the rest of their time the researchers are working at an accounting firm. This special education lasts for three years and it is unique. In Sweden it is only offered in Linköping. A number of adjunct professorships are going to be founded in order to improve the supervision. Recently one professorship in Swedish and International External Accounting was founded. It is held by professor Rolf Rundfelt.

We have initiated cooperation with the Department of Management and Economic Informatics, Latvia, University in Riga, Latvia, and Department of Economics Informatics, Faculty of Economics, Vilnius University in Vilnius, Lithuania. Both departments participate in our research programme, *Computer simulation for professional training and decision support.*

EIS also participate in a research project financed by NUTEK. It is an interdisciplinary project concerning risks in business. Six universities in Sweden are involved. The project has resulted in the publication of a number of books, among them Riskbedömning - kunskap om risker (Risk assessment - Knowledge about Risks), edited by Professor Birger Rapp. In the project four researchers and Professor Birger Rapp are participating from EIS, Linköping.

EIS has also special co-operations with IBM, Ericsson, Computer Associate, ABB-Data.

6.2 Research Projects in Economic Information Systems

Research within EIS is being conducted under the following main headings: information support, agency theory, IT and organizational solutions, computer simulation for management training and decision support, business control and accounting and auditing.

6.2.1 Information Support

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

One sub-project is devoted to the implementation of information systems. (Lars Mattsson) and another sub-project is devoted to the questions of evaluation. Applications are expected above all in the area of FMS-investments and office information systems. (Mehran Noghabai).

6.2.2 Agency theory

An agency relationship arises when a contract, explicitly or implicitly, is set up, where somebody (the principal) commits someone else (the agent) to do something on his behalf.

An organization can be seen as a nexus of contracts, written and unwritten, among owners of factors of production and customers. The contract between the owners and the managers of the organization is the top of the organizations internal structure of contracts.

Because information is not costlessly shared and incentives of principals and agents can not be costlessly aligned, agency costs are always present in agency relationship. Agency costs are costs to structure, administer and accomplish the agency relationship.

Three projects within this field are in progress at the department. One project studies the change in agency costs due to the managers increase in ownership and the heighten leverage of the company when a management buy-out is made. A management buy-out is a high-leveraged acquisition of a company, where managers of the company together with financiers become owners of the company. (Peter Carlsson)

The second project is financed by NUTEK and deals with the contract problems in the relation between the bank and the entrepreneur running a company. (Jonas Lind)

In the third project the book “Vem ska ta risken?” (Who will bear the Risk?) by Birger Rapp and Anders Thorstenson will be published by Studentlitteratur in 1993.

6.2.3 Information Technology and Organizational Solutions

Society is in a state of constant change. There has been a tendency for workplaces to concentrate in larger urban locations. Unfortunately this tendency has brought with it a worsening of the social climate, increased rents for businesses and long commuting-times for employees. Today there are examples where this tendency has begun to be defeated by the employment of modern information technology.

The new information technology facilitates access to and exchange of information, and also makes for a lot more flexibility in temporal and spacial organization. At present there are a number of ways to communicate within a business: personal contact, by telephone, electronic mail, video and fax. For the most part these techniques are easily accessible and give more choices. This entails new conditions for modes of working and forms of organization. The individual is no longer completely bound to a particular office, and the organization no longer needs to be bound to a specific place.

New organizational solutions have been introduced. More organizations and companies are trying out work organized around smaller units and work-places geographically distant from the main work-place. In this connection a central concept is “remote working”. For a number of years this term has been used to denote forms of working which allow flexibility and the absence of geographical restrictions. This applies to single individuals, as well as groups of individuals working at a distance from the main work-place. As well as new forms of organization, another concept which is relevant in this connection is mobility. The new information technology makes it possible to work during a journey, on trains, at stations, in hotels etc. and it gives an increased independence to people whose work naturally involves travel, such as sales staff and repairer personnel.

In the project we are studying the effect of information technology on the formation of new forms of organization, and the conclusions that can be drawn from them. The goal of the project is to gain insight into the interplay between the available techniques and the conditions under which people use these techniques, all seen from an organizational perspective. The motive force leading to these new forms of organization can arise at the level of the individual, the firm, or society as a whole. Positive as well as negative effects can be discerned.

In some of the sub-projects the focus is on the changes in communication that are brought about by remote working, and the long-term effects of these changes. In another sub-project we are studying the economic effects of re-locating divisions of a firm, at the level of the organizational economy of the firm, and the level of the society as a whole. (Kristina Larsen, Anna Moberg, Jan Ollinen, Lars Poignant)

6.2.4 Business Control

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

In one project, financed by NUTEK, special attention is payed to small companies and their need for relevant financial control systems. There is a believe that traditional systems don’t directly help managers of small companies. (Jörgen Andersson, Björn Helander)

In another project Activity-Based Costing is studied in some Swedish companies. (Rolf Larsson)

In a third project there is a discussion relative to a system of reference (Jan Olhager and Birger Rapp, 1985) of trends within modern material and production control. Building from observation of changes within the production function in companies and new technology for MPC systems (Material and Production Control), the project discusses the properties of future MPC systems and expected content of different modules. In progress: “OA in Manufacturing and Planning and Control”.

6.2.5 Computer Simulation for Professional Training and Decision support

Decision-making is an important part of many people’s lives. The main aspects are acquisition and processing of information, which then serve as a basis for decisions. We know that human beings can be very irrational in this process, particularly in new and unexpected situations. One reason is that a decision-maker’s ability to make decisions is related to earlier experience which reflects his or her frame of reference. When new and unexpected situations occur, the individual frequently needs help to improve decisions.

Continuous change is an undeniable feature of the business world. This implies that making decisions is intrinsically more difficult. By decentralization and the introduction of small working group units, more people have to take responsibility and make decisions.

In Eastern Europe, the transformation to a market economy implies totally new types of decisions. Decision-makers are faced with new situations, but do not yet understand basic, market economy relations.

In response, throughout the world, the demand for better professional training is increasing. One part of this project aims at making simulation games that support the training of decision makers before the actual decision-making situation occurs. (Jaime Villegas and Latvia University, Riga) Another part of the project aims at making simulation based decision support systems that support decision makers during their actual decision making process. (Bengt Savén, Thomas Bennet, and Albertas Dvilinskas, Vilnius University)

One of the projects, *Decision Support for Manufacturing Companies,* focuses on the demand for better decision support in Swedish manufacturing industry and the main ways in which these demands can be converted into usable models in a decision support system using discrete event simulation. The project will be based on a survey and case studies. (Bengt Savén)

6.2.6 Accounting and auditing

It was mentioned in the introductory text that a special possibility exists regarding research education to licentiate degree. In cooperation with different accounting firms, 3-5 researchers each year are offered a 3 years programme. Most of their interest are towards accounting and auditing. (Anders Bäckström, Elisabeth Gudinge, Bo Lagerström, Fredrik Nilsson, Anette Stolt)

In the project “*The role of the Swedish auditor*”, partly financed by NUTEK, a historical approach is made. It is now more than 80 years since Sweden got its first chartered accountants. The aim of the project is to describe how the profession has been changed and to understand the present changes. (Camilla Sjöström)

6.3 EIS personnel

Group leadership and administrative staff:

Birger Rapp, Econ. Dr., professor
Rolf Rundfelt, Econ. Dr., adjunct professor
Eva Elfinger, secretary
Eva Johansson, secretary

Employed graduate students:

Jörgen Andersson, B. Sc.
Thomas Bennet, M. Sc.
Stefan Blom, M. Sc.
Anders Bäckström, B. Sc.
Peter Carlsson, B. Sc.
Elisabeth Gudinge, B. Sc.
Björn Helander, B. Sc.
Bo Lagerström, B. Sc.
Kristina Larsen, M. Sc.
Jonas Lind, M. Sc.
Lars Mattsson, M.Sc.
Anna Moberg, B. Sc.
Fredrik Nilsson, B. Sc.
Mehran Noghabai, M. Sc.
Lennart Ohlsén, M. Sc.
Jan Ollinen, M. Sc.
Maria Olsson, B. Sc.

Lars Poignant, B. Sc.
Bengt Savén, Tech. Lic.
Camilla Sjöström, B. Sc.
Anette Stolt, B. Sc.
Jaime Villegas, Ph. D.

Graduate Students:

Tor Berggrund, M. Sc.
Eva Borg, B. Sc.
Thomas Jansson
Rolf Larsson, Ph. Lic.

6.4 Selected publications 1988

6.4.1 Theses (Licentiate)

Savén B.: *Produktionssimulering,* LiTH, Thesis 241 del 1: Inledning ISBN 91-7870-704-8, del 2: Bok ISBN 91524-0944-9.

Larsson R.G.: *Aktivitetsbaserad kalkylering i ett nytt ekonomisystem,* Lic-avhandling nr 298, IDA/EIS, Linköpings Universitet och Tekniska Högskola, 1991.

Noghabai M.: *Värdering av strategiska datorinvesteringar,* Lic-avhandling, IDA/EIS, Linköpings Universitet och Tekniska Högskola, 1993 (forthcoming).

6.4.2 Published

Dvilinskas A.: Management Decision Support in Small Companies, *Third Annual Conference on Computer and Information Science,* Linköping 1992.

Larsen K., Moberg A., Ollinen J.: Informationsteknologi och Nya Organisationlösningar - aspekter på kommunikation, *Third Annual Conference on Computer and Information Science,* Linköping 1992.

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

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

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

Olhager J., Rapp B.: Modern Concepts for Computer-Based Manufacturing Planning and Control Systems, pp 565-576. *Operational Research '90,* Ed. H.E. Bradley, Elsevier Science Publishers B V, Amsterdam, 1990.

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

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

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

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

Rapp B.: Editor (1992) i *Riskbedömning - kunskap om risker (*Risk assessment-Knowledge about Risks, in Swedish). NUTEK.

Rapp B.: ”Informationshantering på individ- och organisationsnivå”. L. Ingelstam, L, Sturesson (eds) *Brus över landet,* 1993 (forthcoming).

Rapp B., Thorstenson A.: *Vem skall ta risken? (Who will bear the Risk, in Swedsih). Studentlitteratur, 1993.*

Savén B.: Factory Simulation - concepts and one approach, *Robotics & Computer Integrated Manufacturing*, Vol. 9, No 4/5, pp 341-345, 1992.

Savén B.: Irrelevance Cost? - towards a better classification of production and inventory control methods. *Third Annual Meeting of the Production and Operations Management Society* (Pom-92), Orlando, Florida, Oct. 18-21, 1992.

Savén B., Villegas J. E.: The Evolution of the User Role in Simulation Studies, *Third Annual conference on Computer and Information Science*, Linköping 1992.

Villegas J., **Sjölin M.:** Industrial Training with Simulation Games, *European Simulation Symposium Proceedings*, Dresden 1992, Germany, p. 555.

Figure 6-1. *Building E++'s interior.*

IISLAB

Lin Padgham
director

7
The Laboratory for Intelligent Information Systems

Default reasoning
Inheritance
Reasoning with time
Object-oriented systems
Databases with History

Some of the members in the laboratory for Intelligent Information Systems.
Back row: Bodil Mattsson-Kihlström, Niclas Wahllöf, Martin Sjölin, Thomas Hall, Michael Jansson.
Front row: Ralph Rönnquist, Lin Padgham, David Partain.

7.1 Overview

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

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

A major part of the work in the group is centered around the **LINCKS** project. The aim in this project is to gradually build up a sophisticated and complex information system, exploring a number of issues such as appropriate representations for knowledge and information, distributed information, integration of expert system services using a shared database of knowledge and information, user interfaces to complex systems, planning and reasoning about actions, default reasoning and other commonsense reasoning. The system also provides a testbed for exploration of research issues.

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

7.2 Members

The following people have been members of the group during all or part of 1991/92:

Laboratory leadership and administrative staff:

Lin Padgham, Ph. D., assistant professor.
Bodil Mattsson-Kihlström, secretary.

Graduate students:

Ralph Rönnquist, Ph. D. May 1992.
Patrick Lambrix, Lic., August 1992.
Michael Jansson, graduate student.

Martin Sjölin, graduate student.
Niclas Wahllöf, graduate student.

Other Members:

David Partain, research engineer.
Leif Running, visiting research engineer from U.S.A. 8/91-8/92.
Michael Ohlsson, programmer, finished July 1992.
Thomas Hall, final project, to be graduate student.
Ying Shu, M. Sc. student.

7.3 LINCKS System

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

The database is multi-user and makes use of a distributed architecture of fileserver(s) and workstations. The database also maintains historical information regarding all of its objects and the commands which have led to changes in these objects.

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

The report 'LINCKS - An Imperative Object Oriented System' [PR87] gives further details regarding differences between our view of object orientation and views that are more commonly propagated based on the message passing paradigm. We argue that we fulfill the essential requirements of object-orientation and that decoupling of types from actions is an important step in obtaining a clean model which is easily manageable for end users.

The bottom layer of the **LINCKS** system consisting of an object-oriented database with some basic support for history maintenance and parallel editing was essentially completed in late 1987. This software was tested, stabilised and used by several external groups for both prototype development and education.

During 1989-1990 there was another intensive implementation effort to realise some of the ideas developed within the project regarding user editing in a broad purpose shared database [TEI90, MP+90]. During 1991-1992 we have been refining and stabilising this layer at the same time as we have been adding some important new functionality.

This layer provides facilities for extracting and editing conceptual objects from the database of interconnected information chunks. Each information chunk may participate in several conceptual objects, used by different users. The information extracted from, and the appearance of an information chunk in the database varies depending on the conceptual object in which the information is used. Information regarding the appearance and structure of objects as well as the location of information parts in the database is itself stored in declarative form in the database. This layer is intended to give a flexible tool for experimenting with issues regarding multiple views and uses of information. It also provides a powerful window oriented user interface to the database, making it possible to start experimenting with the notion of information which is both relatively structured for interpretation by computer software (A.I. programs) and relatively free (e.g. documents) for use by human users.

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

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

We are currently still working with some of the more complex aspects of structured editing which arise when one allows a totally flexible system. We have for example worked with such things as moving a structure from a calendar type of complex object, to a report type of complex object, with minimal change in the underlying database. This functionality is useful when one wants to easily incorporate information found in an object of one type, into an object of another type, with minimal explicit interaction. It also enables the creation of structured objects from flat textual objects. This work requires extensive experimentation, (as well as relatively complex code) in order to develop the system so that it supports the user in appropriate ways.

During the past year we have also added functionality towards making **LINCKS** an *active database* [Sjö92]. The system now "notices" when parallel updating occurs and can notify the user of this. The active database aspect is something which we expect to develop further in the near future.

In the following subsections we describe briefly some aspects of the system which are of research interest. The most significant and well developed of these in the **LINCKS** system is the aspect of a historical database, but the other two - user interface management systems and hypermedia have also resulted in published papers and are thus worthy of mention.

7.3.1 History

History information - that is information regarding the development of objects over time - is information which is very often needed if one is to do even quite simple reasoning. However most database systems contain information regarding only a single point in time, namely the present. There has been a growing interest in database systems which can manage some kind of temporal or versioning information.

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

During the period reported here we have worked with two new aspects of historical information, as well as making some minor revisions with respect to already implemented functionality. Firstly we have investigated the maintenance of versions of complex objects, and have implemented functionality to give correct versioning of shared complex objects. A common approach to this problem is to use static rather than dynamic links in the complex objects requiring versioning. However this sacrifices both efficiency and flexibility. We have implemented an approach where dynamic links are retained in the complex object, whilst static links are computed and stored for an atomic object representing the entire composition and carrying its historical development. This approach also seems promising for eventual schema or type evolution, though this has not been fully explored yet. Versioning of complex objects was developed primarily by Patrick Lambrix.

We have also added a command history to the information maintained in the database. This history maintains information about the relative temporal ordering of user commands, and also about which data was affected by the commands in the command history. It enables the answering of questions such as "was the command which changed object A given before, after, or in parallel to the command which changed object B". We expect to use the command history to aid in reasoning about changes which have happened in the database. The command history was developed primarily by Thomas Hall [Hal92].

Michael Jansson has also been doing work on the use of history or versioning in application systems. He has in particular been studying propagation of changes through versions. This work is expected to result in a licentiate thesis during 1993, [Jan93].

7.3.2 User Interface Management Systems

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

Our system takes this model one step further and provides the mechanisms for declarative specification of the contents of user interface objects (as well as their type). The system then allows for dynamic manipulation of these contents and consequent updating of the underlying database plus propagation to other screen occurrences of that information. This provides a significantly higher level of flexibility than traditional UIMS's and allows for building of more complex interactive systems. This aspect of the system is closely tied to the structure editing capabilities described previously. The article [PL92] gives a description of **LINCKS** as a UIMS using an editor model.

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

7.3.3 Hypermedia

Hypertext, or more generally hypermedia is an area of growing research interest. The basic structure of the **LINCKS** database, combined with the display layer, provides a natural and flexible system for building hypermedia applications. One prototype application [TPH+89]has been developed successfully within LINCKS. The advantages of **LINCKS** compared to other hypertext systems (e.g. Hypercard, HAM, etc.) are that there is an underlying multi-user database and that the **LINCKS** system allows greater flexibility in terms of number and kind of link types and attributes. The number of links, link types and attributes is unlimited within the **LINCKS** system.

During the period reported we have developed the possibility of also having audio-objects within the database.

7.4 Theory

The theoretical work in the lab can be broadly described as falling into two main categories of commonsense reasoning - taxonomic reasoning and reasoning about change. We describe each of these and their relationship to the system work.

7.4.1 Inheritance and Taxonomic Reasoning

The primary work in taxonomic reasoning which we have done is to develop a model and theory for representation of and default reasoning about types [Pad91, Pad92]This model is based on the notion of types being described by sets of features (or properties) which can belong either to a *type core* or to a *type default*. Those features belonging to the core are necessary for all members of the type whereas the features of the default describe the typical member of the type.

The default inheritance theory developed using this model is able to obtain intuitively desirable solutions in a simple manner for many of the classical problem examples from the literature of default reasoning with inheritance [Pad88]. The distinction between necessarily inherited characteristics and characteristics inherited by default is essential for intuitive reasoning about many default inheritance problems. Such a distinction is also critical if one wants a mapping to e.g. default logic. The distinction between strict and default inheritance is now considered standard in the field. However this was not so at the time that we developed this theory, which was developed in parallel to and presented at the same time as that of Horty, Touretzky and Thomason [HT88, Pad88].

One possibly significant aspect of the underlying model developed within the **LINCKS** project is that it maps very well to models from prototype theory in cognitive science [Ros78]. The developed theory also allows for default reasoning about contra-indications, and types probably excluded [lPad89a] which is useful in applications such as pharmaceutical prescription, where contraindications, even if they are only at a default level, are extremely important. A final difference between our approach and that of others is that our approach allows representation of information that a particular subclass is typical for its superclass. This information is expected to be important in such things as learning systems. In addition to the above conceptual advantages we have also developed an essentially linear algorithm for the default reasoning process within the theory [Pad89b].

Recently we have been extending the work on default reasoning with inheritance hierarchies to explore classification reasoning using the same representation of type cores and defaults. Several different approaches to combining classification and default reasoning have been explored. Some serious problems have been found in naive combination of default reasoning and classification, which are similar in nature to shortest path problems in default inheritance. We have developed an approach which avoids these problems for a very simple language. These results are reported in [PN92]. A strict classification system based on KL-ONE style subsumption has been implemented by Ying Shu using the **LINCKS** system [Shu91, Shu92, Shu93].

We have also explored taxonomic reasoning as a basis for a diagnostic system. We have developed a system which represents the knowledge base as a default taxonomy of diseases and then combines classification and default reasoning to do interactive diagnosis within this taxonomy [ZP90, ZP91]. We have recently evaluated this system using an empirical comparison with actual doctors and the system has been found to have a diagnostic accuracy in the same range as doctors. This work has been done in cooperation with ASLAB and the results will be reported fully in Tingting Zhang's Ph. D. thesis to be completed shortly [Zha93].

The work on taxonomic reasoning is integrated with the system work in a number of ways. The initial ideas for the basic taxonomic model arose from system design discussions. System considerations of using the taxonomy for instantiation of new types motivated the representation of types as sets of features. This is uncommon in default inheritance reasoning, but extremely important in classification. Although classification systems (or terminological logics) generally require more expressivity in describing features than we have incorporated, the inclusion of features at all does enable us to more easily reconcile default reasoning and classification than if we had had a more standard representation of unstructured nodes and mixed links (e.g. [HT88]).

7.5 Information System Dynamics

Taking the **LINCKS** system as starting point, Ralph Rönnquist has been studying formal methods for characterising the dynamics of information systems [Rön90]. The working hypothesis is that information systems differ significantly from physical systems, and that therefore it is not obvious that methods developed for physical systems apply for information systems. Some of the more distinct differences are:

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

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

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

This work resulted in a Ralph Rönnquist's Ph. D. thesis [Rön92] in May 1992.

Patrick Lambrix has also used the LITE logic as a formal tool in explorations of the desired changes in composite objects as they are developed over time in an information system. Lambrix has identified two different kinds of part-of connections that can exist in composite objects - strong and weak connections. Using LITE he describes the effect that these connections have on the composite object as its parts are modified over time [Lam92c, Lam92d]. He also defines constraints for correctly maintaining these connections in a database setting. This work resulted in Lambrix' Master's thesis in June 1992 [Lam92b] and licentiate thesis in August 1992 [Lam92a].

We have also started some work in looking at a temporal terminological logic (or language for classification systems) called TLITE, based on the previously described LITE [LR92]. TLITE allows extensions of concepts to be time-dependent in two senses. Firstly individuals belonging to a concept within a particular temporal context: for example John is a car-owner at a particular time t1. Secondly concepts can be defined in terms of temporal development of the objects belonging to the concept: for example a traffic light cycles in colour from green through yellow to red at consecutive times. This temporal behaviour is part of the definition of the traffic light concept.

As with the work in taxonomic reasoning there has been a significant interplay between the theoretical work and the system based implementation work. The development of LITE was motivated directly by a discovery that most temporal logics were unnatural for describing changes in information systems. The logic developed mirrors closely the implemented system, and has also led to refinements and changes in the system. The theoretical exploration of change in composite objects was motivated directly by the need for history of composite objects in the system. The theoretical work has then guided the implementation of system functionality.

The work on reasoning about change has resulted recently in one Ph. D. thesis [Rön92], plus a Masters and a licentiate thesis [Lam92b, Lam92a]. Another licentiate thesis is expected in this area during 1992/93 [Jan93].

7.6 References

The following are publications and papers referenced above. A complete list of IIS-LAB publications can be found in Appendix E.

7.6.1 Theses

[Pad89b] **Lin Padgham**. *Non-Monotonic Inheritance for an Object Oriented Knowledge Base*. Ph. D. thesis no. 213, Linköping University, Sweden, 1989.

[Rön92] **Ralph Rönnquist.** *Theory and Practice of Tense-bound Object References.* Ph. D. thesis no. 270, Linköping University, Sweden, 1992.

[Lam92a] **Patrick Lambrix**. *Aspects of version management of composite objects.* Licenciate thesis no. 328, University of Linköping, Sweden, 1992.

[Lam92b] **Patrick Lambrix**. *Management of historical information of composite objects*. Master thesis no 1, University of Linköping, Sweden, 1992

[Hal92] **Thomas Hall**. *Command history in LINCKS*. Final undergraduate thesis, Linköping University, Sewden, 1992.

7.6.2 Forthcoming Theses

[Zha93] **Tingting Zhang**. *Use of Taxonomic reasoning Techniques in a Diagnosis System* (title not confirmed). Ph. D. thesis, Linköping University, Sweden, forthcoming (1993).

[Jan93] **Michael Jansson**. *Propagation of change in an intelligent information system.* Licentiate thesis, University of Linköping, Sweden, forthcoming 1993.

[Shu93] **Ying Shu.** *Implementing a simplified classifier in LINCKs* (title not confirmed). Master thesis, University of Linköping, Sweden, forthcoming 1992/93.

7.6.3 Internationally Published IISLAB papers

[Lamp2c] **Patrick Lambrix**. Temporal aspects of composite objects. In *Proceedings of the 2nd Golden West conference on Intelligent systems*, pages 97-103, Reno, Nevada, 1992].

[Lam92d] **Patrick Lambrix**. Versioning of stringly connected composite objects. In *Proceedings of the 2nd Golden West Conference on Intelligent systems*, pages 116-125, Reno Nevada, 1992.

[MPÅ+90] **N. Mat, L. Padgham, P. Åberg R. Rönnquist, M. Jansson, T. Teitenberg.** User interface and information management for an object-orinted database. In *Proceedings of Techno-Data*, pages 133-140, Berlin, Germany, 1990.

[Pad88] **Lin Padgham**. A model and representation for type information and its use in reasoning with defaults. In *Proceedings of the 7th National Conference of the American Association for Artiificial Intelligence,* pages 409-414, Saint Paul, MI, 20-26 August 1988. Volume 2.

[Pad89a] **Lin Padgham.** Negative reasoning using inheritance. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence,* page 1086, Detroit, MI, 20-25 August 1989.

[Pad91] **Lin Padgham.** A lattice based model for inheritance reasoning. In M. Lenzerini, D. Nardi, and M. Simi, editors, *Inheritance Hierarchies in Knowledge Representation and Programming Languages*, pages 51-68. John Wiley and Sons, England, 1991.

[Pad92] **Lin Padgham** Defeasible inheritance : A lattice based approach. *Computers and Mathematics with Applications*, 23(6-9):527-541, 1992.

[PL92] **Lin Padgham and Jonas Löwgren**. User interface management for an object oriented database system. In *Proceedings of the First International Conference on Information and Knowledge Management*, Baltimore, Maryland, 1992.

[PR87] **Lin Padgham, Ralph Rönnquist**. LINCKS: An imperative object oriented system. In *Proceedings of the 20th Hawaii International Conference on System Science*, pages 516-522, 1987.

[Rön90] **Ralph Rönnquist**. A logic for propagation based characterisation of process behaviour. In *Proceedings of Fifth International Symposium on Methodologies for Intelligent Systems*, page 297-304, Knoxville, Tennessee, USA, 24-27 October 1990.

[TPH+89] **T. Timpka, L. Padgham, P. Hedblom, S. Wallin, G. Tibblin.** A hypertext knowledge base for primary care: Limed in LINCKS. In *Proceedings of SIGIR'89*. Boston, MA, 1989. ACM Publishers.

[Shu92] **Ying Shu**. Integrating terminological reasoning into an intelligent information system. In *Proceedings of The Second Pacific Rim International Conference on Artificial Intelligence*, 1992

[ZP90] **Tingting Zhang and Lin Padgham**. A diagnosis system using classification. In *Proceedings of Fifth International Symposium on Methodologies for Intelligent Systems*, pages 337-344, Knoxville, Tennessee USA, 24-27 October 1990.

7.6.4 Workshop Papers (IISLAB)

[PN92] **Lin Padgham and Bernhard Nebel.** Combining classification and nonmonotonic inheritance reasoning: A first step. Research report, Department of Computer and Information Science, University of Linköping, Sweden, presented at Workshop on Description Logics AAAI Fall Symposium, Boston, October 1992.

[Shu91] **Ying Shu.** Introducing classic-like functions in LINCKS system. In Terminological Logic Users Workshop Proceedings, Technical University Berlin, Berlin, Germany, December 1991. KIT-Report 95.

[ZP91] **Tingting Zhang and Lin Padgham.** Diagnosis using a default inheritance hierarchy. In ICJAI Workshop on Knowledge Representation in Medicine, 1991.

7.6.5 Internal Reports

[LR92] **Patrick Lambrix and Ralph Rönnquist**. Terminological logic involving time and evolution: A preliminary report. Research report, IISLAB, department of Computer and Information Science, University of Linköping, Sweden, 1992.

[Sjö92] **Martin Sjölin**. Activation of LINCKS - adding an active aspect. In Proceedings of Third Annual IDA Conference on Computer and Information Science, pages 81-87, University of Linköping, Sweden, 1992. Department of Computer and Information Science.

[Tei90] **Tim Teitenberg.** Preliminary Documentation of the Application Interface Manager (AIM) for the LINCKS System. Department of Computer and Information Science, University of Linköping, 1990. System Documentation 34.1.

7.6.6 Other References (non-IISLAB)

[HT88] **John F. Horty and Rich H. Thomason.** Mixing strict and defeasible inheritance. In *Proceedings of the 7th National Conference of the American Association for Artificial Intelligence,* pages 427-432 Saint Paul, MI, 20-26 August 1988. Volume 2.

[Ros78] **E. Rosch.** Principles of categorization. In L. Loyd E. Rosch, editor, *Cognition and Categorization*. Erlbaum, Hillsdale, NJ, 1978.

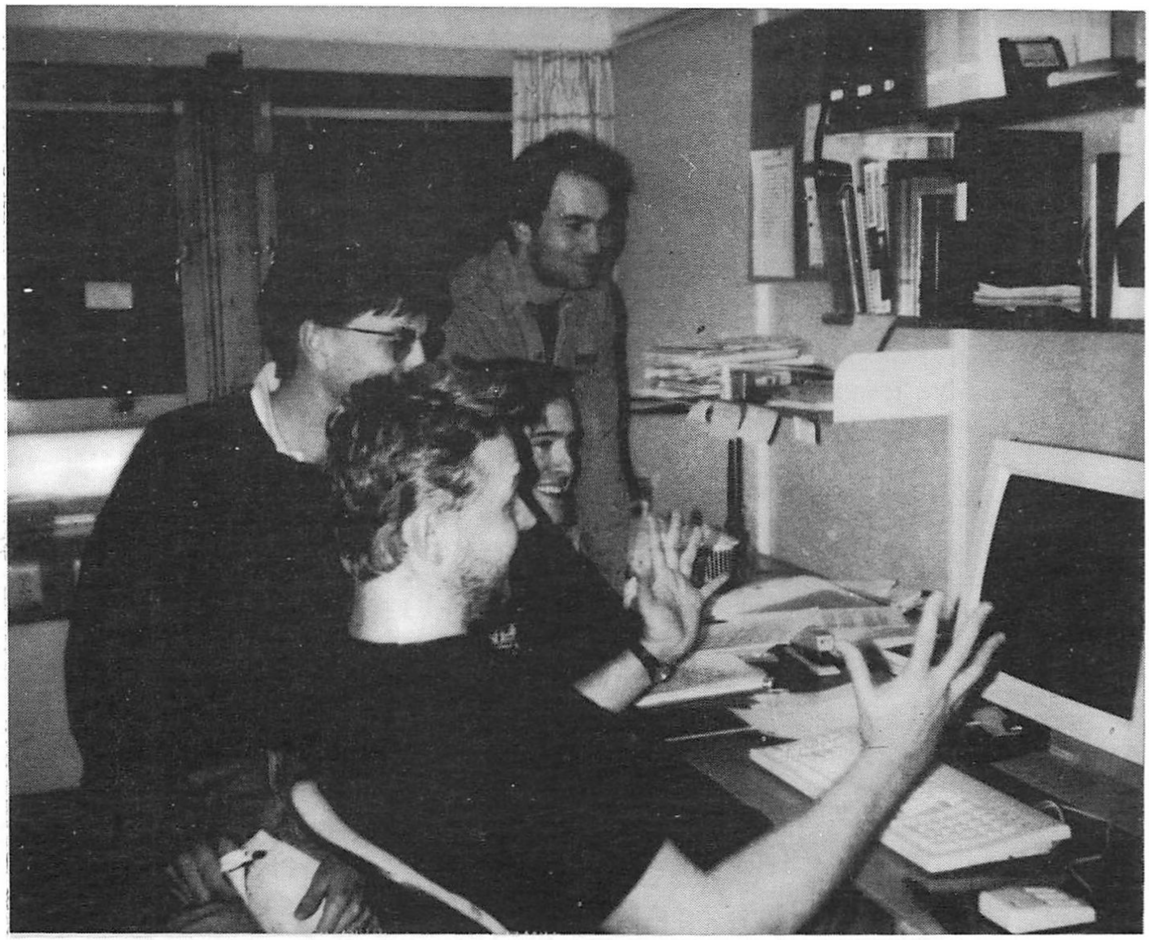
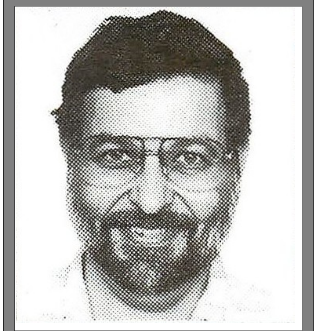


Figure 7-1. *The Third International Programming Contest 1992. The most successful team from Linköping was the“Gamlingarna” (”Oldies”) team from IDA, made up of Bernt Nilsson, Martin Sjölin, Ralph Rönnquist and David Partain, who were 4th in division 3, and 7th in the whole contest.*

LIBLAB

8
The Laboratory for Library and Information Science



Roland Hjerppe
director

User interfaces
Document description
Knowledge organization
Hypermedia
Information management

Some of the members in the laboratory for Library and Information Science.
From the left: Lisbeth Björklund, Andreas Björklind, Roland Hjerppe, Jonas Persson.

8.1 Introduction

Research at the Laboratory for Library and Information Science - LIBLAB, is focused on long term studies of the interactions, positive and negative, between information technology and the generation, access and use of documents and document collections.

Within this broad area the main object of study is the problem of designing and using catalogues - tools for access to large collections of documents. The application domain within which this research has been carried out is libraries.

One of the effects of early information technology, writing, and later printing, was a proliferation of texts. Libraries have for a very long time been one of the two main social responses to the cumulation of writings, a device for providing access to publications – experiences, ideas and knowledge documented in text and made available to the public. Libraries are hence quickly and deeply influenced by any change in information technology that impinges on the creation, distribution and use of publications. New media, new forms of publications, and new methods of scientific communication and knowledge organization, and their interaction with library functions are hence of primary interest to LIBLAB.

The other main social response to the proliferation of texts is archives. One of the main differences between libraries and archives is in the type of texts they collect and organize. Whereas libraries are mainly concerned with publications - texts that are intended for the public consequently have usually been produced in multiple copies - archives are mainly concerned with records - texts that provide evidence of actions, e.g. administrative or commercial or justiciable, and that mostly exist in one or a few copies. The creation, use and storage of such records is increasingly computer-based which raises questions similar to those in libraries when access is considered.

The collections of libraries and archives are an important part of the total cultural heritage of mankind but societies can survive without them. The concept of a document can be broadened to encompass "*that which serves to show or prove something*" or "*something written, inscribed, etc.,which furnishes evidence or information upon any subject, as a manuscript, titledeed, coin etc*." (The Shorter Oxford English Dictionary on Historical principles 3rd. Ed.). Artefacts, small and large, as well as processes (e.g. customs and procedures) and structures (e.g. of organizations) are also important carriers of culture and can therefore be regarded as documents that carry a "text" that, however, is not as easily "read" as a traditional writing. The information that can be deciphered by a trained "reader" is usually transcribed and documented in "ordinary" documents, cf. the notes, photos, sketches etc. produced by an ethnographic researcher during a field study.

Museums are the social institutions that for artefacts (and *objets trouvées*) have the same functions as libraries and archives have for texts. For artefacts (e.g. buildings or environments) that because of their nature (size etc.) cannot be collected and organized in one place there are usually also special national heritage institutions.

The trend towards the use of powerful workstations by knowledge workers implies that catalogues are one kind of tool (that has wide applicability) among many other kinds. They should therefore be designed considering both the personal information management situation of the individual user and the characteristics of the collection and its items. The scope of the research at LIBLAB has thus in recent years been broadened to cover implications for archives, museums and cultural heritage as well as libraries. Technical documentation is another area that also is receiving attention, mainly because it seems to be the domain in which formalism for document description and architecture are having an impact.

8.1.1 LIBLAB's Research program

Within the broad area of study - catalogues as tools for access to large collections - there are two main themes:

- users and information access systems, and
- document description and representation.

In the first theme the focus is on the use and users of interactive information systems, emphasizing users as both producers and consumers of information and information services. This theme has two subthemes:

a) user participation and user behaviour, and

b) orientation in databases: maps and other tools.

The second theme is concerned with descriptions and representations of documents (of all kinds as indicated in the introduction) and collections and their relations at different levels. The context of these descriptions and their representations is computerized catalogues as tools for access to documents. Within this theme we have four subthemes:

c) the convergence of hypertext and multimedia, Hypermedia,

d) HyperCatalogs, and

e) formalisms for document description within documents and catalogues, and

f) document description and catalogues for libraries, archives, museums and national heritage.

8.1.2 The HYPERCATalog

The HYPERCATalog is the designation for our vision of an expanded, extended and enhanced catalogue. It is a vision, or a metaphor for a complex yet flexible way of giving users access to the information and the structures that are more or less implicit in document collections, both their own and other internal and external collections. Our vision is of an information system that is an integrated part of a familiar working environment, a tool for information management rather than merely information retrieval. It is a vision which guides all the various research efforts of LIBLAB and which is itself influenced by the results achieved.

For a number of years the work on the HYPERCATalog consisted of explorations of various aspects of the central vision, with an emphasis on the perceived potentials and problems of hypertext and hypermedia as a complement to traditional retrieval in library catalogues. Early in 1991 work was initiated on a small hypercatalogue, TemaKat, for a limited number of users and their common, small collection.

8.2 Laboratory Members

Laboratory leadership and administrative staff:

Roland Hjerppe, M. Sc., researcher
Barbara Ekman, secretary

Graduate students:

Andreas Björklind, B. Sc., doctoral student since 1991
Lisbeth Björklund, B. Sc, doctoral student since 1985.
Jonas Persson, M. Sc. in Computer Science, doctoral student since 1991
Arja Vainio-Larsson, M.A..,doctoral student since 1984.

Visitor

Rosalind Johnson, student from the Department of Information Studies at the University of Sheffield, visitor for six months, June - November 1991, to do a project resulting in a M.A. dissertation, with the support of the Commission of the European Communities within the framework of the COMETT Programme.

Former member:

Birgitta Olander, Doctoral degree awarded by the Faculty of Library and Information Science, University of Toronto, Canada, spring 1992.

8.3 Progress Report and Current Research

The research activities since 1989/90 are described briefly, to provide a background to present activities as well as continuity for readers of previous reports.

In 1990 FRN, the main funding agency for LIBLAB, decided that the program orientation of the funding for LIBLAB (and INFORSK in Umeå) was to be replaced by a project orientation. As a consequence of this the long term work on the HYPER-CATalog had to be replaced by the more specific short term projects, which are described below.

8.3.1 Cooperation

In 1990 formal long-term co-operation was initiated with the National Defence Research Establishment, FOA, in the area of Geographical Information Systems, GIS, with special attention to information handling in crisis/emergency management systems.

In 1990 co-operation was likewise established with Nordic Design Consultants Inc., NDC, and with Prof. Ching-chih Chen from Graduate School of Library and Information Science of Simmons College, Boston, on the development of multimedia systems for use in museums and a feasibility study on joint development was carried out.

LIBLAB also participated in the planning and program of the 1990, 1991 and 1992 conferences on multimedia in Linköping.

Roland Hjerppe has since 1990 participated in Working Group 3, Remote Collaboration based on Multimedia Workstations, of the COST 14 initiative on Co-operation Technology - CoTech.

8.3.2 Research areas

Past (since the previous annual report) and current research at LIBLAB has for the purposes of this report been grouped into four areas:

- Virtual Reality/Cyberspace & CSCW
- TemaKat
- Description, and
- User Interfaces

that only in part coincides with the categorization of the research programme.

8.3.2.1 Cyberspace, GIS & Cooperation

The subtheme “Orientation in databases: maps and other tools” which partly emanates from the hypertext approach to the use of catalogues has been pursued by exploring, mainly by theorizing, the potential and problems of virtual realities and cyberspace as a tool for exploration of databases.

The same subtheme was also one of the reasons why the cooperation centering on Geographical Information Systems, GIS, was initiated with FOA. The other reason was that geographical location is an important aspect of the items in archives, museums and national heritage. The research in GIS is for the moment concentrating on two areas

- generalization, which can be seen as a kind of abstracting, and
- description of documents with dynamic spatio-temporal objects, using SGML and HyTime.

In an exploration of design for information handling by crisis/emergency management teams the cyberspace studies have been joined with the GIS studies. Braided Reality, a specific solution found and proposed to one of the problems of Virtual Reality - combining being and perceiving in actual natural reality with being and perceiving in a virtual reality - is in this design exploration also being examined as a means for Computer Supported Cooperative Work. The means and modes of reactive planning, seen as an information management problem with the transformations to symbolic form of multimedia input by a team of information handlers, are the main interest.

The implications of what might be called the virtualization of libraries, shifting the emphasis from the acquisition of physical objects to the provision of access to electronic forms, especially the concomitant problems of overview, navigation and use of network resources have also been the subject of investigation.

8.3.2.2 TemaKat

TemaKat is at present the major research undertaking of LIBLAB. The goal is to provide two of the themes, Communication, and Technology and Social Change, of the department of Themes with a catalogue that provides those of the functionalities envisaged for the HYPERCATalog that are feasible, given the constraints of equipment available at the themes, and the resources available at LIBLAB. In addition TemaKat is used as a platform for development of methodology, both with regard to systems analysis and design, and specific parts such as qualitative studies of users. TemaKat has been planned to consist of traditional phases:

- analysis of present user situation and formulation of requirements,
- design, construction and implementation,
- introduction, use and evaluation.

In the first phase, which largely is finished, most of the prospective users of TemaKat have been interviewed extensively with regard to their reading and writing habits, their use of catalogues, libraries and literature.

As a methodological development for the qualitative study of user needs, as expressed in transcribed interviews, an SGML Document Type Declaration has been designed for the transcriptions. The resulting documents are being used as databases in the compilation of the results from the user studies for a qualitative analysis and synthesis of requirements.

In the second, current, phase the findings from the first phase and the ideas and results from earlier HYPERCATalog experiments are being merged, and tools for development and construction are evaluated, as a precursor to actual development.

In the third phase a TemaKat will be introduced and its uses will be studied as a part of an evaluation. The aim is not to evaluate primarily the TemaKat as a system but rather some of the basic beliefs about the uses of bibliographic information that the HYPERCATalog vision is based on.

A basic technical assumption for TemaKat is that a client-server approach can be used but some of the implementation issues that are unsolved are who and how to provide the basic networking capabilities, and the responsibility for maintenance after the first introduction and use.

8.3.2.3 Description of documents

In the area of description of documents there are two directions for the work at LIBLAB. One is the application of SGML and lately HyTime, and the other is the study of descriptions in neighbouring domains, e.g. archives, museums and heritage.

8.3.2.4 User Interfaces

In the area of user interfaces the theoretical and empirical studies have been devoted to emphasizing the close relations between design and evaluation, to relating human computer interaction to tool use, to developing concepts and methods for the evaluation of interfaces from the point of view of their usability for the user, their functionality for the functions to be performed, and the applicability to the task at hand.

8.4 Towards an Undergraduate Education in Information: Resources and Access and in Library and Information Science

During the spring term of 1992 a 10p. undergraduate course in "Electronic media: technology, uses and consequences" was offered for the first time. It was followed in the fall of 1992 by the 10p. course "Information Resources of Society".

These two are the first courses in a planned programme of education in "Information: Resources and Access" that will be developed over the next years. In addition there will be two more courses offered, "Information Needs and Uses" and "Knowledge Organization: Methods and Systems" in the years to come. Each of the four courses will be extended with a 11-20p. follow-up so that the programme in "Information: Resources and Access" eventually will provide studies up to 80p. The "Information: Resources and Access" programme is aimed at students who are interested in being professional information users, in a variety of domains ranging from systems analysis through communication and mass media to studies of culture.

In parallel a proposal has been submitted for a programme of education in Library and Information Science, with an emphasis on mediation of documents and information, that would also be applicable in e.g. museums and archives, and that would also be broadened to cover the full publication chain: generation, publishing, distribution, acquisition, second-hand distribution. The aim is also to give the Library and Information Science programme an international orientation, and to accept students from abroad, e.g. by having English as the course language.

8.5 Publications

External refereed publications:

Vainio-Larsson, A.: As We May Learn: Training With Multimedia. In: *Proceedings of the IFIP Conference on Human Jobs and Computer Interfaces, Tampere, Finland, 24-28 June 1991. Edited by: P. Järvinen, G. Weir and M. Nurminen.*

Vainio-Larsson. A.: Training Within an Interactive Multimedia Environment. In: *People and Computers, HCI 91, Usability Now, Edinburgh, Scotland, 20-23 Aug., 1991, pp.*

Larsson, T. and Vainio-Larsson, A.: Software Producers as Software Users. In: *NATO-ARW: User-Centered Requirements for Software Engineering Environments, Chateau de Bonas, Toulouse, France, 5-10 Sept., 1991 (In print*)

Hjerppe, R.; Björklund, L.; Björklind, A.: Framtidens bibliotek - reella och virtuella. In: *Biblioteken och framtiden. Bok 2. Nordisk idédebatt. Konferens i Borås 11-13 November. 1991. Red. Lars Seldén. Bibliotekshögskolan, Borås, 1992. pp 272-290.*

Björklind, A.: Wide Area Information Server — a Prototype for the Personal Information Tool of the Future?. In: *"Teknologi och Kompetens" Proc. of the 8th. Nordic Conference for Information and Documentation, Helsingborg, Sweden,May 19-21, 1992. Adler et. al. Ed. TLS,Stockholm, 1992 p.105-110.*

Björklind, A.: Document architecture for public service case processing and archival search. In: *"Teknologi och Kompetens" Proc. of the 8th. Nordic Conference for Information and Documentation, Helsingborg, Sweden, May 19-21, 1992. Adler et. al. Ed. TLS,Stockholm, 1992 p.65-70.*

Björklund, L.: Document description in the future.. In: *"Teknologi och Kompetens" Proc. of the 8th. Nordic Conference for Information and Documentation, Helsingborg, Sweden,May 19-21, 1992. Adler et. al. Ed. TLS,Stockholm, 1992 p.71-74.* (Också utgiven som WP: 79)

Hjerppe, R.: Virtuella verkligheter och cyberrymder: Implikationer för informationsmanipulering. Presentation at the *8th. Nordic Conference for Information and Documentation, Helsingborg, Sweden, May 19-21, 1992*

Hjerppe, R.: Database visualization: Asking "What do you have?" instead of answering "What do you want?". In: *"Teknologi och Kompetens" Proc. of the 8th. Nordic Conference for Information and Documentation, Helsingborg, Sweden, May 19-21, 1992. Adler et. al. Ed. TLS,Stockholm, 1992 p.89-92.*

Persson, J.; Jungert, E.: Generation of multi-resolution maps from run-length-encoded data. Accepted for publication in: *International Journal of Geographical Information Systems*

Björklind, A; Björklund, L; Hjerppe, R.: Libraries of the Future - Real and Virtual. Paper presented at *15th International Essen Symposium, Essen Oct. 12-15*

Björklund, Lisbeth: The need for inventing user needs. Paper presented at the *Anglo-Nordic Seminar 1992: The Effects of Digitisation on Library and Information Services, Edinburgh Sept.17-20*

Internal publications:

LiU-LIBLAB-WP:51 **Bergman, D.; Hjerppe, R.; Chen, C-c:** Feasibility Study: Joint Development of Multimedia Systems on the Basis of DOCIM, Project Emperor_I and HYPERCATalog. *Dec. 1990, 7p.*

LiU-LIBLAB-WP:52 **Hjerppe, R.:** LIBLAB- Current Research. *Jan. 1991, 4p.*

LiU-LIBLAB-WP:53 **Hjerppe, R.; Jungert, E.:** Virtual Environments: Braided Reality Information Systems and their application in collaborative contingency planning. *Jan. 1991, 8p.*

LiU-LIBLAB-WP:54 **Hjerppe, R.; Jungert, E.:** Braided Reality Information Systems as Interaction Model for Collaborative work. *Jan. 1991, 3p.*

LiU-LIBLAB-WP:55 **Hjerppe, R.:** Beskrivning av dokument och samlingar i bibliotek, arkiv och muséer. *Febr. 1991, 4p.*

*LiU-LIBLAB-WP:5*6 **Björklind, A.; Björklund, L.; Hjerppe, R.:** Ärenden som dokumentsamlingar - en förstudie. *Febr. 1991, 4p.*

LiU-LIBLAB-WP:57 **Hjerppe, R.:** TemaKAT: En hyperkatalog för Tema-biblioteket. *Febr. 1991, 7p.*

LiU-LIBLAB-WP:58 **Björklund, L.; Kristiansson, G.:** Funktionsorienterad informationssökning i arkivmiljö. *Febr. 1991, 4 p.*

LiU-LIBLAB-WP:59 **Hjerppe, R.; Olsson L.:** Några utgångspunkter för en förnyad bibliotekarieutbildning. *April 1991, 8p.*

LiU-LIBLAB-WP:60 **Hjerppe, R.:** Tema/IDAKat: En hyperkatalog för Tema/IDA. *May 1991, 7p.*

LiU-LIBLAB-WP:61 **Vainio-Larsson, A.:** Datorbruk med förnuft. Seminarium vid NUK-91, Lillehammer. *April, 1991, 10p.*

LiU-LIBLAB-WP:62 **Hjerppe, R.; Björklund, L. & Björklind, A.:** Framtidens bibliotek - reella och virtuella. *Oct. 1991, 13p.*

LiU-LIBLAB-WP:63 **Hjerppe, R. & Björklund, L.:** Elektroniska media: teknik, användning, konsekvenser. Kursbeskrivning. *Oct. 1991. 4p.*

LiU-LIBLAB-WP:64 **Hjerppe, R.; Björklund, L. & Björklind, A.:** Samhällets informationsresurser. Ett kursförslag. *Oct. 1991. 3p.*

LiU-LIBLAB-WP:65 **Hjerppe, R:** TemaNät - Ett lokalt nät för Tema. *Jan. 1992, 6p.*

LiU-LIBLAB-WP:66 **Hjerppe, R. & Jungert J.:** Reactive Planning and Visualization for Location Related Time-Critical Tasks. *Jan. 1992, 6p.*

LiU-LIBLAB-WP:67 **Hjerppe, R. & Jungert J.:** Interactive Manipulation of Symbolic Multimedia Data in a Loosely Coupled Proximate Group. *Jan. 1992, 7p.*

LiU-LIBLAB-WP:68 **Hjerppe, R:** LIBLAB - Current Research. *Jan. 1992, 5p.*

LiU-LIBLAB-WP:69 **Hjerppe, R:** Från informationsbeteende till katalogutformning: Utgångspunkter för revision av ett intervjuformulär. (Oavslutad) *Febr. 1992, 16p.*

LiU-LIBLAB-WP:70 **Hjerppe, R:** Datorbaserade sambrukssystem i bibliotek: en förstudie. *Febr. 1992, 4p.*

LiU-LIBLAB-WP:71 **Björklund, L:** Samhällets informationsresurser - en informationsteoretisk ram för en deskriptiv studie. *Febr. 1992, 4p.*

LiU-LIBLAB-WP:72 **Björklund, L:** Informationsbeteende och kataloganvändning hos tvärvetenskapare - en metod och fallstudie. *Febr. 1992, 4p.*

LiU-LIBLAB-WP:73 **Hjerppe, R:** Från intervjuer till systemutformning och -konstruktion. TemaKat fas 2. *Febr. 1992, 7p.*
LiU-LIBLAB-WP:74 **Björklind, A:** Virtuella bibliotek. *Febr. 1992, 3p.*
LiU-LIBLAB-WP:75 **Hjerppe, R, & Björklind, A:** Ömsesidig anpassning av dokumenttypsdeklarationer och katalogiseringsregler. *Febr. 1992, 3p.*
LiU-LIBLAB-WP:76 **Hjerppe, R:** Utformning av nya beskrivningmodeller för bibliotek, arkiv, museer. *Febr. 1992, 3p.*
LiU-LIBLAB-WP:77 **Hjerppe, R:** Elektroniska industriella dokument: Generering och hantering samt medium, form och innehåll. Två samordnade projektförslag för CENIIT. *Febr. 1992, 5p.*
LiU-LIBLAB-WP:78 **Björklund, L:** Automatic abstracting and subject analysis. *March 1992, 7p.*
LiU-LIBLAB-WP:79 **Björklund, L:** Document Description in the Future. *March 1992, 5p* (See: External publications: Björklund, L., 1992)
LiU-LIBLAB-WP:80 **Hjerppe, R. & Björklund, L:** Samhällets Informationsresurser. Kursbeskrivning. *Aug. 1992. 4p.*
LiU-LIBLAB-WP:81 **Björklund, L:** Studier av användarbehov vid systemutformning: utdragning, upptäckt, uppfinning? *Sept. 1992, 4p.*

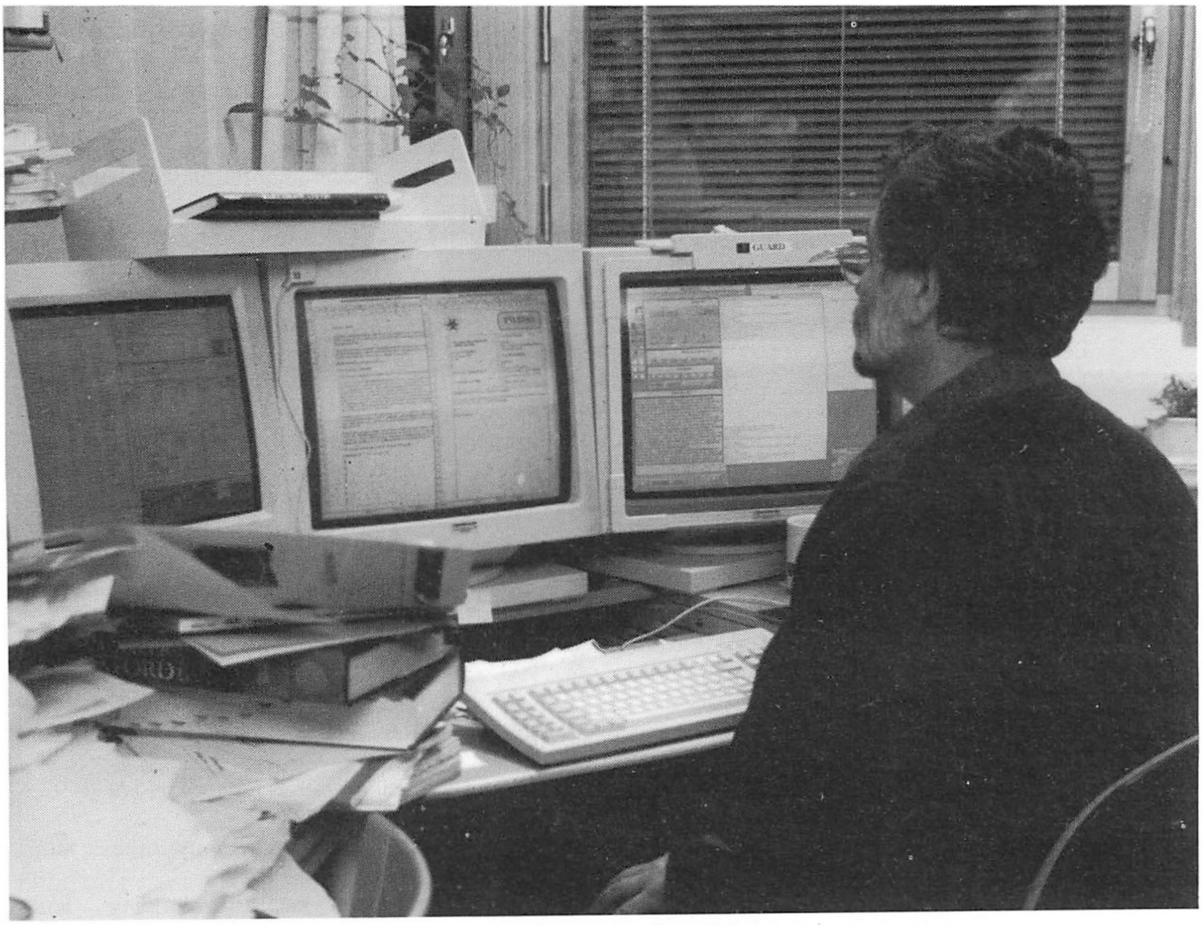


Figure 8-1. *Sun workstation equipped with three monitors.*

LOGPRO

9
The Laboratory for Logic Programming

Foundations of logic programming
Integration of programming paradigms
Program transformation

Jan Maluszynski
director

Some of the members in the laboratory for Logic Programming.
Back row: Anders Haraldsson, Jan Maluszynski, Johan Boye, Magnus Andersson.
Front row: Jonas Wallgren, Roland Bol, Gunilla Lingenhult, Ulf Nilsson, Andreas Kågedal, Jukka Paakki, Hans Olsén.

9.1 Introduction

The research of the Laboratory for Logic Programming concentrates on the foundations of logic programming systems and on the relation between logic programming and other computational paradigms. In the reporting period the work was organized in the research projects “Logic Programs with External Procedures” and “Non-monotonic Theories and Logic Programming”. The latter has been a joint project with the Laboratory for Knowledge Representation of Knowledge in Logic and with Halina Przymusinska and Teodor Przymusinski at the University of California, Riverside. The research of LOGPRO has been supported by the Swedish Board for Technical Development (STU) until 1991, by the Swedish Natural Science Research Council (NFR)and by the Swedish Research Council for Engineering Sciences (TFR).

The following persons participated in LOGPRO activities during the academic year 1991/92.

Laboratory leadership, Administrative staff:

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

Senior researchers:

Wlodzimierz Drabent, Ph. D. lecturer
Anders Haraldsson, Ph. D. associate professor

Visiting senior researchers:

Laurent Fribourg, Ph. D., Ecole Normale Superieur, Paris
Feliks Kluzniak, Ph. D., Warsaw University, Warsaw
Halina Przymusinska, Ph. D., University of California, Riverside
Teodor Przymusinski, Ph. D., University of California, Riverside

Post-doctoral Visiting Scholars:

Roland Bol, Ph. D., Dutch post-doc

Graduate students:

Johan Boye
Staffan Bonnier, (Ph. D., May 92)
Andreas Kågedal, (licentiate, June 1992)
Ulf Nilsson, (Ph. D., March 1992)
Jonas Wallgren, graduate student

A total of 13 papers by permanent LOGPRO-members were published, or accepted for publication in international journals, international conferences or as book chapters during the report period, as listed in this report. Two Ph. D. theses and one licentiate thesis were defended during the spring 1992.

9.2 The objectives and the results

LOGPRO research can generally be characterized as the study of relations between logic programming and other areas of computer science.The motivation is to facilitate transfer of concepts, methods and techniques between these areas and to contribute to a unified view of different formalisms. It has been known for a long time that there are many such connections.With limited resources our research focused on the two research projects mentioned above. Below we survey the results obtained during the reporting period.

9.2.1 Logic programs with external procedures

Semantic foundations

There are many Prolog systems that allow the usage of library procedures written in other languages. However, unrestricted use of such “external” procedures in a logic program blurs the declarative reading of the program. The objective of this research has been to define a clean “declarative” integration of logic programs with external procedures, i.e. an integration where the resulting program has a clean declarative semantics. The approach proposed in our previous publications is based on equational logic programming. For its more recent presentation see [Bon91, Boy91]. Its extension developed during the last year combines concepts originating from term rewriting with the notion of polymorphic types, as used in ML. The approach can be summarized as follows:

- The functional component of the language is not fixed. In principle, functions can be written in any language. For communication with the logic programming component the arguments and the results of functional computations are represented by terms. Seen from the logic programming component the functions have no side-effects.
- The declarative semantics of the integrated programs is obtained by viewing them as equational logic programs.
- The operational semantics is based on an incomplete equational unification, called *S*-unification (as in Structure-oriented).

During the reporting period the following new results have been obtained:

- An extension of the approach to cater for polymorphically typed external procedures [Bon92].
- A condition for static type checking of such programs [BW92].
- Techniques for implementing an interface between Horn clauses and ML [LMO92]

A prototype language based on S-unification

We defined and implemented a prototype language, GAPLog, incorporating our idea of logic programs with external procedures. In contrast to the common approach aiming at integration of logic and functional programming, GAPLog is an “open”language: it includes no mechanism for defining the functional procedures. The implementation is a compiler from GAPLog to SICStus Prolog. It has been developed in a systematic way [KK91, Kåg92] starting from the formal definition of *S*-unification [Bon91] and S-SLD resolution [Boy91].

Informally, *S*-unification is a combination of term unification with evaluation of calls to the external procedures. However, the external procedures can only evaluate calls not containing variables. Formally *S*-unification has been defined by transformation rules for equations. The outcome of *S*-unification is either a failure or a normalized set of equations. *S*-unification subsumes the special case of term unification where the unifier is represented as an equation. Generally, the sets of equations transformed by *S*-unification can be viewed as a kind of constraint in the sense of constraint logic programming.

9.2.2 Static analysis of logic programs

Compilation of a logic program can be optimized if some run-time properties of the program are known in compile time.We investigated two methodologies for static analysis of logic programs and GAPLog programs:

- Abstract interpretation is a method proposed in the seventies in the framework of imperative languages.It was later adopted also for logic programming. During the reporting period we presented the following new results on abstract interpretation:
    - A forrmal framework for constructing and computing abstract interpretations of logic programs with equality [Nil90, Nil92b]
    - An idea of using abstract interpretation for implementation of the deductive data bases. In [Nil91] we relate the magic templates approach to bottom-up evaluation of deductive databases with the abstract interpretation of logic programs proposed by C. Mellish. This facilitates transfer of techniques between these fields and also relates two different approaches to abstract interpretation: the bottom-up approach and the top-down approach.
- Dependency analysis is based on the concept of dependency relation introduced in the formalism of attribute grammars. Our work on relations between attribute grammars and abstract interpretations (see [DM90 , Mal91] for more recent presentation) facilitates the usage of this concept also in the context of logic programs. There have been many recent publications using dependency analysis for proving various properties of logic programs, such as groundness, termination, occur-check, freeness etc. Our result is a sufficient condition for completeness of GAPLog programs [Boy91] obtained by dependency analysis.

9.2.3 Deriving abstract machines by partial deduction

The notion of partial deduction is an adptation of the concept of partial evaluation to the context of logic programming. Recently logic programming has been enriched by a large number of logic programming languages. Since these languages often have a similar basis, but come with different flavours the question arises: Is it possible to single out some common methods for the design of specialized abstract machines for this family of languages? In [Nil90, Nil92b] we have shown a simple set of transformations to derive abstract machine instructions for the control flow of logic programming languages a la Prolog (i.e. depth-first search with backtracking). The major transformation technique used is that of partial deduction but also other (non-standard) program transformations are exploited.

9.2.4 Non-monotonic theories and logic programming

The objective has been to investigate relations between logic programming and non-monotonic formalisms. This concerns in particular:

- Studies of the non-monotonicity introduced to logic programming by negation as failure.
- Defining appropriate declarative and procedural semantics for logic programs with negation and for disjunctive logic programs.
- Defining translation of non-monotonic theories into logic programs.

Below we briefly describe some of our results:

Constructive Negation

A standard approach to negation in logic programming is negation as failure. Its major drawback is that it cannot produce answer substitutions to negated queries. Approaches to overcoming this limitation are called *constructive negation*. We introduce a concept of failed tree for normal programs, as a basis for constructing answers for negative queries [Dra92, Dra91]. This is an extension of our previous approach presented in [MN89]. In the case of finitely failed trees the semantics of our method is given by Clark's completion and our approach is a proper extension of SLDNF-resolution. If extended to infinite failure the method is sound with respect to the *well-founded semantics*.

Semantics of logic programs and non-monotonic formalisms

We investigated various semantics of logic programs with negation and disjunctive logic programs which might be used as a target language for tranlation of non-monotonic theories:

- In [Prz90] we introduced *partial stable model semantics* of normal and disjunctive logic programs, and investigated its properties. This semantics naturally extends the concept of the standard (total) stable model semantics, defined only for some normal (i.e. non-disjunctive) programs. After translation of the program P into a suitable *autoepistemic* (resp. *default*) *theory* $\hat{P}$ the partial stable semantics of P coincides with the 3-valued autoepistemic (resp. default) semantics of $\hat{P}$.
- In [BNN91] we gave simple fixpoint characterization of three-valued stable model semantics.
- [Prz91b, Prz91a] we extended the well-founded semantics, originally defined for normal (non-disjunctive) logic programs, and the partial stable semantics, discussed above, to the class of all disjunctive logic programs and deductive databases. We investigated properties of this new semantics, which we call the *stationary semantics*. This semantics is the only currently known semantics

which is defined for all disjunctive programs and extends both the well-founded semantics of normal programs and the perfect model semantics of stratified disjunctive programs.

- These results should allow us to significantly broaden the class of non-monotonic theories which can be translated (compiled) into logic programs. In particular, it should now be possible to translate non-monotonic theories not only into stratified logic programs, but into arbitrary normal (or even disjunctive) programs. This fact is crucial for the development of a non-monotonic reasoning system which will combine a relatively efficient logic programming engine and a translation unit, allowing translation of non-monotonic theories into logic programs. The paper describing these results is currently under preparation.
- In [Prz92] we introduced a new characterization of the partial stable semantics of logic programs.
- In [PP92] we introduced the class of so called stationary extensions of a default theory. Stationary extensions include, as a special case, Reiter's original default extensions, but allow us to eliminate some of their drawbacks. Any logic program P can be translated into a default theory $\Delta(P)$ so that their respective stationary semantics are preserved, i.e., there is a one-to-one correspondence between stationary (or, equivalently, partial stable) models of P and stationary of $\Delta(P)$. The natural correspondence between logic programs and default theories — preserving their respective semantics — is not only important from a theoretical standpoint but it also allows us to use logic programming as an inference engine for default logic.

9.3 International Cooperation

The international cooperation of LOGPRO includes:

Joint Research Projects:

- Participation in the ESPRIT Basic Research Action 3020 "Integration" completed in March 1992.
- "Non-monotonic Theories and Logic Programming" with Halina Przymusinska and Teodor Przymusinski from the University of California, Riverside

Participation in the ESPRIT Compulog Network of Excellence

Visits of guest researchers at LOGPRO (see above)

Visits of LOGPRO members abroad:

- Wlodek Drabent at the University of California, Riverside April-June 1992 as a visiting lecturer.
- Ulf Nilsson at the State University of New York, Stony Brook as a visiting scholar, September 1991 - September 1992.

- Staffan Bonnier at Ecole Normal Superieur, Paris as a visiting researcher/scholar, November 1991 - December 1992.

Individual cooperation:

- with Pierre Deransart, INRIA, Rocquencourt on relation between logic programming and attribute grammars,a joint book in preparation.
- with Feliks Kluzniak,Warsaw University, on implementation of GAPLog, a joint paper in preparation.
- with Gary Lindstrom, University of Utah, Salt Lake City, a joint paper [LMO92].

Program Committees of the International Conferences:

- Int'l Conference on Logic Programming, Paris 1991, Wlodek Drabent.
- Int'l Workshop on Metaprogramming in Logic Programming, Uppsala 1991, Wlodek Drabent.
- Int'l Symposium on Programming Language Implementation and Logic Programming, Passau, Germany 1991, Jan Maluszynski (co-chair).
- Int'l Conference on Computers and Information, ICCI'92, Toronto, Jan Maluszynski.
- Joint Int'l Conference and Symposium on Logic Programming, Washington D.C. 1992, Jan Maluszynski
- 4th Int'l Symposium on Programming Language Implementation and Logic Programming, Leuven 1992, Wlodek Drabent

9.4 References

Dissertations

[Bon92] **S. Bonnier**. *AFormal Basis for Horn Clause Logic with External PolymorphicFunctions*. Ph. D. thesis, Dep. of Computer and Information Science, Linköping University, May 1992.

[Nil92b] **U. Nilsson**. *Abstract Interpretations & Abstract Machines: Contributions to a Methodology for the Implementation of Logic Programs*. Ph. D. thesis, Dep. of Computer and Information Science, Linköping University, March 1992.

[Kåg92] **A. Kågedal**. *Logic Programming with External Procedures: An Implementation*, Licentiate thesis, Dep. of Computer and Information Science, Linköping University, June 1992.

External Publications

[BNN91] **S. Bonnier, U.Nilsson, and T. Näslund**. A Simple Fixed Point Char acterization of Three-Valued Stable Model Semantics. *Information Processing Letters*, 40:73-78, 1991.

[Bon91] **S. Bonnier**. Unification in incompletely specified theories:a case study. In A. Tarlecki, editor, *Mathematical Foundations of Computer Science* 1991, pages 84-92. Springer-Verlag, 1991.LNCS 520.

[Boy91] **J. Boye**. S-SLD-Resolution: An operational semantics for logic programs with external procedures. In Jan Maluszynski and Martin Wirsing, editors, *Proc. PLILP'91*, pages 383-394. Springer-Verlag, 1991.LNCS 528.

[BW92] **S. Bonnier and J. Wallgren**. Logic Programs with Polymorphic Types: A Conditionfor Static Type Checking. In *Proc. 3rd International Workshop on Conditional Term Rewriting Systems*. Springer-Verlag, LNCS, 1992.To appear.

[DM90] **P. Deransart and J. Maluszynski**. What kind of grammars are logic programs. In P. Saint-Dizier and S. Szpakowicz, editors, *Logic and Logic Grammars for Language Processing*, chapter 2, pages 29-55. Ellis Horwood, 1990.

[Dra91] **W. Drabent**. *Constructive Negation by Fail Answers*. Research Report LiTH-IDA-R-91-23, Department of Computer and Information Science, Linköping University, 1991. Accepted for Acta Informatica.

[Dra92] **W. Drabent**. What is failure? or: Constructive negation by fail answers. In *ICCI Proceedings*. IEEE Computer Society Press, 1992.

[KK91] **A. Kågedal and F. Kluzniak**. Enriching Prolog with *S*-unification. In John Darlington and Roland Dietrich, editors, *PHOENIX Seminar and Workshop on Declarative Programming*, Workshops in Computing, pages 51-65, Sasbachwalden, Germany, 18-22 November 1991. Springer-Verlag.

[LMO92] **G. Lindstrom, J. Maluszynski, and T. Ogi**. Our LIPS Are Sealed: Interfacing Functional and Logic Programming Systems. In Maurice Bruynoogh eand Martin Wirsing, editors, *Proc. PLILP'92*, pages 428-442.Springer-Verlag, 1992. LNCS 631.

[Mal91] **J. Maluszynski**. Attribute grammars and logic programs: A comparison of concepts. In Henk Alblas and Borivoj Melichar, editors, *Attribute Grammars, Applications and Systems*,Prague Summer School, pages 383-394.Springer-Verlag, 1991. LNCS 545.

[MN89] **J. Maluszynski and T. Näslund**. Fail substitutions for negation as failure. In E.L. Lusk and R.A. Overbeek, editors, *Proc. of NACLP*, pages 461-476.MIT Press, Cleveland, Ohio, 1989.

[Nil90] **U. Nilsson**. Systematic semantic approximations of logic programs. In P. Deransart and J. Ma luszynski, editors, *Programming Language Implementation and Logic Programming*,International Workshop PLILP'90, pages 293-306. Springer-Verlag, 1990. LNCS 456.

[Nil91] **U. Nilsson**. Abstract interpretation: A kind of magic. In Jan Maluszynski and Martin Wirsing, editors, *Proc.PLILP'91*, pages 299-310. Springer-Verlag, 1991.Extended version accepted for publication in Theoretical Computer Science.

[Nil92a] **U. Nilsson**. Towards a methodology for the design of abstract machines of logic programming languages. *Journal of Logic Programming*, 1992 (to appear).

[PP92] **H. Przymusinska and T. C. Przymusinski**. Stationary extensions of default theories. In *Proceedings of the Fourth Workshop on Non-Monotonic Reasoning*, Plymouth, Vermont, 1992 (In print).

[Prz90] **T. C. Przymusinski**. Extended stable semantics for normal and disjunctive logic programs. In *Proceedings of the Seventh International Logic Programming Conference*, Jerusalem, Israel, pages 459-477, Cambridge,Mass., 1990. Association for Logic Programming, The MITPress.

[Prz91a] **T. C. Przymusinski**. Semantics of disjunctive logic programs and deductive databases. In C. Delobel, M. Kifer, and Y. Masunaga, editors, *Proceedings of the Second International Conference on Deductive and Object-Oriented Databases DOOD'81*, pages 85-107, Munich, Germany, 1991. Springer-Verlag.

[Prz91b] **T. C. Przymusinski**. Well-founded completions of logic programs. In *Proceedings of the Eigth International Logic Programming Conference*, Paris,France, pages 726-744, Cambridge, Mass., 1991.Association for Logic Programming, The MIT Press.

[Prz92] **T. C. Przymusinski**. Two simple characterizations of well-founded semantics. In I. Havel and V. Koubek, editors, *Proceedings of the International Conference on Mathematical Foundations of Computer Science (MFCS'92)*, pages 451-462, Prague, Czechoslovakia, Springer-Verlag, 1992.

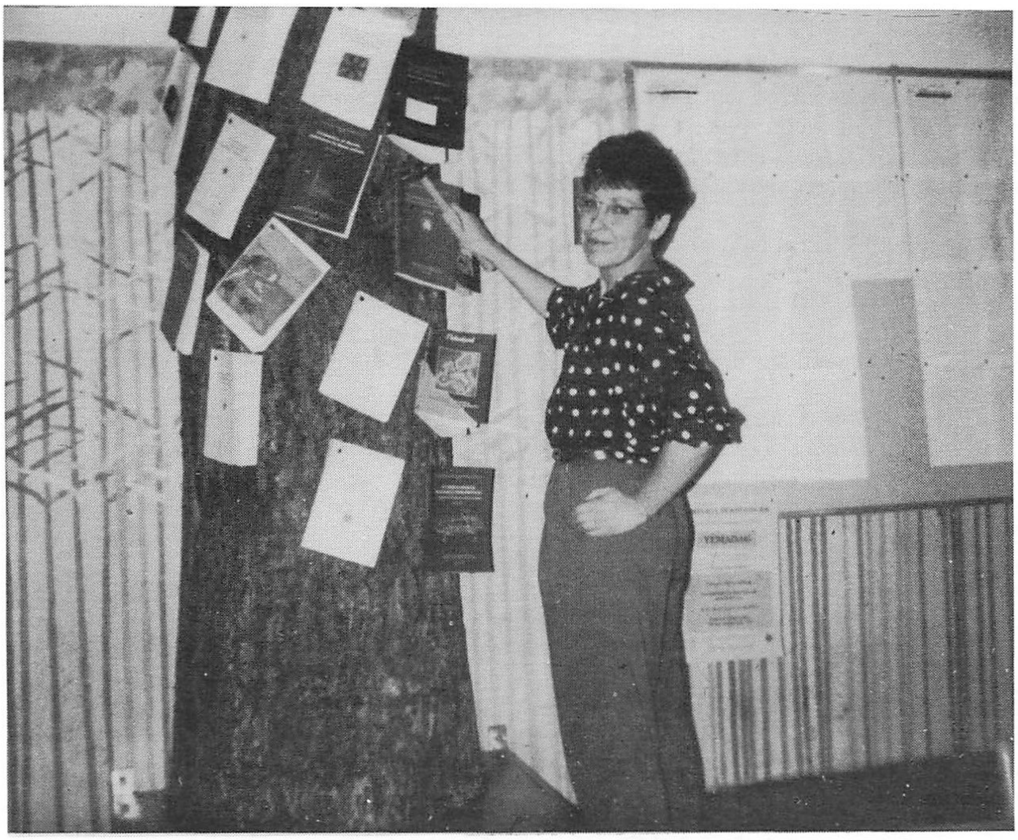
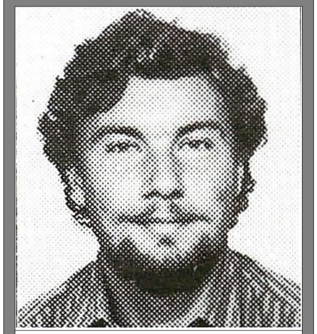


Figure 9-1. *Staffan Bonnier's and Ulf Nilsson's doctoral theses nailed up by the administrators Gunilla Lingenhult and Lillemor Wallgren. .*

MDA

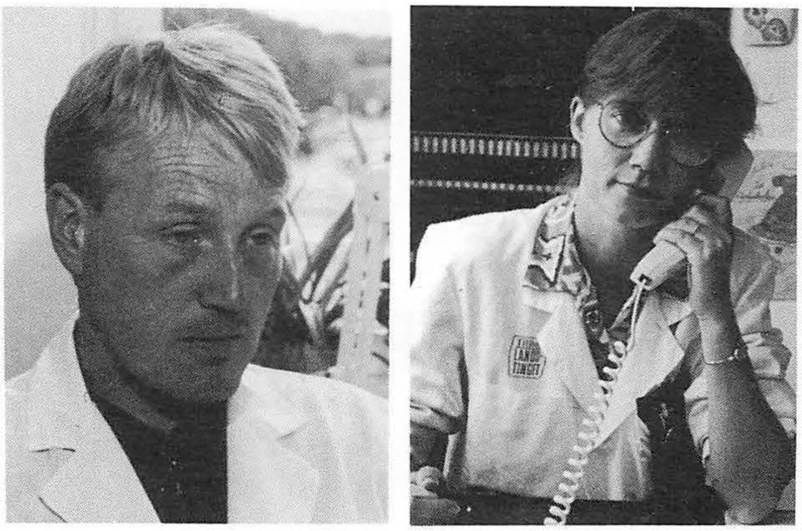


Toomas Timpka
director

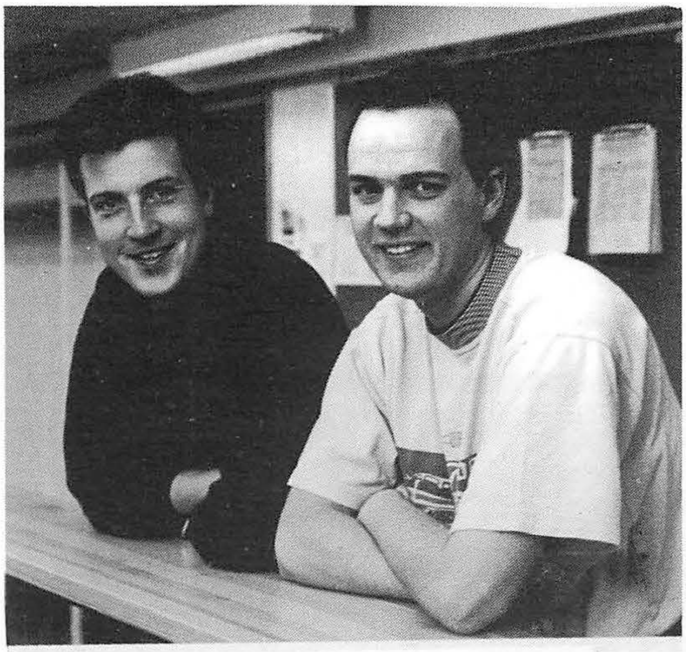
10
People, Computers and Work

Computerization in work life
Hypermedia
Action research
Video studies
Qualitative methods

Some of the members in the MDA group.
Back row: Cecilia Sjöberg, Birgitta Svensson, Ingalill Andersson, Leili Lind, Siv Söderlund, Ewa Rauch.
Front row: Arne Fäldt, Per Hedblom, Tom af Klercker, Toomas Timpka, Mikael Johansson, Patrik Lindblom.



Håkan Karlsson (General practitioner) and Laila Sandström (Nurse), Health care professionals partitioning in the MEDEA project.



Patrik Lindblom (left) and Per Hedblom are developing software for the hypermedia system, and tools for structuring of text to hypertext in the MEDEA project.

Birgitta Svensson (District nurse) is responsible for transforming medical texts into hypermedia used in MEDEA.

Figure 10-1. *Members of the MEDEA project.*

10.1 Introduction

With the point of departure in the concept of *design*, the MDA-group conducts research into *work-life informatics* integrating views from computer science, psychology, and the social organization of work development. This includes studies of computer system design, effective implementation within organizations, maintenance, use, organizational value, and their effects for people and an organization's clients. Work-life informatics as research area entails, e.g.:

- the roles of computerized systems in altering work, group communication, power relationships, and organizational practices.
- how people and organizations use systems in practice;
- how system designers translate people's preferences into requirements;
- the functioning of software development teams in practice; and
- the conditions that foster and impede the implementation of computerized systems within organizations;

Thus, the MDA-group studies the development and use of computerized information systems and communication systems in organizations.

Within the MDA-group, activities at the Department of Computer and Information Science and the Medical Faculty have been coordinated to develop and evaluate experimental information systems. Since the formation of the group in 1988 the actual research has been directed towards four interdisciplinary areas: action research, Computer Supported Cooperative Work (CSCW) and hypermedia, studies of work practices, and the role of video in computer and information science.

10.1.1 Action research

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

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

For this to be performed, equivalence between development, design and organizational action is presumed. Hence, experiences from a variety of development activities are fed into a discussion of information system design methodologies, this within the framework of *action research*.

10.1.2 Computer Supported Cooperative Work and Hypermedia

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

A special issue is the role visual material has or should have in CSCW, and also in hypermedia systems. Reasoning in many areas in work life, for instance in health care and engineering, is based on spatial abstraction of structures, and for this reason issues of visualization need to be brought into the design of these systems to be used for decision support. Integrated images and video links at least gives the *potential* to design computer systems which address issues more conventional communication media can not. However, such a view also implies that development of CSCW systems can not be based only on methods and traditions from computer and information science, but also on knowledge from psychology, social sciences, graphical design, and directly from professionals practicing in the application field. Here, in the MDA-group we have found reflection through action research essential to find the *forms* for such an inter-disciplinary development process.

10.1.3 Studies of work practices

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

10.1.4 Video in Computer Science research

Another of our interests is the use of video as a component and tool in computer and information science research. Video technology can, first, be included in an application, either as a real-time communication facility or as a means of visualization or narration of an episode, movement or sequence of actions. Second, video can be used for documentation in more qualitatively oriented empirical studies to understand human-computer and human interaction. It constitutes a direct and contextually rich means for capturing observable behavior. However, one major problem lies in analysis of the recorded material. Use of qualitative methods is an interesting alternative from our perspective. Third, video can be used directly in the system design process for communication of design alternatives. Design can be seen as a social activity where video is the most efficient mediumin in many situations for documentation and communication of ideas.

10.2 The group

The members of the MDA-group at the Department of Computer Science are (October 1992):

Toomas Timpka, M. D., Ph. D. (director)
James M. Nyce, Ph. D. (co-director)
Hans Holmgren, M. Sc. (co-ordinator to July 1992; on sabbatical 1993)
Leili Lind, R. N. (co-ordinator from August 1992)
Lena Wigh, (secretary to July 1992; on maternity leave 1992)
Siv Söderlund, (secretary from August 1993)
Tom Buur, M. D., Ph. D. (visiting researcher at Leeds University Hospital 1993)
Per Hedblom, M. Sc.
Mikael Johansson, B. Sc.
Tom af Klercker, M. D.
Patrik Lindblom, M. Sc.
Ewa Rauch, M. Sc.
Lars Reshagen, M. Sc.
Birgitta Svensson, R. N.
Cecilia Sjöberg, B. Sc.

Göran Goldkuhl from the VITS-group has been associated with the group since its formation in 1988. At the Department of Community Medicine Per Bjurulf, Elisabeth Arborelius, and Anders Benktsson have been involved in the MDA-group. During 1991 Mikael Peolsson and Hans Vind have completed their Master's Theses in the group. From 1992, the MDA-group is a partner in the EC/AIM (Advanced Informatics in Medicine) consortium A2005 "DILEMMA". The consortium involves partners from England, France, Germany, Portugal, the Netherlands, and is co-ordinated by the Imperial Cancer Research Fund in London. In addition, the MDA-group and Rank Xerox EuroPARC have integrated activities regarding the development and evaluation of the *design rationale* concept for building arguments around decisions taken in a design or change process. During 1992 this has resulted in partnership for the MDA-group in the EC/ESPRIT BRA (Basic Research Action) consortium 7040 "AMODEUS II".

10.3 Funding

Up to July 1992 The MDA-group has mainly been supported by a grant financed on an equal basis by the Swedish Work Environment Fund (AMFO) and the Swedish National Board for Technical Development (NUTEK) through the MDA-program. An equipment grant comprising 8 workstations with colour displays was received from the Swedish Council for research in the Natural Sciences. From 1991 the MDA-group also holds grants from the ITYP-program (NUTEK) and the EC/AIM-program (through NUTEK). Additional funding has during the period been received from the Swedish Telecom and Östergötland County Council.

10.4 Projects

During 1991-92 The MDA-group has been involved in two main projects investigating, respectively: Computer Support for Cooperative Work in Primary Health Care (MEDEA), and development of contextual and participatory software design to be used in work life (Action Design).

10.4.1 Computer Support for Cooperative Work in Primary Health Care

Between 1988 and 1992, the main focus for the MDA-group has been the “Datorstö(r)d” or “MEDEA” project, which has had as its aim to study, and develop computer support for, the work of a primary health care team. The first phase of the project was reported in July 1992 by, in addition to formal reports, production of a video, exhibition of a prototype at a national computer fair (Data Office ’92), and several articles in newspapers and trade journals.

However, project work continues in cooperation with the staff of the primary health care center in Mjölby, situated 40 km west of Linköping. The project is also discussed on regular basis with representatives from the County Council of Östergötland, who are responsible for the provision of health care in Mjölby. During 1992, co-operation has been established with UI-design AB, Linköping, regarding the productification of prototypes developed in the project. Close contacts are maintained with the groups of Professors Werner Schneider and Gösta Tibblin, both at Uppsala University. Project supervisor is Professor Per Bjurulf from the Department of Community Medicine.

10.4.1.1 Results 1991-92

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

10.4.1.1.1 Studies of work practice

To study the adequacy of their professional qualifications, daily dilemmas encountered by the members of interprofessional primary health care teams were investigated (Timpka and Nyce 1992). The entire staff at four primary health care centres (PHCs) were surveyed, and 199 personal interviews were conducted using the Critical Incident Technique. Medical dilemmas were mainly reported by GPs and nurses, organizational dilemmas by laboratory staff, nurses' aides, and secretaries, and dilemmas in the patient-provider relation by nurses, nurses' aides, and secretaries. One dilemma out of four was not solved within the daily routine. GPs appeared from a *work role* point of view to have a priviliged position, since they reported few organizational problems, seldom had to face patient complaints, and usually were able to solve their dilemmas. A difference between the GPs' and nurses' medical dilemmas was that while GPs mainly addressed issues in the care-taking of the patient *in the health care organization*, nurses focused on the relation between the patient's functional health status and the demands put on them *in their day-to-day life*. Organizational and communication dilemmas reported by nurses, nurses' aides, and secretaries often had their cause outside the control of the individual professional. These dilemmas were often "caused" by GPs or other nurses, e.g., by not keeping appointment times or neglecting information.

10.4.1.1.2 Studies of work and technology integration

Building on theoretical stances from activity theory and data from several qualitative studies of work practices in medicine, Timpka and Nyce (1992) argued that the main reason that hypermedia systems are not being used today in health care is that the clinical work they are supposed to support is neither sufficiently understood nor are its principles built into the design process. "Common-sense" guesses in clinical hypermedia design are shown to be inadequate because they can show significant differences from real user needs. More importantly, these "common-sense" guesses by designers could divert a system design process because users' work practices and knowledge are either neglected or misunderstood. To better understand the relation between use and practice, on one hand, and technology on the other, a need for a *pragmatics* of medical hypermedia systems was identified.

Another empirical study *analyzed the possible advantages* of supporting a General Practitioner by *expertext*, i.e., the combined functionality of an expert "alert" system, a hypermedia system, and an on-line library (Timpka, Buur and Hedblom 1992). This was done by letting an expert physician panel simulate an "optimal" expertext system. In the study, *support situations* for the expertext components were defined based on user problem awareness and the availability of an expert consensus. The diagnostic dilemmas regarding female genitourinary infections as presented to the primary care physician were chosen for case study. The results showed that *triangulation* of decision support functions into *expertext* systems is advantageous, if not fundamental, for the usability of computer-based decision support in work life. A hypothetical "optimal" expertext system could provide decision support for almost half of the cases, seen by a physician in daily practice. None of its component parts could provide support for more than every fifth case.

Finally, to consider video phone support, what constitutes "work" in clinical consultations was looked at (Nyce and Timpka 1992). Using several methods (participant-observation, video and interviews), pathologists and surgeons were found to both share and not share similar understandings of what a consultation is, what one should achieve in a consultation, and what in fact constitutes a "successful" consultation.

Further, the same objects of consultation (the products of “offstage” work) was used and defined quite differently depending on how a consultation is framed. These differences and the disjunctions that can arise from them need to be better understood if Computer Supported Cooperative Healthcare Work (CSCHW) applications and environments, such as video phones, are to be adopted and accepted.

10.4.1.1.3 Technical development and implementations

“Gösta’s book” is a collection of hypermedia documents for use in primary care, developed in co-operation with the Department of Family Medicine at Uppsala University (Timpka, Hedblom, and Tibblin 1990). The authoring of this hypermedia collection is a continous process, where three steps are iterated: content authoring, structure editing, and systems implementation. Hence, the development of hypermedia systems for work life does not just rest upon technology. For such a system to be useful and acceptable, issues regarding information quality and volume also need to be addressed. This information today has two critical characteristics. First, it is delivered on a paper-based medium, and, second, it is structured in sequence. Therefore, paper-based information material has to, first, be transferred to an electronic medium, and then converted to a non-sequential hypermedia structure.

Within the MDA-group, a method for transfer of available and relevant clinical texts to “Gösta’s book” to hypermedia has been developed (Sjöberg, Timpka, Nyce et al 1992), including procedures from text and image scanning to screen lay-out. When a book is transferred, a graph is created over its structure, conceptualizing its parts and entry points to the book’s contents and information. This map or graph of its actual structure is then compared to a “target structure”. If there are differences, information is added, restructured, and converted to the target structure. This can be done, for example, by moving text in a file, writing overview nodes or by adding missing parts. Finally the information is divided into elements that can be referred to and displayed as separate hypermedia nodes. We found it important for the professionals that the corpus of information at the task support level has a semantic structure, with semantic relations that mirror the clinical relationships between parts. Moreover, not only does the information material have to be rebuilt to conform to a non-sequential structure, so do the indices. The two are done in parallel. Thus, a controlled vocabulary for description of information elements is continuously built as the corpus increases. It has two purposes: to reduce redundancy by giving elements in the corpus unique names and it makes it possible to locate specific facts in the corpus through new indices adapted to the non-sequential structure. The vocabulary is based on available indices in material brought into the corpus and the vocabulary is added to as new material is transferred and converted.

10.4.2 Action research and design studies

A major problem with close-to-work methods for software design, and with user participation in general, has been *making use* in the actual software development process of the vast amount of empirical data and experiences collected in the first phases of design. Our solution to this problem is to extend the methods to *Action Design* where an interdisciplinary group functions as a reference source for exploration and experimentation at the workplace, and where object-oriented programming environments (OOEs) are used as framework for the software development (Timpka, Hedblom, and Holmgren 1992).

Action Design is a method, or “methodological toolbox”, for system development, inspired by the action research paradigm. Action Design includes detailed recommendations on three levels: perspective, analysis tools (forms of documentation) and project management, all of which should support an action research style of information systems development. However, the main point of Action Design is still that *real* users should reflect and act upon early versions of software while doing *real* work. By using an OOE where functionality and user interface are very tightly coupled it is possible to run applications almost from the start of software development. It is also possible to make rapid changes, even if this entails going deep down in the structure of the application.

10.4.2.1 Results 1991-92

Three main areas have been addressed, extension of Action Design to representation of design arguments, evaluation, and integration in a European context in the “DILEMMA” consortium.

10.4.2.1.1 Argumentative Design as a component of Action Design

Like all system development methods that rely strongly on user participation, Action Design can be criticized for lack of rigor. As a response to this criticism an elaboration within the Action Design framework, Argumentative Design has been introduced (Holmgren, Timpka, Goldkuhl et al 1992). This method makes the argumentation and justification aspects of design discussions explicit through a notational system which is used to represent the design space of a system. This representation, the design rationale for the system, is seen as a co-product of the design process. Action Design supports all aspects of participative system development, but in the main concern is with the later, iterative phases of development. Activity theory is introduced as a theoretical frame of reference for these phases, which are crucial for meeting extreme demands on the performance of information systems in medical settings.

The idea is that in a design process, models of the planned artefact are continuously built. However, one is not always explicitly viewing a design proposal as one of several design options. Still, all design options can be considered as possible answers to a design question. This stance can be used as a heuristic device in the design process: When someone puts forth a design proposal, one can request the underlying design question if it is not explicit already. Thus, if a design proposal is an answer; what is the question (design issue) to this answer? And are there other possible answers? Through such a process one can reconstruct design issues and also stimulate other alternative solutions. What this means is that the design space is made larger and more explicit. To support these processes we need further methodological concepts. Design options are evaluated using criteria.When evaluating several options (which are different answers to a design issue) one should not use a single criterion but several. These different criteria might be in conflict. In the justification process one must take into account these conflicts and try to resolve or manage them in some way. Argumentative Design offers a way to identify and manage these issues.The application of criteria to options means an evaluation. Through this evaluation, consequences and properties of options are justified and stated. With an explicit argumentation and justification process, it is possible to reconstruct, describe, consider and sharpen different design criteria and goals. With Argumentative Design, these evaluation and choice processes are not separated from design but are instead an inherent part of it. What is described here is a method that will make the design process more transparent and thus more negotiable to those involved in it.

10.4.2.1.2 Experiences with Action Design

Action Design is continuously evaluated, recently in the design of the information corpus for "Gösta's book" (Timpka, Nyce, Sjöberg et al 1992). Here an Action Design group was formed of designers (informaticians, software engineers, organizational scientists) and practitioners (GPs, nurses, nurse's aides, secretaries) with at least one participant from each professional category. The participation of the practitioners was divied at two levels: "employed" (one nurse, one GP), meaning daily participation in design activities; and "affiliated" (an entire PHC team), meaning participation mainly in evaluations and task groups. The designers and employed practitioners arranged weekly "design seminars", while the entire Action Design group met on "workshop" basis monthly. Several "corpus mock-ups" were compiled by the designers in the group, and presented for the practitioners. The mock-ups consisted first of paper-based books and documents only, then of prototypes built in HyperCard (TM) and FrameMaker (TM).

Two main lessons were learnt. First, a first Action Design group was formed according to the above organization, but had to be split after a few months because of reorganization at the PHC. This, however, gave a possibility to change in the group structure. The first group had a constant workshop chairman, an informatician. However, this was interpreted by the Action Design group to shift the power during the meeting to the designers within it. Also, the "affiliated" PHC team occasionally had major problems to find time for the workshops, which additionally decreased their influence on the design. Thus, for the second group, the workshops were presented as *explicit negotiations* towards consensus, and *meeting rules* and *agendas* were prepared and discussed in advance. Chairmanship was rotated, so that workshops of evaluative character always were led by practitioners. Moreover, an agreement was reached with the management at the PHC which made it possible for the "affiliated" team to schedule workshop time several weeks ahead.

Second, in the workshops, the practitioners commented on the mock-ups at three levels: the "corpus proper", performance of work tasks, and organization and division of labour. When the corpus mock-up was first seen, severe criticism was formulated by the practitioners on its diverse structure. Thus, after the evaluation of the very first version in HyperCard (TM), a consistent hypermedia structure was defined, and the core of the corpus was made to contain only material having this structure. At the level of work tasks, the nurses in the group formulated a need for integration of images to support the texts. This was found especially beneficial in the telephone counseling situation. Regarding division of work, several texts were deleted from the corpus, because they formulated too detailed prescriptions of which patients were to be seen by the different professional categories. It was a consensus in the PHC team that such rigid guidelines were impossible to follow, and so were redundant.

10.4.2.1.3 CEC/AIM project A2005 “DILEMMA”

The general goal for the consortium has been to develop a technological and organizational foundation for support of primary care practitioners in terms of decision-making and communication with specialist physicians. The MDA-group has the responsibilty in the consortium for requirements analysis and system design methods development. As a basis for this, first, an analysis was performed of the “perspectives” underlying available “basic” methods, which resulted in that considerable differences were identified (Sjöberg 1992a). As a first measure to overcome these differences, the relation between information systems development and organizational change was further investigated (Sjöberg 1992b). These studies were then used in an integration on notational level (Sjöberg and Timpka 1992), where a generic business model (Common Basic Specification (CBS) from NHS/IMC in Birmingham) is used for health care activity description, and Argumentative Design/Contact for problem and goal analyses, and for representing design or change decsions.

Second, methods consultation has been provided, mainly in a two-day course provided to the consortium where the integrated method was presented. In addition, personnel from NHS/IMC and the MDA-group have provided at-site service at all clinical sites in the consortium, and the MDA-group has synthesized their prior experiences of studies of primary care work in a summary document.

Third, different parts of the “Action Design Toolbox” have been exploratively evaluated. Here, first, to become familiar with its properties, the potential of CBS as a tool for investigating and forecasting technology dissimination in health care has been tried out (Timpka, Nyce, Sjöberg et al 1992). Second, the guidelines for formulation of purpose and aim of interdisciplinary action design groups, which are the fundation of Action Design, have been evaluated. An action research study has been performed, in the context of information material management, where several amendments to the guidelines where found necessary (see above). These revised guidelines have been evaluated by a video study of the resulting group process (Karin Willborg, Master’s Thesis, 1992, in preparation).

10.4.3 The “end-user failure”: some reflections over action research and systems development in work life

Even though they might seem to be of only academic interest, some matters have appeared which the MDA-group have experienced to bring consequences of very practical character for systems development in work life.

A basic fact is that most computer-based decision-support systems for work-life have never come to practical use. Hence, our point of departure has been that since the software products from the MDA-group are supposed to be exactly such systems, there is a need to take this risk seriously. Evaluation of previous efforts have showed that one reason for development of “shelf products” is that too little attention has been paid to studies of the practice settings where the products are to be used (cp. Forsythe D. Blaming the user in medical informatics: The cultural nature of scientific practice. In: Rip A, Layne L, Hess D. Knowledge and Society. Greenwich, Conneticut: JAI Press, 1992.), and that the “voice” of the future users have been “muted”.

To be specific, the MDA-group has made an issue of the underlying *perspective* (cp. Floyd C. Outline of a paradigm change in software engineering. In: Bjerknes G, Ehn P, Kyng M. Computers and Democracy: A Scandinavian challenge. Brookfield: Gower Press, 1987) inherent in systems development projects in work life. This is many times based on a “top-down” view of organizations and processes. The contention is that there is a high risk for the above described “end-user failure” if such a view is again and again applied on decision support system development in working life.

We have thus experienced “difference in perspective” being a serious pragmatic factor for the ways methods in systems development are understood and used. But we also believe that a serious, continuous, and “up-front” discussion of perspectives and underlying standpoints can lead to new and fundamentally improved systems and methods, both for professionals and their organizations.

10.4.4 Courses and supervision

The MDA-group provides post-graduate courses and supervision of master’s students in our fields of interest. The intention is to bring to the students an inter-disciplinary approach to research, and to expose them to problems faced by practicing professionals outside the University. Three courses are given, first, *Hypermedia: History, concepts, and applications* where the aim is to give a basic knowledge of hypermedia systems and their use in work-life and education. This course was given for the first time 1989/90. Second, *Empirical research methods*, first given in 1990/91, which provides a basic knowledge of quantitative and qualitative methods for empirical research. Finally, *Computer Support for Cooperative Work*, given for the first time 1991/92. This course is provided in a problem-oriented fashion, giving opportunity to focus both on content learning and reflection over the group process mediating it.

10.5 Publications

Papers in international scientific journals:

Timpka T, **Arborelius E**. A method for study of dilemmas during health care consultations. *Medical Informatics* 1991;16:55-64.

Timpka T, **Buur T**. Medical reasoning and patient requests in decision-making for female genitourinary infections. *Methods of Information in Medicine* 1991;30:215-20.

Timpka T, **Buur T**, **Hedblom P**. Is expertext support feasible in work life applications? An empirical study using an expert panel for simulating “optimal” expertext support. *Information and Decision Technologies 1992*;18:81-87.

Arborelius E, **Timpka T**, **Nyce J**. Patients comments on video-recorded consulations: The “good” GP and the “bad”. *Scand J Soc Med* 1992, in press

Arborelius E, **Timpka T**. A comparison of patients’ and physicians’ comments on video-recorded consultations. *Scand J Primary Health Care* 1992, in press.

Timpka T, **Nyce J**. Dilemmas at a primary health care centre: a baseline study for Computer Supported Cooperative Work (CSCHW). Meth Inf Med 1992;31:204-9.

Nyce J, **Timpka T**. Work, Knowledge and Argument in Specialist Consultations: Incorporating Tacit Knowledge into System Design and Development. *Med Biol Eng Comp*, 1992, in press.

Book chapters, papers in conference proceedings and other scientific periodicals:

Timpka T, **Buur T**, **Hedblom P**. Evaluating the need for triangulation of expert systems, hypermedia, and online libraries. *Proc. of World Congress on Expert Systems*. New York: Pergamon Press, 1991.

Holmgren H, **Timpka T**, **Goldkuhl G**, **Nyce J**, **Sjöberg C**. Argumentative Design: a necessary complement to Action Design. *Proceedings of MedInfo '92* (World Conference of Medical Informatics). Amsterdam: Elsevier, 1992.

Sjöberg C, **Timpka T**, **Nyce J**, **Peolsson M**, **af Klercker T**. From clinical literature to medical hypermedia: procedures and experiences. *Proceedings of MedInfo '92* (World Conference of Medical Informatics). Amsterdam: Elsevier, 1992.

Timpka T, **Nyce J**. Towards a pragmatics of medical hypertext. *Proceedings of MedInfo '92* (World Conference of Medical Informatics). Amsterdam: Elsevier, 1992

Timpka T, **Hedblom P**, **Holmgren H**. Action design: Using an object-oriented environment for group process development of medical software. *Software Engineeering in Medical Informatics*, Elsevier, Amsterdam, 1992.

Timpka T, **Nyce J**, **Sjöberg C**, **Hedblom P**, **Lindblom P**. Developing a clinical hypermedia corpus: Experiences from a practice-centered method. *Proc. of SCAMC '92*. Baltimore: IEEE Press, 1992.

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

Sjöberg C, **Timpka T**. Activity Documentation: To bridge development and evaluation of technology in primary health care. Proc. of ISTAC 1992, Vancouver, Canada.

Timpka T. Dagliga dilemmor på en vårdcentral. Svenska Läkarsällskapets Riksstämma. *Hygia 1991*;100. (Abstract)

Svensson B, **Hedblom P**, **af Klercker T**, **Lindblom P**, **Rauch E**, **Tibblin G**, **Timpka T**. Utveckling och hantering av elektroniskt informationsmaterial. Svenska Läkarsällskapets Riksstämma. *Hygia 1992*;101. (Abstract)

Sjöberg C, **Timpka T**. Argumentativ Design: ett grafiskt språk för samarbete vid utveckligsarbete inom vården. Svenska Läkarsällskapets Riksstämma. *Hygia 1992*;101. (Abstract)

Master's Theses:

Peolsson M. Medicinskt beslutstöd inom primärvården: en kravspecifikation och systemdesign. LiTH-IDA-Ex-9131, 1991.

Vind H. User interface design with human activity in mind. *LiTH-IDA-Ex-9207*, 1992.

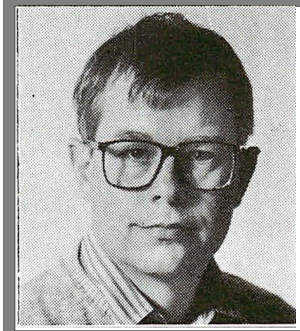
Research reports:

Timpka T, **Nyce J**, **Sjöberg C**, **Rehnwall H**, **Herbert I**. Bar code technology in health care: Using a business model for study of technology application and dissimination. Report. Submitted to MIE '93.

Sjöberg C. An analysis of the perspectives behind CBS and Action Design. *Technical Report IDA-MDA-9202*, Department of Computer Science, Linkoping University, 1992.

Sjöberg C. "Change" vs. Information system development in health care. *Technical Report IDA-MDA-9205*, Department of Computer Science, Linkoping University, 1992.

NLPLAB



Lars Ahrenberg
director

11 The Laboratory for Natural Language Processing

Dialog systems
Discourse representation
Unification-based grammars
Incremental parsing
Text understanding
Multi-lingual generation
Machine-aided translation

Some of the members in the laboratory for Natural Language Processing.
Back row: Richard Hirsch, Arne Jönsson, Mats Wirén.
Middle row: Åke Thurée, Peter Ingels, Lisbeth Linge, Magnus Merkel, Stefan Svenberg, Lena Strömbäck.
Front row: Nils Dahlbäck, Lars Ahrenberg.

Figure 11-1. *Lisbeth Linge nails up Mats Wirén's doctoral thesis on the "Doctoral Tree", the thesis committee is gathered to evaluate the thesis, and Mats Wirén receives the congratulations by the department.*

11.1 The Group and its Members

The Natural Language Processing Laboratory was formed in 1986 with five members. Since then the group has grown slowly but steadily and research interests have been extended in several directions; from analysis to generation and translation, from dialog to text understanding, and from grammar formalisms to general knowledge representation languages.

The group receives external support from the Swedish National Board for Industrial and Technical Development (NUTEK), the Board of Research in the Humanistic and Social Sciences (HSFR) and the Board of Research in the Technical Sciences (TFR). There is also a project within the local interdisciplinary research program concerned with industrial information technology, CENIIT.

Laboratory leadership, administrative and technical staff:

Lars Ahrenberg, Ph. D., associate professor
Lisbeth Linge, secretary
Bernt Nilsson, research engineer

Employed research staff:

Nils Dahlbäck, Ph. D., assistant professor
Richard Hirsch, Ph. D., lecturer
Arne Jönsson, Tech. Lic, lecturer
Åke Thurée, M. Sc., research assistant

Graduate students:

Peter Ingels, M. Sc.
Magnus Merkel, Ph. Lic.
Lena Strömbäck, Ph. Lic.
Stefan Svenberg, M. Sc.

Associated persons:

Mats Wirén, Ph. D, (DFKI, Saarbrücken, as from November 1992)

11.2 Overview of Current Research

11.2.1 Natural Language Dialog Systems

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

11.2.1.1 The LINLIN architecture

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

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

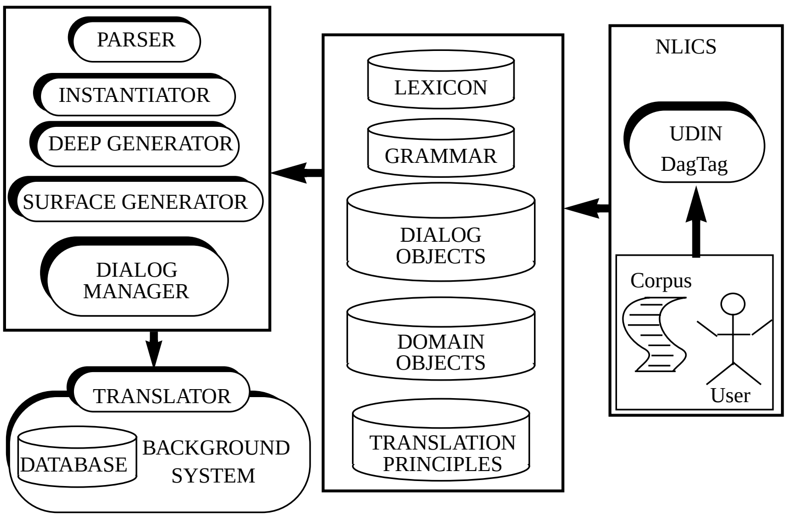


Figure 11-2. *The LINLIN architecture. The processing modules appear in the upper left part of the figure, while the customizable knowledge bases appear in the middle. The right part represents various tools needed for customization of the system.*

An aim of our work has been to represent all knowledge in the same structure, and in the same representation language. This would make it possible to develop the linguistic knowledge and the domain knowledge simultaneously in the same environment. It also makes it possible in principle to integrate syntactic and semantic processing. The processing modules that we have implemented so far, however, differ in the representation languages they assume.

The central processing module of the system is the dialog manager, DM, which receives user inputs, controls the data-flow of the system and maintains the discourse representation (Figure 11-3).

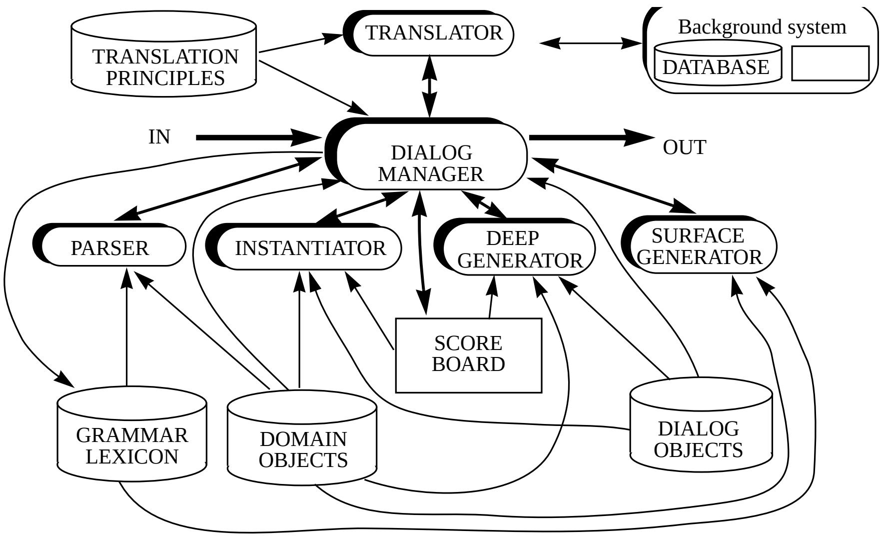


Figure 11-3. *Data flow in LINLIN.*

The discourse representation consists of three dynamic structures. The first one, the score-board, keeps information about salient objects and properties which is needed by the instantiator and generator modules. The score-board is basically an interface to the second dynamic structure, a dialog tree which represents the entire dialog as it proceeds in the interaction. The nodes of the dialog tree are instances of dialog objects, i.e. various types of moves and segments. They carry information about properties such as speaker, hearer, topic and focus, and are associated with a local plan. The plan is structured in terms of actions and combine with similar plans of other nodes to form the third structure, the action plan stack where the actions to be performed by the DM are stored.

The DM is thus characterized by its distributed control. The actions of the action plan stack are distributed on the nodes of the dialog tree that are still open. This means that if, say, the parser fails with a certain node being current, that node creates an instance of a clarification request segment, which will control the dialog during the clarification. This segment consists of two parts, one part for prompting the user with a clarification request and another to interpret the user input. Finally the user response is integrated into the dialog tree. The distributed design has the advantage that we can use quite simple, local plans. Detailed descriptions of the dialog manager can be found in Ahrenberg, Jönsson and Dahlbäck, (1990) and Jönsson (1991).

Thesis work

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

Nils Dahlbäck have been conducting, as described in the following section. The work is also concerned with the problems of customizing a natural language interface to an application.

11.2.1.2 Wizard-of-Oz studies and NL-dialog characteristics

An important part of the work on dialog systems has been to find characteristics of the sublanguage of man-machine communication in NL, which would be useful for the design of NL-interfaces. Empirical studies of this kind of dialogs have been undertaken for some time now in our group using so-called Wizard of Oz experiments, i.e. by letting users communicate with a background system through an interface which they have been told is a natural-language interface, but which in reality is a person simulating such a device (Dahlbäck, Jönsson & Ahrenberg, 1992).

In an earlier project nine different background systems and scenarios were tested and a corpus of 21 dialogs with five of these systems were analysed (Dahlbäck & Jönsson,1989; Jönsson & Dahlbäck, 1988). It appeared that only 48% of the utterances could be interpreted in isolation, ellipsis (64%) and definite descriptions (29%) being the most common anaphoric devices, whereas the use of pronouns was more limited (16%). Indirect speech acts, on the other hand, were few and seem to performed using only a small set of standardized expressions.

The background systems were of three different kinds in this study: data base systems supporting information-seeking interactions, order systems supporting both information-seeking and ordering tasks, and an expert system supporting advice-giving. In Dahlbäck (1991) the results of analysing the corpus with a grammar-based model using IR-segments as the basic unit are presented. In general the model fits the data well, although there were differences between the different types of systems. While there is an almost perfect fit for the data base and ordering dialogs, the advice dialogs show some more complex interaction patterns. This work is now being followed up in a project aimed at comparing grammar-based and plan-based models of dialog for the purposes of the analysis of natural language dialog with computers.

This work, as well as similar studies by others, indicates that dialogs with computers in written natural language differ from dialogs between people. It is still, however, an open question to what extent these differences are due to assumed and real differences between people and computers as dialog partners, or due to the qualities of the communication channel. In a recent project we have collected a corpus of 60 dialogs to study these questions. Three different scenarios were used, two of which concerned querying a data base for information, but on different domains. The third scenario involved both ordering and data base querying. For each scenario, 10 subjects were told that they were interacting with a computer system directly, and 10 were told that they were interacting via terminal with a person having such a system on his desk. The analysis of this corpus is still in progress.

Thesis

Nils Dahlbäck's thesis (Dahlbäck, 1991) is concerned with empirical studies of connected discourse. The first part starts out from Johnson-Laird's theory of mental models, and evaluates it as a theory of discourse representation both theoretically and empirically. The main result is that the term “mental model” is used in two different senses; as an image-like representation used for processing text with a strong visual content, and which seems to be under voluntary conscious control, and as a term for a discourse representation structure richer than a mental representation of the text’s surface structure, but with less constraints on its format than in the former case. This result is also related to the general discussion within AI and philosophy of mind on the so-called representational theory of mind. In the second part the characteristics of human-computer dialogs in natural language is studied using the Wizard of Oz technique as described above. The thesis reports results not only on dialog structure but also on the use of referring expressions such as pronouns and definite descriptions.

Workshop

In connection with the Third Conference on Applied NLP in Trento, Italy, March 31 - April 3 1992, Lars Ahrenberg, Nils Dahlbäck and Arne Jönsson organized a workshop on Empirical Models and Methodology for Natural Language Dialog Systems. Ten papers were presented at the workshop covering a range of topics on the use of empirical data, and simulation data in particular, in system development and dealing both with speech and text interfaces.

11.2.2 Unification-Based Grammar Formalisms

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

11.2.2.1 The need for extensions

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

In our work on grammar development we have explored several of the proposed extensions and here three topics of current interest will be reported: (i) how to make unification-based systems more flexible, (ii) the development of unification algorithms for feature structures with disjunction and, (iii) the exploitation of field structure in feature structures for the expression and control of word order constraints in configurational languages.

11.2.2.2 Flexible unification

Lena Strömbäck has explored various possible and proposed extensions to unification-based formalisms. In her licentiate thesis, Strömbäck (1992a), she surveys various possible and proposed extensions to these formalisms and some of the recent systems that attempt to be comprehensive in the sense of allowing several of them in the same formalism. A number of evaluation criteria are also provided. Beside expressiveness and tractability flexibility and predictability are noted as important riteria. Flexibility means that it should be possible to vary and adjust the constructions provided by the system to the needs of a given application. Predictability means that the grammar writer should be able to predict the consequences on system behaviour when using the constructions allowed by the system for a certain purpose. Strömbäck notes that no existing system meets the demands on flexibility and proposes a framework that would make formalisms and systems better in this respect.

Thesis work

The work on extensions to unification-based formalisms is the subject for Lena Strömbäck's continued thesis work. The aim of the thesis is to develop a formalism that is as powerful as a unification-based grammar with many of the extensions proposed in current literature, but flexible in the sense defined above. This work is performed in the project General Unification sponsored by the TFR.

11.2.2.3 Disjunction in feature structures.

We have so far given most attention to the use of disjunction and negation in feature structures. Various known unification algorithms for disjunction have been implemented and integrated with negation. A new algorithm has also been specified and implemented (Strömbäck, 1992b) which is predictable and flexible in the sense that it is as fast as an algorithm without disjunction when no disjunctions are part of the structures to be unified, as fast as an algorithm handling only local disjunctions when there are no non-local disjunctions, and expensive only in the case of full disjunction. The specification is given in the framework of graph unification algorithms which makes it easy to implement it as an extension of such an algorithm.

11.2.2.4 Field grammar

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

We have used Diderichsen's ideas in a formalism, termed Field-and-Category Grammar (Ahrenberg, 1989, 1990), which can be viewed as a variant of LFG. Apart from the syntactic notions recognized in the phrase-structure part of an LFG, such as category, dominance and linear precedence, it also recognizes fields and schemas. Schemas account for the basic word order constraints of major categories such as clauses and noun phrases. Thus, regularities of word order can be stated independently from rules of constituency just as in context-free grammars written in the ID/LP-format. A schema gives more information than an LP-rule, however, as it encodes a word order pattern for a major category. On the other hand it is less general since its application is restricted to that particular category. The formalism uses category definitions rather than rewrite rules, and relies on property inheritance to express generalizations.

Current work includes the development of a chart-parser that exploits topological constraints in predictions. We are also looking at how fields and schemas can be integrated in a framework such as HPSG.

11.2.3 Text Understanding

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

11.2.3.1 Incremental parsing and interpretation

For human beings reading and interpreting a text is an incremental process; it does not happen instantly, but is carried out bit by bit as the text is consumed. Incremental processing is required also by many types of real-time and interactive natural language applications, for example, language-sensitive text editing and speech-to-speech translation of telephone dialogues. Systems of this kind must be capable of analysing text on-line, as it is received, and sometimes also to cope with changes of previous input without reverting to exhaustive recomputation.

Common techniques for natural language analysis such as chart parsing and unification easily support ordinary left-to-right incremental processing. Mats Wirén has also studied the problem of full incrementality, which means that parsing and interpretation is not restricted to piecemeal, left-to-right analysis but handles *arbitrary* changes of the input. Such changes typically occur when a text is produced by a text-editor; words or word strings in the middle of a text are deleted, replaced or separated by inserted words. The algorithm is based on chart-parsing and unification.

A chart parser basically works by recording partial analyses of input and hypotheses as regards the phrases that occur in it, in a common structure, the chart. It avoids back-tracking by ensuring that any partial result is registered in the chart, and avoids repeated computation by making sure that any such partial result is only registered once. The hypotheses, or active edges, need to find partial analyses, or inactive edges, that satisfy their needs, e.g. in terms of having the right phrasal category. Apart from the edges corresponding to words of the input, which are obtained from the lexicon, and the initial hypotheses, which may be determined in different ways, most partial results come from successful combinations of active and inactive edges.

To achieve full incrementality the algorithm makes use of dependencies between chart edges, in particular dependencies between edges that combine successfully and their results. Using these basic dependencies a disturbance set can be calculated for any edge corresponding to a word and for any space between two words. Since a change in the input will affect either one or more existing words or a space somewhere in the string (if new material is inserted), the part of the chart that needs to be reanalyzed can be determined exactly. The algorithm basically works in a change-update loop, where changes are minimal events in the input text such as a deletion or insertion, and the updating involves deparsing, modifications of the basic chart structure, and reparsing.

The algorithm has been implemented in a system called LIPS (Dahlén, 1991). The system, which can be thought of as a rudimentary language-sensitive text editor, consists of four processing modules: editor, scanner, incremental chart parser and an optional chart displayer. At each moment, the parser tries to keep an accurate analysis of what the user has typed in so far. The editor keeps track of each user change, down to the keystroke (character) level. This information is passed to the scanner, which groups new input into tokens and translates the change corresponding to the change-update loop to an update operation (delete, insert or replace). The resulting update request is then queued to the parser, which in turn produces an analysis in the form of an updated chart.

As shown in Wirén (1990) the operation of a chart parser with edge dependencies has striking similarities with the behaviour of a reason maintenance system. It has therefore been named a reasoned chart parser.

Thesis work

Incremental parsing and interpretation is the subject of Mats Wirén's Ph.D. thesis (Wirén, 1992). It describes the incremental chart parsing algorithm referred to above and further develops the notion of a reasoned chart parser. It also shows how the incremental framework can be coupled with other forms of inference, such as abduction, and how it can be extended to handle aspects of discourse interpretation such as noun phrase reference resolution.

11.2.3.2 Parsing Ill-formed Input

A problem for natural language interfaces is that the user is likely to input strings that it cannot parse, because of miss-spellings and other mistakes resulting in the input being ill-formed. Similarly, an important purpose of a language-sensitive text-editor is to detect mistakes in the user's input.

In a Master's thesis project Peter Ingels works on an algorithm that detects ill-formed input and attempts to find reasonable corrections. The algorithm extends the algorithm presented in Mellish (1988) to unification-based grammars and also incorporates relaxation techniques (Douglas & Dale, 1992). Instead of simply relaxing an identity constraint, it collects the conflicting feature values it encounters and infers an interpretation on the basis of its lexicon and annotations in the grammar.

This work has partly been performed at IRST, the Institute for Scientific and Technological Research in Trento, Italy, in collaboration with the IRST research group on NLP.

11.2.3.3 Discourse representation

Several theories of discourse that are relevant for NLP make central use of some notion of a discourse segment. A problem with all of them, however, is that they do not provide a definition of segment which is general and precise enough for computer applications. In these circumstances we saw it necessary in our dialog system project to adopt a sublanguage approach to discourse representation and processing, using simulation data as the primary source of data for development of a model.

A basic finding of the studies was that almost all input from users (and output from the systems) could be classified as either initiatives or responses and that initiatives typically introduces a single goal in the form of a single question or request. Nestings could occur, however, so that an initiative from the system could be countered by an initiative from the user e.g. requesting some clarification from the system. Still, the overall structure of the dialogue can be given a simple tree structure in terms of segments defined by initial initiatives and closing responses. Moreover, this segment structure correlated strongly with the range of anaphoric references (Dahlbäck, 1991) and it seemed possible to keep track of the focused information in each segment by means of a small list of attributes that hold items that are likely to be referenced by a pronoun or be implicit in a following utterance (Ahrenberg, Jönsson, Dahlbäck, 1990). These results can be summarized by saying that a grammar-based approach to discourse representation seems sufficient for many important application areas so that the complexity associated with the more general plan-based approaches can be avoided (Jönsson, 1991).

In current work we are further investigating the role of segments in discourse representations (Ahrenberg, to appear) on the one hand, and the relative merits on plan-based vs. grammar-based accounts of discourse structure.

11.2.4 Generation and Translation

There are currently two projects directed towards generation and translation, both still on a small scale. In one project the problem of inter-linguas is investigated in the context of a system for automatic multi-lingual generation and translation, and in the other project we investigate computer support tools for translators and technical writers. Both projects are developed with the aim of finding appropriate applications of NLP-techniques in the document production process.

11.2.4.1 Contextual text representations

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

In the project Conceptual Text Representation for Automatic Generation and Translation, funded by CENIIT, Lars Ahrenberg and Stefan Svenberg are investigating a knowledge-based approach to text generation and translation (Ahrenberg & Svenberg, 1992). The project works with a small corpus of paragraphs from service manuals and Swedish and English as target languages. The primary purpose of the project is to develop a representation language for the descriptive texts of such manuals and test its usefulness by generating actual Swedish and English text from sample representations. A demand on the representation language is that it should be rich enough to represent the components that the texts describe and their properties and relations. It should also be able to express the links between various parts of the text that contributes to its coherence, such as relations between anaphors and antecedents and rhetorical relations between text chunks.

So far a system environment has been set-up and a two-step sentence generation algorithm has been implemented based on graph unification and SLD-resolution. The first step generates a language-specific grammatical representation given a specification of the content of the sentence and the second step realizes the grammatical representation as a text sentence in the manner of van Noord (1989). In the first step the knowledge base is consulted both for linguistic information and domain information. Thus, a content specification may have the form of a question and the generator retrieves the answer to this question as part of the processing. Currently we are extending the system to be able to handle small paragraphs, using the idea that a paragraph in the descriptive part of the manuals we are working with can be analysed as meeting a specific information need, expressed by one or several questions.

11.2.4.2 A text analysis tool

Already in the introduction the importance of studying the text genre of interest in a given application was stressed. This has long been recognized in the field of machine translation, where the notion of sublanguage was first applied. For purposes of machine-aided translation one may however also explore more superficial properties that characterize text genres. One such property that Magnus Merkel have investigated is recurrency (Merkel, 1992).

Merkel has developed a tool that pre-processes the source text and gives various kind of information that forms decision support whether translation tools should be applied at all. The system identifies both recurrent sentences and recurrent word strings of a text and state their frequencies. It has been run on several real handbook texts from major computer software companies with interesting results: up to 43 per cent of a text could be repetitious. If we consider both repetitions within one document as well as repeated patterns across documents, there is evidence in the corpus that 55 per cent of the text in one document can be regarded as recurrent.

The system is planned to be developed in several directions. One extension concerns the identification of more abstract linguistic structures such as lemmas and construction types. Another extension concerns the recognition of parts (say, chapters) of a large document that are similar in the sense of having a high percentage of the vocabulary and phrasing in common.

References:

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

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

Ahrenberg Lars (to appear). On the integration and scope of segment-based theories of discourse. Paper presented at the Third Nordic Conference on Text Comprehension in Man and Machine, Linköping, 21-23 April, 1992.

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

Ahrenberg, Lars & Stefan Svenberg (1992). Conceptual text representations for multi-lingual generation and translation. *Proceedings of the Scandinavian Conference on Computational Linguistics*, Bergen 28-30 November, 1991, pp. 7-18.

Dahlén, Ulf (1991). LIPS. Ett system för interaktiv och inkrementell parsning av naturligt språk. Master’s Thesis LiTH-IDA-Ex-9136. Department of Computer and Information Science, Linköping University.

Dahlbäck, Nils (1991). Representations of Discourse: Cognitive and Computational Aspects. Doctoral diss., Linköping Studies in Arts and Science 71 and Linköping Studies in Science and Technology 264.

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

Dahlbäck, Nils, Arne Jönsson and Lars Ahrenberg (1992). Wizard of Oz studies - Why and How. Research report LiTH-IDA-R-92-19. Revised version to appear in *Proceedings of the International Workshop on Intelligent Systems*, Orlando, Florida, January 4-8, 1993.

Douglas, & Robert Dale (1992). Towards Robust PATR. In *Proceedings of the 15th International Conference on Computational Linguistics*, Vol. 2, pp. 468-474.

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

Jönsson, Arne (1991). A Dialogue Manager Using Initiative-Response Units and Distributed Control. *Proceedings of the Fifth EACL Conference*, Berlin, 9-11 April, 1991, pp. 233-238.

Mellish, C. S. (1988). Some Chart-based Techniques for Parsing Ill-formed Input. In *Proceedings of the 27th Annual Meeting of the ACL*, pp. 102-109.

Merkel, Magnus (1992). Recurrent Patterns in Technical Documentation. Research Report LiTH-IDA-R-92-31.

Strömbäck, Lena (1992a). Studies in Extended Unification-Based Formalisms for Linguistic Description: An Algorithm for Feature Structures with Disjunction and a Proposal for Flexible Systems. Linköping Studies in Science and Technology, Thesis No. 318.

Strömbäck, Lena (1992b). Unifying Disjunctive Feature Structures. *Proceedings of COLING-92*, Nantes, August 23-28, 1992, pp. 1167-1171.

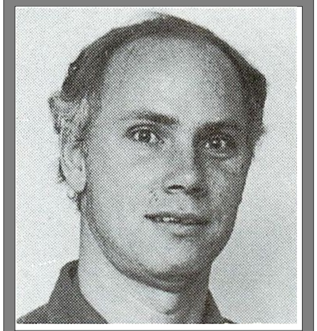
van Noord, Gertjan (1989). BUG: A Directed Bottom-Up Generator for Unification-Based Formalisms. Working Papers in Natural Language Processing 4. Katholieke Universiteit Leuwen.

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

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

Wirén, Mats (1992). Studies in Incremental Natural-Language Analysis. Linköping Studies in Science and Technology, Dissertation, No. 292.

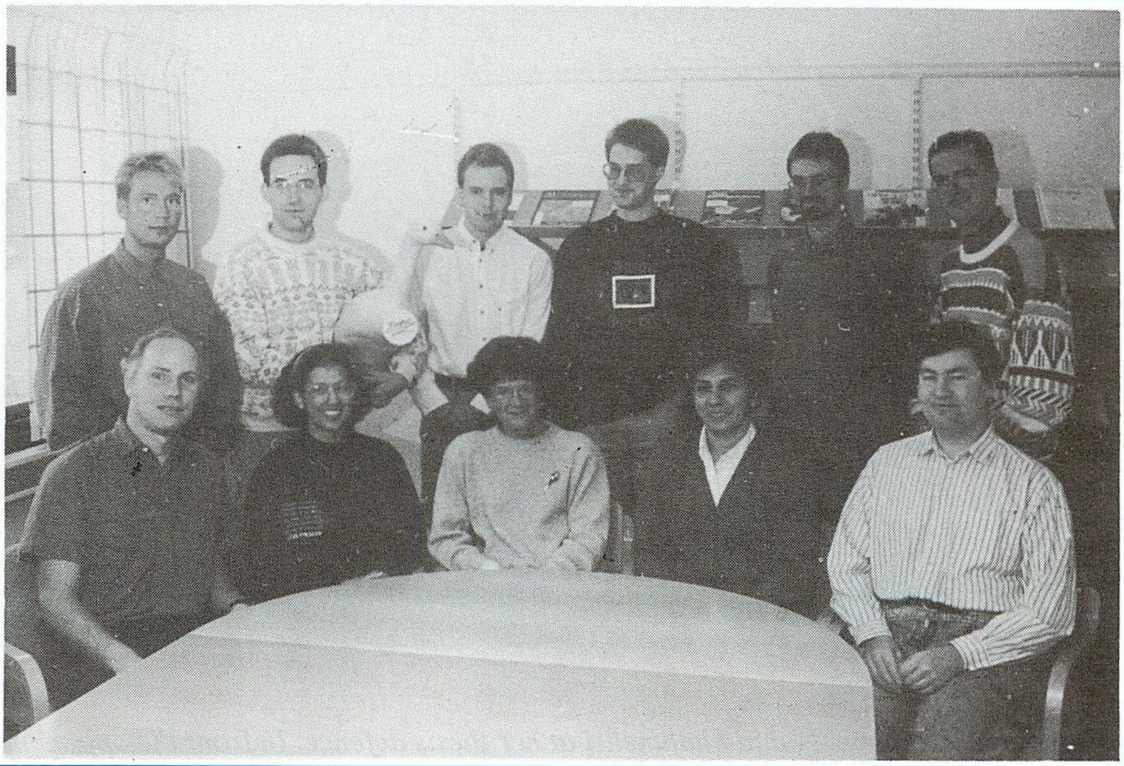
PELAB



Peter Fritzson
director

12 The Laboratory for Programming Environments

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



Some of the members in the laboratory for Programming Environments.
Back row: Lars Viklund, Mikael Pettersson, Joakim Malmén, Henrik Nilsson, Johan Herber, Niclas Andersson.
Front row: Peter Fritzson, Mariam Kamkar, Gunilla Lingenhult, Nahid Shahmehri, Johan Ringström.

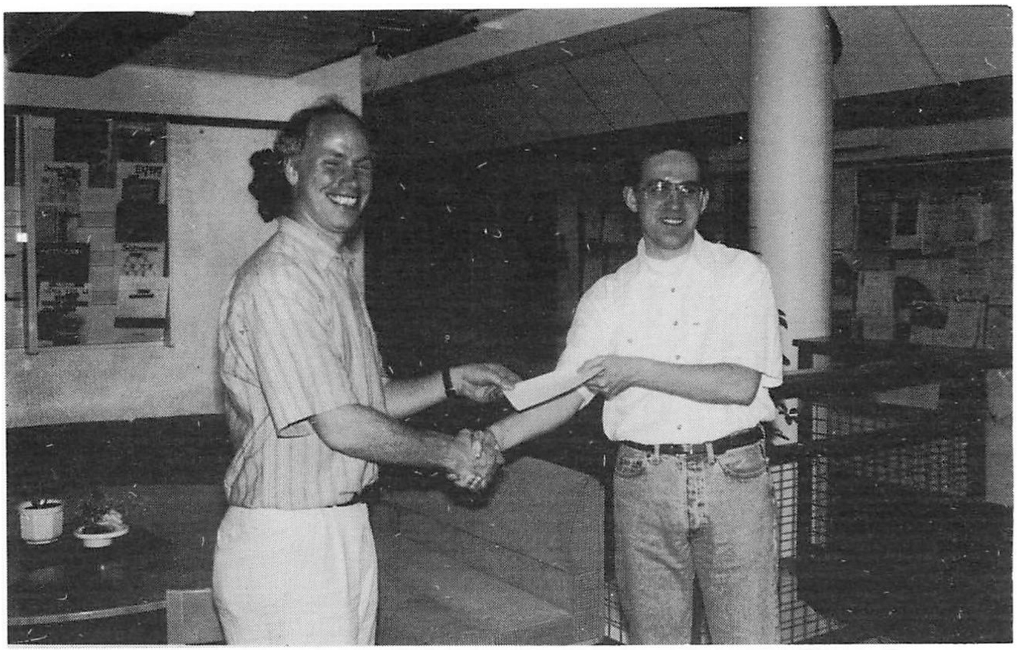


Figure 12-1. *Above, Nahid Shahmehri at her thesis defence. In front the opponent MaryLou Soffa and to the left Nahid's supervisor Peter Fritzson. Below, Mikael Pettersson is congratulated by his supervisor, Peter Fritzson, after the presentation of his licentiate thesis.*

12.1 Introduction

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

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

- Programming environment support for high-level scientific programming in equations and compilation to efficient code
- Semi-automatic debugging support through generalized algorithmic debugging
- Support for debugging parallel and distributed real-time systems
- Dynamic interprocedural data flow analysis
- Practical compiler generation from denotational semantics specifications
- Parallel programming and compiler generation for parallel hardware

12.2 Laboratory Members and Guests

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

Laboratory leadership and administrative staff:

Peter Fritzson, Ph. D., associate professor.
Gunilla Lingenhult, secretary.

Laboratory members having a Ph. D. degree:

Johan Fagerström, Ph. D., assistant professor.
Nahid Shahmehri, Ph. D., lecturer.

Employed graduate students:

Mariam Kamkar, Tech. Lic. (Ph. D. spring 1993)
Mikael Pettersson, Tech. Lic.
Johan Ringström, M. Sc.
Lars Viklund, M. Sc.

The work in PELAB is supported by NUTEK, The Swedish Board for Industrial and Technical Development

Johan Herber, M. Sc.
Niclas Andersson, M. Sc.
Henrik Nilsson, M. Sc.
Joakim Malmén, M. Sc.

Guest Researchers:

Mikhail Auguston, Latvia University, Riga, Latvia, March-June 1992.
Vadim Engelson, Latvia University, Riga, Latvia, January–July, 1993.

Full-time teachers

Rober Bilos, Tech. Lic.
Lars Strömberg, Tech. Lic.

Part-time employed persons

Rickard Westman
Mats Palmgren
Tommy Hoffner

12.3 Degrees and papers

The following degrees were awarded to PELAB members during 1991/92:

Nahid Shahmehri completed her Ph. D.
The title of the thesis is: *Generalized Algorithmic Debugging.*

Mikael Pettersson completed his licentiate degree.
The title of the thesis is: *DML– A Language and System for the Generation of Efficient Compilers from Denotational Specifications.*

Eighteen papers were accepted for publication or presented at international conferences by PELAB members during 1991/92. In addition to these, three journal papers were accepted or published and five technical reports were written.

12.4 Current Research Projects

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

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

Another approach is to raise the language level, i.e. to provide more powerful programming constructs. This is most natural in the context of specialized application areas. Three of our projects use this approach. The goal of one project is to compile language specifications written in DML, Denotational Meta Language, into an efficient language processors that generates quadruples. During 1992, this resulted in a compiler generator that can produce efficient stand-alone compilers implemented in

C. A second project aims at raising the level of parallel programming. The goal is to provide a machine independent way of parallel programming that can be efficiently compiled onto different parallel hardware. A third project aims at constructing a high-level programming environment for scientific programming, which provides support for high-level programming in equations, instead of Fortran.

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

International Cooperation

It should also be mentioned that PELAB is associated partner in the industrial Esprit project GIPE II (Generation of Incremental Programming Environments II). Within this project we are doing work on debugging tools and on generating parts of environments for scientific computing. PELAB has recently joined another industrial Esprit project: PREPARE (Programming Environments for Parallel Architectures), where work is being done on compilers and environments for massively parallel computers.

12.4.1 Generalized Algorithmic Debugging

Peter Fritzson, Nahid Shahmehri, Henrik Nilsson, Joakim Malmén, Rickard Westman

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

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

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

Algorithmic Debugging for Imperative Languages

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

During the actual bug localization phase of algorithmic debugging, the system traverses the execution tree of all activation records. Thus, a trace has to be 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. This can be improved by also allowing general assertions as answers.

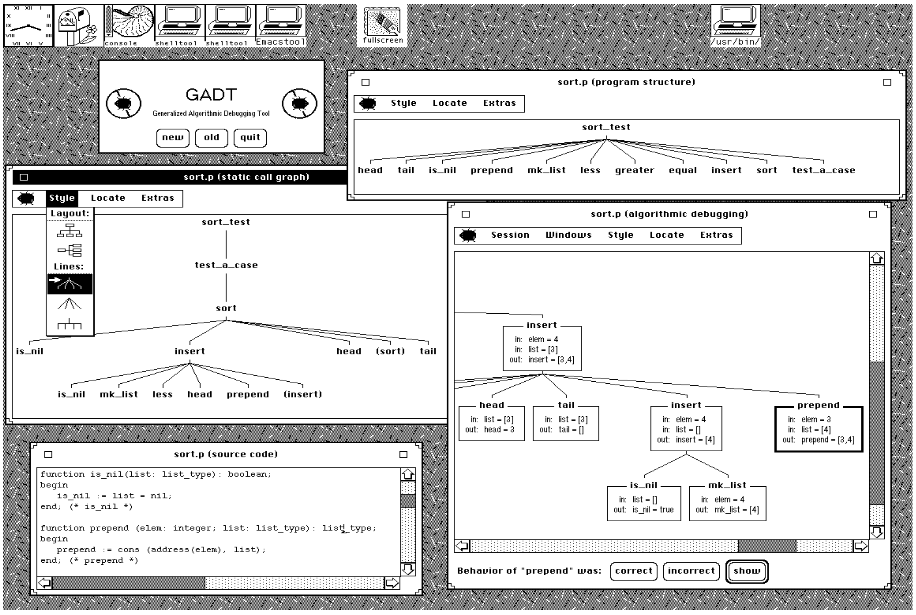


Figure 12-2 *The graphical user interface of the GADT system, including a graphical display of the execution tree. The bug localization algorithm searches the execution tree while interacting with the user when necessary.*

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

Focusing Bug Localization through Static Program Slicing

Program slicing, a data flow analysis technique, can be used to focus the search process during bug localization with algorithmic debugging. Given a variable at a certain point in the program, program slicing will compute the set of all statements whose execution might have contributed to the variable’s current value. Thus the slicing technique will extract a subset (or slice) of the program. The slicing is applied each time the user of the algorithmic debugger is questioned by the debugger provided the user can point out an erroneous data value. It eliminates the parts of the execution tree which are independent of the erroneous value, and thus irrelevant for finding the bug.

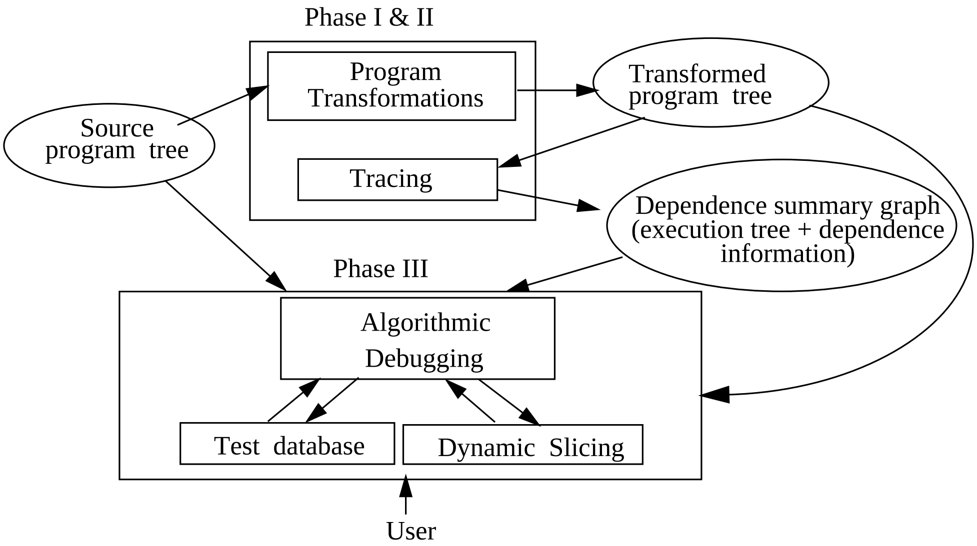


Figure 12-3 *The functional structure of the generalized algorithmic debugging system when integrated with dynamic slicing. Arrows denote information transfer.*

Algorithmic Debugging for Lazy Functional Languages

Lazy functional languages have non-strict semantics, i.e. function arguments are only evaluated when needed, and are purely declarative. Traditional debugging techniques are, however, not suited for lazy functional languages since computations generally do not take place in the order one might expect, which is confusing for the user.

Since algorithmic debugging allows the user to concentrate on the declarative aspects of program semantics, and will semi-automatically find functions containing bugs, we have extended this technique for debugging lazy functional programs. Because of the non-strict semantics of lazy functional languages, arguments to functions are in general partially evaluated expressions which often are hard to understand. The user is usually more interested in the values that these expressions represent. We address this problem by providing the user with a *strictified* view (i.e. a view where the partially evaluated expressions have been evaluated) of the execution trace whenever possible.

An algorithmic debugger according to these ideas was implemented for a lazy functional language and was found to be useful in debugging some program examples. The strictified view of the execution trace makes the debugging process independent of the complexity of lazy evaluation order and also helps the user to focus on the high-level declarative semantics of the application program. A problem is that perfect strictification is not achieved in the current implementation; another problem is the large space and time overhead caused by building the entire execution tree.

12.4.2 Interprocedural Dynamic Slicing

Peter Fritzson, Mariam Kamkar, Tommy Hoffner

One drawback of static program slicing, i.e. slicing based on data flow analysis, is that it may produce larger slices than necessary. For example, consider an if-then-else statement. If we do not know whether the then-part or the else-part will be executed, both of them must be included in the static slice. On the other hand, if statements are traced during execution, a smaller and more precise *dynamic slice* can be computed.

```
Instruction Program
number      text
------------------------------------
      program sumofoddnrs;
      var n, sum, i: integer;
      begin
1         sum:= 0;
2         i:= 1;
3         read(n);
4         while n>0 do begin
5             sum := sum + i;
6             i :=i + 2;
7             n := n -1;
          end;
8         write(sum);
      end.
```

(a): P

Execution point	Instruction number	Instruction text
1	1	sum:= 0;
2	2	i:= 1;
3	3	read(n);
4	4	while n>0 do begin
5	5	sum := sum + i;
6	6	i :=i + 2;
7	7	n := n -1;
8	4	while n>0 do begin
9	8	write(sum);

(b): T

Execution point	Instruction number	Instruction text
1	1	sum:= 0
2	2	i:=1;
3	3	read(n);
4	4	while n > 0 do begin
5	5	sum := sum + i;
9	8	write(sum);

(c): T’

```
Instruction Program
number      text
------------------------------------
      program sumofoddnrs;
      var n, sum, i: integer;
      begin
1         sum:= 0;
2         i:=1;
3         read(n);
4         while n > 0 do begin
5             sum := sum + i;
          end;
8         write(sum);
      end.
```

(d): P’

Figure 12-4: *(a) The source program, P for an example program which sums the n first odd numbers starting at 1. (b) The trace execution T of the program P. (c) An execution slice T′ of T with respect to variable sum at execution point 9. (d) The dynamic program slice P′ of the source program P. Note: If termination-preserving control dependency had been used when computing the execution slice in (c), the program slice in (d) would have been a terminating program.*

Previous methods for dynamic slicing could only handle small toy languages without procedures. However, in this project a new algorithm for *interprocedural dynamic slicing* has been developed. During execution of each procedure, this algorithm builds a temporary dependency graph to represent exact dependences between statements in the procedure. This graph is subsequently collapsed, and the dependencies are transferred to dependencies between input and output parameters and global variables in an interprocedural *summary graph*, which is substantially more compact. The method is described in [Kamkar et al. 92] and in a forthcoming Ph.D. thesis due in March 1993 by Mariam Kamkar. The method is currently being integrated with the algorithmic debugging system to improve bug localization.

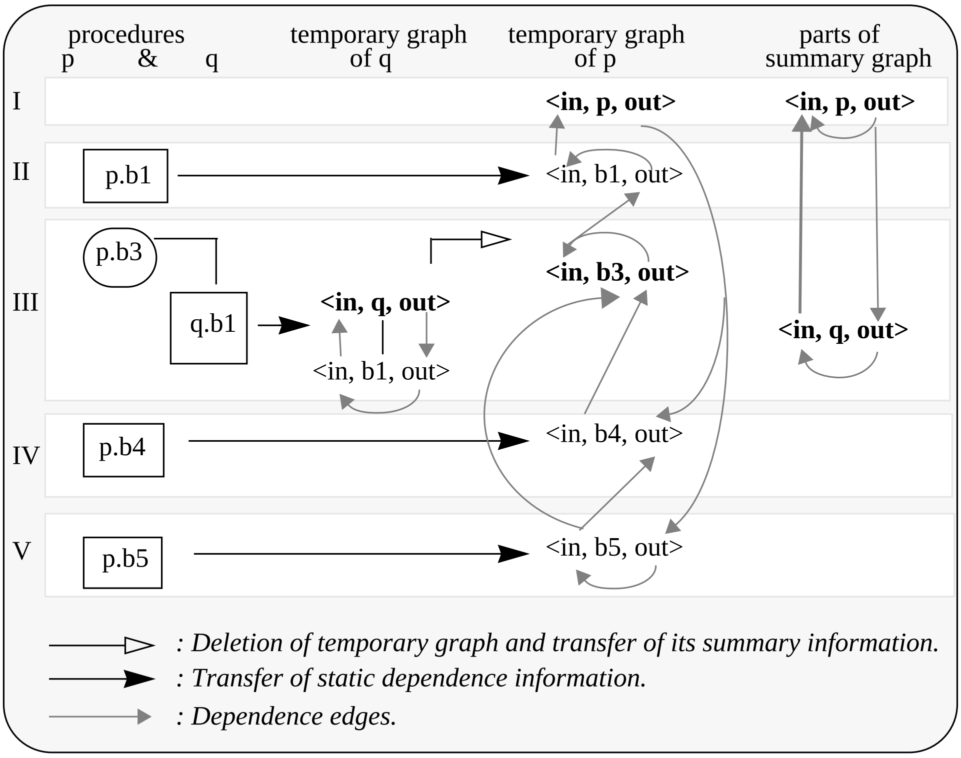


Figure 12-5: *An example of building the temporary graph during dynamic interprocedural slicing of procedure p. (I) The entry block of the temporary graph p and the procedure activation of procedure p in the summary graph. (II) The block instance of basic block b1and its internal dependencies created from compile-time information. The external dependencies of b1 are shown, too. (III) The creation of a block instance and its internal dependencies for the call block b3 with help of the temporary graph of procedure q. The procedure activation of q and its internal dependencies are created in the summary graph, too. (IV) The block instance for basic block b4 and its external dependencies. (V) The block instance of b5 and its internal dependencies together with its external dependencies.*

12.4.3 Debugging Tools for Distributed and Real-Time Systems

Peter Fritzson, Mikhail Auguston, Mats Palmgren

A Visualizing Debugger for a Multi-Paradigm Parallel Programming Language

The initial motivation for this work was to make parallel programming primitives from different paradigms available within a single environment for teaching purposes. For example, primitives such as semaphores, monitors, channels, remote procedure calls and rendezvous are all available within the developed PREDULA language system. Some interesting synergy effects were observed when students in courses on parallel programming used these different primitives together.

Figure 12-6. *The PREDULA visualizing multi-process debugger in action.*

The PREDULA language is compiled into an intermediate language called ANTS, which is then interpreted. The system has successfully been used in courses on parallel programming at Linköping University. A highly visual debugger for multi-process applications has also been developed. The visualization approach used by the debugger with one small window for each process is useful for applications of up to approximately 20-30 processes until the screen becomes too cluttered with small windows. It is highly instructive to observe execution using this debugger, as the program pointer in each process window is animated in this debugger. Even though the current debugger implementation only supports PREDULA, the user interface could be transferred to other parallel language implementations.

Debugging and Computation over Program Execution Histories

PARFORMAN (PARallel FORMal ANnotation language) is a high-level specification language for expressing intended behaviour or known types of error conditions when debugging or testing parallel programs. The high-level debugging approach which is supported by PARFORMAN is model-based. Models of intended or faulty behaviour can be succinctly specified. These models are then compared with the actual behaviour in terms of execution traces of events, in order to localize possible bugs. PARFORMAN can also be used as a general language for expressing computations over execution histories.

PARFORMAN is based on a precise axiomatic model of target program behavior. This model, called H-space (History-space), is formally defined through a set of general axioms about three basic relations, which may or may not hold between two arbitrary events. Events may be sequentially ordered (`SEQ`), they may be parallel (`PAR`), or one of them might be included in another composite event (`IN`).

The general notion of *composite event* is exploited systematically, which makes possible more powerful and succinct specifications. The high-level notion of *event-gram-*

mar is introduced to describe allowed event patterns over a certain application domain or language. Auxiliary composite events such as *snapshots* are introduced to be able to define the notion “occurred at the same time” at suitable levels of abstraction. Program paths can be described by regular path expressions over events. Current work involves implementation of a PARFORMAN interpreter for Occam and finding realistic applications.

12.4.4 Generation of Practical Compilers from Denotational Semantics

Peter Fritzson, Mikael Pettersson

Generating compilers from formal specifications of programming languages has for a long time been a research goal in the compiler-writing community. Several efforts to generate compilers from denotational semantics specifications, starting with the SIS system by Peter Mosses 1979, have resulted in compilers and code that run very slowly - usually a factor of 1000 slower than for commercial compilers, and also do not interface to commercial product-quality parsers or optimizing code generators. Another problem has been the poor modularization of denotational specifications, where high-level and low-level aspects, together with static and dynamic properties of language descriptions, are inter-mixed.

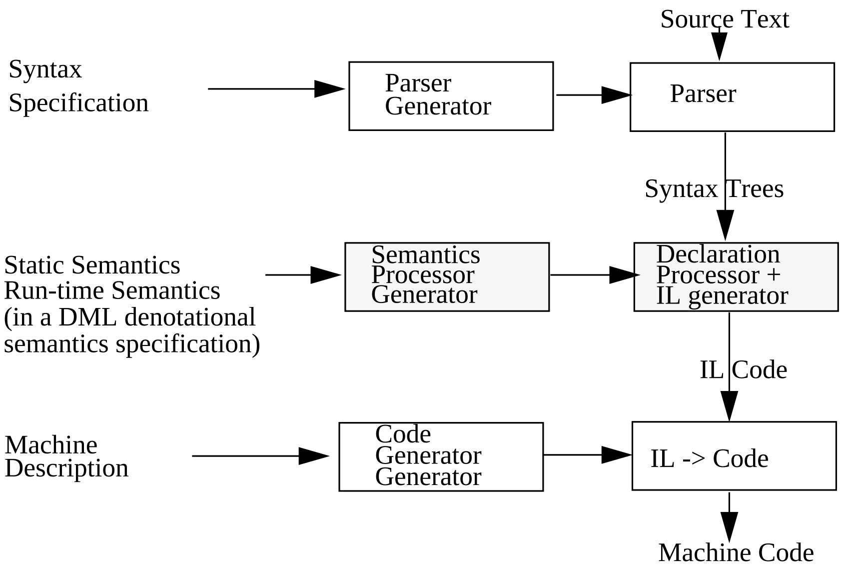


Figure 12-7 *The semantics analysis part of compilers is generated by the DML system.*

Recent work by Peter Lee 1989 has demonstrated that it is indeed possible to generate realistic compilers from denotational semantics definitions. However, his system, called MESS, is monolithic and does not interface well with standard parser generators and code generators. Also the code generator in his system needed to be hand-written in Prolog. In comparison to the MESS system, the DML system presented here goes several steps further. It interfaces well with standard tools written in C-compatible languages and it can automatically generate a code generator for intermediate quadruple code. Also, the DML specification language supports readable and concise specifications through several language enhancements.

The Current DML System

We call the current system and its specification language DML (Denotational Meta Language – essentially a superset of ML). Its specification language can be used for denotational semantics specification, and the system itself can generate compilers that produce efficient code. The method for generating efficient intermediate code was prototyped already in the fall of 1989. The quality of code produced by the generated compiler is close to that of commercial compilers. This choice of ML-like syntax in DML is similar to Peter Lee's approach, and avoids the cryptic syntax of Oxford school denotational semantics.

However, compared to Standard ML, our DML has the enhancement of *BNF rules* allowing in-line concrete syntax with *pattern matching* operators, which simplifies access to structures and gives *shorter and more readable specifications*. DML also provides optional lazy evaluation, a *foreign-language interface* and *efficient compilation to C*. Compilers generated by the DML system can produce quadruple code, which when fed into a standard back-end gives production-quality code.

Small C Example - the function fac:

```
int fac(int n)  {
   if (n==0)
      return 1;
   else
      return n * fac(n-1);
}
```

Mc68020 Code, after feeding generated Quadruples through an optimizing backend:

```
_fac: move1 d2,sp@-
      move1 sp@(8), d2
      seq d0
      btst #0,d0
      jeq L3
      moveq #1,d0
      jra L1
L3:   move1 d2,d1
      subq1 #1,d1
      move1 d1,sp@-
      jbsr _fac
      mulsl d2,d0
      addqw #4,sp
L1:   move1 sp@+,d2
      rts
```

Figure 12-8 *Example of high code quality produced by a generated compiler.*

The implementation of this tool has been done as follows: First a parser and type checker for DML and a DML to Scheme translator have been implemented in Scheme, since Scheme can handle local procedures and closures. Scheme is then compiled to C, using Bartlett's Scheme-to-C translator. To enhance run-time efficiency we have implemented tail-recursion removal for C for the SPARC and MC68020 processors. The final result is a DML to C translator, which can be compiled into C. Finally, DML will be bootstrapped in itself, by rewriting the semantics for DML in DML.

The first version of the DML tool that could generate stand-alone compilers in C was completed during the fall of 1992. There seems to be substantial general interest in this kind of compiler generation tool. For example, when Peter Fritzson visited MIT

1990, giving a talk on this project, Richard Stallman expressed great interest in the future availability of this tool.

12.4.5 Architecture-Independent Parallel Programming and Compiler-Generation for Parallel Hardware

Peter Fritzson, Johan Fagerström, Johan Ringström

Programming of parallel computers is very complicated today. The programmer has to deal both with complexities caused by the parallel hardware in addition to normal implementation details. Also, most of the commonly used programming languages such as C or Pascal require implementation details that constrain compilers too much when generating efficient code for parallel hardware. A better way would be to use higher level languages that conveniently specify operations on collections of data objects rather than single objects, and permit declarative specification of data layout. This would enable the compiler to automatically generate the low-level message-passing and synchronization operations needed in parallel programming.

Our current work in this area is centered around the problem of generating compilers for parallel languages, which also includes selecting suitable data-parallel language constructs and intermediate forms, and experimenting with a small data-parallel language, a modified version of the Predula language mentioned in the section on debugging. We have developed a formal specification of this language and used the DML system to generate a compiler for it, using a two-level approach: the denotational semantics of Predula is expressed in terms of intermediate primitives used to generate final code. These primitives are then defined using structured operational semantics. The denotational specification is expressed in DML (Denotational Meta Language), see the previous section.

The code emitted by the first version of this compiler is a machine independent data parallel vector code (VCode - developed by Guy Blelloch and his group at CMU), for which implementations exist both on the Connection Machine, the Cray, and a simulator on the Sparcstation. A later version of the compiler is instead generating C-code with direct calls to a library of parallel vector operations, the CVL library, which implements the operations executed by the VCode interpreter

The results from the two level semantic approach are encouraging; a small compiler has been generated, which have been used to compile a small example program that was executed on a Sparcstation and a Cray Y-MP. However, we have found that the current stack based VCode abstract machine is unsuitable as a target for compiling imperative languages for efficiency reasons. Thus, we have recently changed the compilation model to generate direct calls to the CVL library mentioned above.

12.4.6 High-level Programming Environment Support for Equational Modeling in Scientific Computing

Peter Fritzson, Lars Viklund, Johan Herber, Niclas Andersson, Joakim Malmén.
In cooperation with Dag Fritzson, SKF.

This project is developing a high-level programming environment for scientific programming that supports programming in equations instead of low-level procedural programming. The high-level equational representation also gives better chances to utilize the inherent parallelism of a problem to generating efficient code for parallel hardware. As an initial example application domain in scientific programming, we have chosen mechanical analysis in machine elements. This work is done in close cooperation with SKF, which enables us to apply the developed programming environment on realistic problems, and get important feedback and suggestions on design decisions and problem solving approaches.

Application domain: modeling in mechanical analysis

At the current state of the art, there are two main classes of activities within research projects or advanced development projects in mechanical analysis:

- Theory development
- Numerical computation

During theory development, a mathematical model is developed that describes the relevant properties of the investigated machine element. This is normally done manually, using only pen and paper.

In order to perform numerical computations, the model must be translated to some programming language, or to input specifications for some existing numerical modeling program, such as for example a finite element analysis program. Normally, existing high level tools can at best be used for limited parts of the total computational problem, because it usually is too complex.

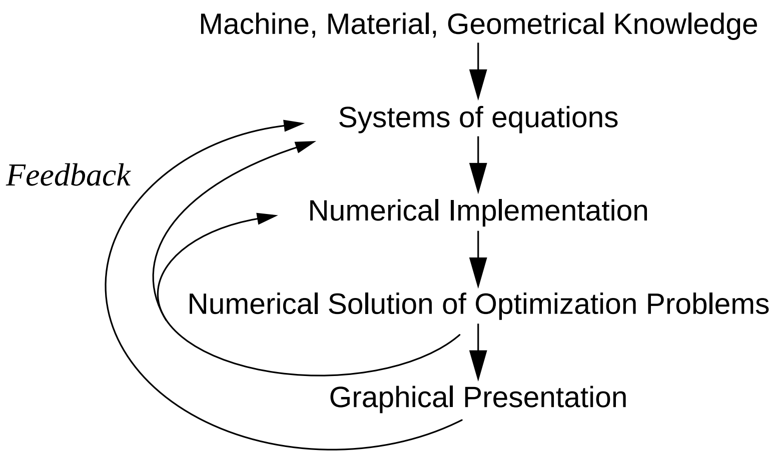


Figure 12-9. *The iterative process of modeling in traditional mechanical analysis. Often 50% of the time is spent on numerical implementation, usually FORTRAN programming.*

The ideal programming environment: programming in high-level equations

The ideal high level programming environment would automatically transform systems of equations into efficient symbolic and numerical programs. It would select optimization routines with good convergence properties for the given problem. The environment would also aid in formulating equations given geometrical constraints and transforming equations between different coordinate systems. However, fully automatic versions of some of these capabilities will be hard to achieve. It is more realistic to assume that the user will work in dialogue with the interactive system, and that the user can supply valuable hints and information that will guide the system to choose the right algorithms and transformations. Some desired capabilities are listed below:

- Support in expressing systems of equations, e.g. handling geometrical constraints and coordinate transformations
- Integration of object-oriented techniques in equational modeling
- Algebraic transformations of equations
- Compilation/transformation of model equations to efficient numerical programs
- Transformation of equations for computation on parallel hardware
- Convergence and selection of optimization methods
- Evaluation of Numerical Experiments
- Graphical presentation

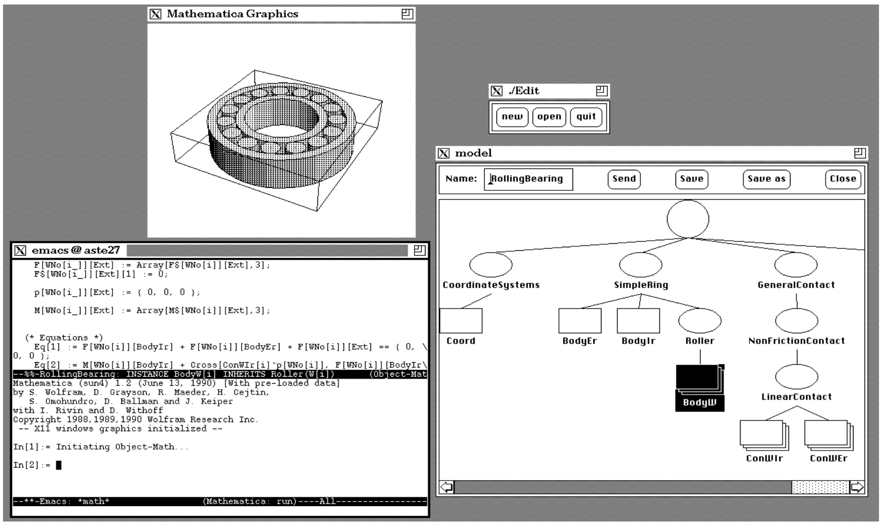


Figure 12-10 *The ObjectMath high-level modeling environment in use on a 200-equation rolling bearing example. The three-dimensional view of the bearing has been automatically generated from equations expressed in ObjectMath.*

The ObjectMath environment and language

The current version of the ObjectMath (Object oriented Mathematical language for scientific computing) programming environment provides several of the capabilities of our ideal programming environment. Using ObjectMath, it is possible to model classes of equation objects, to support inheritance of equations, and to solve systems of equations. Objects are used to structure models that might consist of a large number of equations while inheritance facilitates reuse of equations and formulae. The class specialization hierarchy is also used to generate specialized efficient numeric code from general symbolic equations and expressions.

The ObjectMath environment is designed to handle realistic problems. This is achieved by allowing the user to specify transformations and simplifications of formulae in the model, in order to arrive at a representation which is efficiently solvable. Such algebraic transformations can conveniently be supported since ObjectMath models are translated into the Mathematica computer algebra language. When necessary, equations can be transformed to C++ code for efficient numerical solution.

The first ObjectMath prototype was used successfully to represent and solve a 3-dimensional example model of 200 equations describing a rolling bearing. The re-use of equations through inheritance reduced the model by a factor of two, compared to a direct representation of the model in the Mathematica computer algebra language.The latest version of ObjectMath, released 1992, has been successfully used on several other applications problems at SKF. Some of the ongoing work is on including multiple inheritance and composition into the modeling language, and on graphic visualization of dependencies between equations to help the user in understanding complex mathematical models

Generating parallel code from equations

For a long time efficient use of massively parallel computers has been hindered by dependencies introduced in software through low-level implementation practice. However, by allowing the user to represent mathematical equational models at a high level of abstraction, it should be possible to extract most inherent parallelism from the application.

We have recently added some capability of performing dependency analysis of equations and generation of parallel code to the ObjectMath environment. The dependency analysis is done using a standard algorithm for finding strongly connected components in a graph. The obtained strongly connected components are systems of equations to be solved. So far, a parallel Runge-Kutta solver in the data-parallel language Modula-2* has been generated for a small application problem. Substantial improvements in the generated parallel code are still possible, e.g. by using pipe-line parallelism between the Runge-Kutta solvers for ordinary differential equations. Other types of parallel equation-solvers should also be integrated into the system.

12.5 List of publications

The following are publications by PELAB members during the year 1991-92. For the full list of publications, please refer to Appendix E.

Ph. D. Theses

Nahid Shahmehri: *Generalized Algorithmic Debugging*, Dec 1991.

Licentiate Theses

Mikael Pettersson: *DML - A Language and System for the Generation of Efficient Compilers from Denotational Specifications*, May 1992.

External publications

Niclas Andersson, Peter Fritzson: Comparative Evaluation and Industrial Application of Code Generator Generators, In *Proc. of the Nordic Workshop on Programming Environment Research*, Tampere, January 8-10, 1992. Submitted to the Journal of Systems and Software.

Mikhail Auguston, Peter Fritzson: PARFORMAN - an Assertion Language for Specifying Behaviour when Debugging Parallel Applications. In *Proc. of the Euromicro Workshop on Parallel and Distributed Processing*, Gran Canaria, Jan 27-29, 1993. Proc. published by IEEE.

Johan Fagerström, Peter Fritzson, Johan Ringström, Mikael Pettersson: A Data-Parallel Language and its Compilation to a Formally Defined Intermediate Language. In *Proc of ICCI'92 - the 4:th International Conference on Computing and Information*, Toronto, May 28-30, 1992.

Dag Fritzson, Peter Fritzson, Lars Viklund, Johan Herber: Object-Oriented Mathematical Modelling - Applied to Rolling Bearings. In *Proc. of SCAFI-92 - Studies in Computer Algebra for Industry*, Amsterdam, Nov 5-6, 1992. Also submitted to the Journal of Computers and Structures.

Peter Fritzson, Dag Fritzson: The Need for High-Level Programming Support in Scientific Computing - Applied to Mechanical Analysis. *Computers and Structures*, Vol. 45, No. 2, pp. 387-295, 1992.

Peter Fritzson, Lars Viklund, Johan Herber, Dag Fritzson: Industrial Application of Object-Oriented Mathematical Modeling and Computer Algebra in Mechanical Analysis, In *Proc. of TOOLS EUROPE'92*, Dortmund, Germany, March 30 - April 2, 1992. Published by Prentice Hall.

Peter Fritzson, Dag Fritzson, Lars Viklund, Johan Herber: Transformation of Equation-Based Real-World Models to Efficient Code, Applied to Machine Elements Geometry, In *Proc. of the 1:st National Swedish Symposium on Real-Time Systems*, Uppsala, August 19-20, 1991.

Peter Fritzson, Mikhail Auguston, Nahid Shahmehri: Using Assertions in Declarative and Operational Models for Automated Debugging. Accepted 1992 to the *Journal of Systems and Software*. An abridged version in *Proc. of the fifth International Conference on Software Engineering & its Applications*, Toulouse, Dec 7-11, 1992.

Peter Fritzson, Nahid Shahmehri, Mariam Kamkar, Tibor Gyimothy: Generalized Algorithmic Debugging and Testing. Accepted 1992 to *ACM LOPLAS - Letters of Programming Language and Systems*. A previous version in *Proc. of SIGPLAN'91 Conference on Programming Language Design and Implementation*, Toronto, June 24–28, 1991.

Mariam Kamkar, Nahid Shahmehri, Peter Fritzson: Interprocedural Dynamic Slicing, In *Proc. of PLILP'92 - Symposium on Programming Language Implementation and Logic Programming*, Leuven, Belgium, August 1992. LNCS 631, Springer Verlag.

Petr Kroha, Peter Fritzson: Software Features of an Extended Single Instruction Machine, In *Proc. of the EuroMicro 91 conference*, short note session, Vienna, Sept. 1991. Proc in a special issue of the EuroMicro Journal.

Henrik Nilsson, Peter Fritzson: Algorithmic Debugging for Lazy Functional Languages, In *Proc. of PLILP'92 - Symposium on Programming Language Implementation and Logic Programming*, Leuven, Belgium August 1992. LNCS 631, Springer Verlag.

Mikael Pettersson and Peter Fritzson: DML - A Meta-language and System for the Generation of Practical and Efficient Compilers from Denotational Specifications. In *Proc. of the 1992 International Conference on Computer Languages*, April 20-23, 1992, Oakland, California.

Mikael Pettersson, Peter Fritzson: A General and Practical Approach to Concrete Syntax Objects within ML, In *Proc. of the ACM SIGPLAN Workshop on ML and its Applications*, San Francisco, California, June 20-21, 1992.

Mikael Pettersson: A Term Pattern-Match Compiler Inspired by Finite Automata Theory. In *Proc. of CC'92 - International Workshop on Compiler Construction*, Paderborn, Germany, October 1992. LNCS 641, Springer Verlag.

Johan Ringström, Peter Fritzson, Johan Fagerström: PREDULA - A Multi-Paradigm Parallel Programming and Debugging Environment, In *Proc. of the EuroMicro 91 conference*, Short note session, Vienna, Sept. 1991. Proc in a special issue of the EuroMicro Journal.

Lars Strömberg, Peter Fritzson, Johan Fagerström: Postmortem Debugging of Parallel Real Time Programs, In *Proc. of the 1:st National Swedish Symposium on Real-Time Systems*, Uppsala, August 19-20, 1991.

Lars Viklund, Peter Fritzson: An Object-Oriented Language for Symbolic Computation - Applied to Machine Elements Analysis. In *Proc of ACM ISSAC'92 - International Symposium on Symbolic and Algebraic Computation*, Berkeley, July 1992.

Lars Viklund, Johan Herber, Peter Fritzson: The Implementation of ObjectMath - A High-Level Environment for Scientific Computing. In *Proc. of CC'92 - International Workshop on Compiler Construction*, Paderborn, Germany, October 1992. LNCS 641, Springer Verlag.

Research reports, etc.

Niclas Andersson, Peter Fritzson: Comparative and Industrial Application of Code Generator Generators, LiTH-IDA-R-92-06.

Mikhail Auguston, Peter Fritzson: PARFORMAN - An Assertion Language for Specifying Behaviour when Debugging Parallel Applications, LiTH-IDA-R-92-16.

Johan Fagerström, Peter Fritzson, Johan Ringström, Mikael Pettersson: A Data-Parallel Language and its Compilation to a Formally Defined Intermediate Language, LiTH-IDA-R-92-15.

Mariam Kamkar: An Overview and Comparative Classification of Static and Dynamic Program Slicing. LiTH-IDA-R-91-19.

Mariam Kamkar: Interprocedural Dynamic Slicing Applied to Algorithmic Debugging, LiTH-IDA-R-91-20.

Henrik Nilsson, Peter Fritzson: Algorithmic Debugging for Lazy Functional Languages, LiTH-IDA-R-92-17.

Mats Palmgren: A Visualizing Debugger for Predula. LiTH-IDA-Ex-9214.

Mikael Pettersson: A Term Pattern-Match Compiler Inspired by Finite Automata Theory, LiTH-IDA-R-92-09.

Other references

Mats Andersson: Omola – An Object-Oriented Language for Model Representation. *Lic Thesis*, Dept. of Automatic Control, Lund Institute of Technology, 1990.

Johan Fagerström: *A Paradigm and System for Design and Test of Distributed Systems*, Ph.D. Thesis, Dept. of Computer and Information Science, Linköping University, 1988.

Peter Fritzson: Symbolic Debugging through Incremental Compilation in an Integrated Environment. *The Journal of Systems and Software* 3, 285–294 (1983)

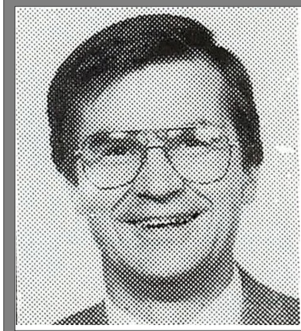
Peter Lee: *Realistic Compiler Generation*, MIT Press, Cambridge, Massachusetts USA, 1989.

R. Seviora: Knowledge-Based Program Debugging Systems, *IEEE Software*, May 1987.

E.Y. Shapiro: *Algorithmic Debugging*. MIT Press, 1983.

RKLLAB

13
The Laboratory for Representation of Knowledge in Logic



Erik Sandewall
director

Artificial intelligence
Real-time systems
Non-monotonic logic
Planning
Autonomous agents
Temporal reasoning
Qualitative physics

Some of the members in the laboratory for Representation of Knowledge in Logic.
Back row: Lars Degerstedt, Tommy Persson, Patrick Doherty, Douglas Busch, Witold Lukaszewicz, Jacek Malec.
Middle row: Christer Bäckström, Anne Eskilsson, Erik Sandewall, Per Österling, Magnus Morin.
Front row: Simin Nadjm-Therani, Dimiter Driankov, Ezra Kim.

Figure 13-1. *Christer Bäckström is handed the hammer and nail by Marianne Kratz to put his doctoral thesis up on the "Doctoral Tree". Magnus Morin receives his licentiate degree.*

13.1 Research area

The area of interest for RKLLAB is theoretical aspects of knowledge based systems. The activity of “knowledge engineering”, or the design of expert systems and other knowledge based systems, is often a rather ad hoc activity. Logic (and discrete mathematics) with suitable extensions, may be applied to strengthening the theoretical basis for knowledge engineering. It is the objective of RKLLAB to contribute in this respect.

13.2 Researchers and Projects

13.2.1 Activities

The activities of RKLLAB during the academic year 1991-1992 have been in the following, overlapping and interacting areas:

Non-standard logics and their implementations, in particular:

1. non-monotonic logic
2. temporal logic, and associated algorithms for temporal reasoning and planning
3. logic of uncertainty, fuzzy logic, and fuzzy control

Plan-guided systems, i.e. autonomous vehicles and autonomous agents, including the following research topics:

1. software architectures for autonomous agents
2. description of physical change in particular using bond-graph methods
3. modelling of the behavior of autonomous agents

Work on plan-guided systems has been done in the framework of several applications areas, in particular the Swedish parts of the pan-European Prometheus project where most of the (West) European automobile manufacturers are cooperating in the development of information technology for future cars. Also some of the work has been done in cooperation with the FOA research institute in Linköping.

13.2.2 Laboratory members

The following researchers have participated in RKLLAB’s activities during the academic year 1991-1992:

Laboratory leadership, administrative and technical staff:

Erik Sandewall, Ph. D., professor
Anne Eskilsson, secretary
Leif Finmo, research engineer

Laboratory members having or completing a Ph.D. degree during the period:

Douglas Busch, Ph. D.
Christer Bäckström, Ph. D.

Brant Cheikes, Ph. D.
Patrick Doherty, Ph. D.
Keith Downing, Ph. D.
Dimiter Driankov, Ph. D.
Jacek Malec, Ph. D.

Visitors during extended periods:

Dr. Jerome Lang	University Paul Sabatiér, France
Prof. Witold Lukaszewicz	University of Warszaw, Poland

Laboratory members having a licentiate degree during the period:

Simin Nadjm-Tehrani
Hua Shu

Other graduate students:

Lars Degerstedt, M. Sc.
Georg Fodor, ABB Marine, Västerås, M.Sc.
Ezra Kim, M. Sc.
Magnus Morin, Tech. Lic.
Tommy Persson, M. Sc.
Lennart Staflin, M. Sc.
Ulf Söderman, M. Sc.
Per Österling, M. Sc.

Papers by RKLLAB members published, or accepted for publication, in international journals or conferences during the same period, are listed at the end of this chapter.

13.3 The practical and the theoretical visions

In order to make progress you must define a point far ahead of you as the target that you are working towards. In RKLLAB we have chosen two such “visions” for the research, one theoretical and one applied.

The applied “vision” is the eventual development of *plan-guided systems*, i.e. a system such as an unmanned vehicle, or an “intelligent” automatic manufacturing cell, which is able to accept an assignment or request, make a plan for how to carry out the assignment, execute the plan, recognize problems which may impede the plan execution, revise the plan if necessary, and report success or failure.

It is clear that A.I. and knowledge engineering techniques are only one part of what is needed for designing plan-guided systems. Automatic control, sensor technology including sensor data fusion, and often computer vision are also needed. Our point is 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 each build their own 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, knowledge based planning, qualitative reasoning, and the logic of uncertainty into a unified theory. In the contemporary A.I. literature these are seen as distinct subfields. We foresee that in the future they will have a common theoretical base, consisting of a *logical theory of change* that includes both “physical change” (change which occurs spontaneously and continuously) and change that is brought about by the actions of agents, operating in the world at hand. Qualitative reasoning hopefully will be based on such a theory of change, or conversely: one of the requirements on such a theory of change is that it must be sufficient as a basis for what we today call qualitative reasoning about physical systems. Similarly, knowledge-based planning should be explainable as a logical operation (abduction, presumably) that is well defined in the theory of change.

The first steps towards a logical theory of change are taken model-theoretically, by defining the logical language and its formal semantics. However from a computer science standpoint, the project will not been completed until we also have developed efficient algorithms and other computational methods for the various kinds of analyses which are needed in such a logic: temporal prediction and postdiction, planning and plan revision, plan recognition, identification of mechanisms and faults in them (i.e. diagnosis), etc.

It is also a reasonable guess, based on current research in this area, that non-monotonic reasoning and the logic of uncertainty should be used in the theory of change.

The theoretical “vision” of a comprehensive logical theory of change, combines naturally with the applied “vision” of plan-guided vehicles and other plan-guided systems. We hope and believe that theoretical progress in this area will be useful in the work towards the applied goal. We feel likewise that experimental work on relatively simple plan-guided systems can also provide useful insights and guidance 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 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 of necessity brief. Please refer to the publications listed at the end of the paper for more details.

13.4 Non-monotonic Reasoning and Partial Logics

Douglas Busch, Lars Degerstedt, Patrick Doherty, Witold Lukaszewicz

During the period reported here, we have pursued an approach to non-monotonic reasoning that distinguishes itself from traditional approaches (circumscription, default logic, auto-epistemic logic) by basing our formalism, NML3, on multi-valued logic and an explicit default operator which permits one to syntactically distinguish between assumptions and assertions. A default rule is represented in NML3 as the formula, $La \wedge Mb \longrightarrow Dc$, which is read as "if a is known and b is possible, then assume c by default". We use a non-monotonic Kleene three-valued logic extended with a weak negation operator (EKL) as the semantic basis for interpreting such formulas. Kripke-like model frames, defined as a pair (Δ, u) where Δ is a set of partial interpretations and u is a member of Δ, are used to represent a partial description of the actual world, u, and its possible alternative informational developments Δ. A preference relation is defined on model frames where the informationally-minimal actual worlds are preferred.

There are a number of interesting theoretical issues which arise in shifting from total interpretations to partial interpretations. We have investigated the relation between the non-monotonic version of the base logic EKL and Gentzen axiomatizations of it. This leads to a very nice three-valued intuitionistic framework for logic programming investigated by Douglas Busch. We have also studied reductions of NML3 to standard classical modal logics. The purpose here is to find a useful basis for developing a decision procedure for NML3 based on analytic tableaux.

In the area of applications, we have focused on two areas involving joint work with Dimiter Driankov, another member of the lab whose activities are described in section 13.6. In the first application, we extended EKL to the four-valued case and used it as a semantic basis for modeling non-standard preference structures of the type used in the area of multi-criteria decision making. In the second case, IF-THEN control rules of the type used in fuzzy control theory were extended to IF-THEN-UNLESS rules, a non-monotonic generalization. In addition, an implementation based on justification-based reason maintenance systems was proposed.

13.5 Efficient Algorithms for Planning and Prediction

Christer Bäckström

This project aims at a better understanding of the computational difficulties underlying action planning and related problems. The primary concern is to identify formally defined, computationally tractable planning problems with applications especially in, but not restricted to, automatic control.

The general research strategy is to work in a bottom-up fashion. Starting with some tractable planning problem, restrictions are removed or replaced such that a new, different or more general, yet tractable, problem is achieved. This process is guided both by theoretical considerations and by studying application problems.

During the past year, the previous results on tractable planning for the SAS-PUBS and SAS-PUS problems have been further generalized to the less restricted, but also tractable, SAS^+-PUS problem. The identification of these problems has been possible by using a variation on the traditional STRIPS formalism, the SAS^+ formalism.

The SAS^+ formalism uses partial states defined by multi-valued state variables instead of sets of propositional atoms. Action conditions are also modelled slightly differently by separating the changed preconditions from the unchanged ones. Although the SAS^+-PUS problem is very restricted it is sufficient to encode a version of the *blocks world problem*. Furthermore, the planning algorithm for the SAS^+-PUS problem can be used as a polynomial-time approximation algorithm for the NP-complete blocks world optimization problem, producing correct, although non-minimal, plans. The SAS^+ formalism is proven expressively equivalent to a number of more traditional formalisms for propositional planning. Hence, restricted versions of the SAS^+ problem can, in principle, be translated into equivalent restricted STRIPS problems. This is not so easy in practice, though, and the shift of formalism has been an essential prerequisite for identifying the tractable problems mentioned above. Furthermore, a complexity analysis of various combinations of the restrictions defining the SAS^+-PUS problem has been carried out. Bäckström defended his Ph.D. thesis on the above topics in September 1992.

Bäckström has also cooperated actively with Dr. Bernhard Nebel (DFKI, Saarbr-complexity on temporal projection, planning and plan validation, and on the relationships between these problems.

The project has been granted funding for another 3 years with Bäckström as project leader and the research will continue along the lines mentioned above. The search for tractable subproblems of the SAS^+ planning problem will continue but other issues in planning complexity will also be of interest. Bäckström will mainly concentrate on the theoretical issues in planning complexity while Klein will focus on applications in automatic control, although the group will work in close cooperation. The cooperation with Nebel is also continuing actively.

13.6 Logic of Uncertainty

Dimiter Driankov, Ezra Kim

During the reported period we have pursued two somewhat different research topics: (1) connections between *formal fuzzy logic* and *non-monotonic reasoning*, and (2) non-monotonic extensions of reasoning with *fuzzy if-then rules*. However, when formal fuzzy logic and fuzzy if-then rules are reformulated in a way which makes them exhibit certain aspects of non-monotonicity this leads to the general problem of *preference modelling*. The latter problem is closely related to the problem of modelling *non-standard preference structures* in *multiple criteria decision making*. In a joint work with Patrick Doherty and Alexis Tsoukias (Politecnico di Torino) we have proposed a semantic basis for the modelling of such non-standard preference structures. In what follows we will briefly outline the work done on all three research topics.

Formal monotonic fuzzy logic uses fuzzy sets to represent extensions of predicates. This has as a consequence that the truth of a predicate belongs to the interval [0,1] rather than the set {0,1}. In this case the underlying logic is a monotonic many-valued one in which the law of excluded middle does not hold. This is due to the presence of a truth-value which expresses ignorance about whether an object has a property or not without rejecting the possibility that it might have this property. This is exactly the type of knowledge used in non-monotonic reasoning systems which allows a fact to be asserted as true by default. We have capitalized on the natural existence of such a truth value when interpreting fuzzy predicates and proposed a novel formalization of a fuzzy non-monotonic logic. The basis of our formalization is the extension of formal monotonic fuzzy logic with two additional truth-functional connectives **M** and **L** where $\mathbf{M}\alpha$ reads as "it may be the case that α is true to a degree" and $\mathbf{L}\alpha$ reads as "it is the case that α is true to a degree". In addition, a default operator $\mathbf{D}\alpha$ is added where $\mathbf{D}\alpha$ is interpreted as "α is true by default". The non-monotonic formal fuzzy logic obtained has an intuitive model theoretic semantics without any appeal to the use of a fixpoint semantics for the default operator. The semantics is based on the notion of *preferential entailment*, where a set of sentences Γ preferentially entails a sentence α, if and only if a preferred set of the models of Γ are models of α. We also show that the logic belongs to the class of cumulative non-monotonic formalisms which are a subject of current interest. This work was done in cooperation with Patrick Doherty.

A fuzzy if-then rule of the form "*if X* is *A then Y* is *B*" is used to represent one particular sample of the causal relationship in the direction from *X* to *Y*. *X* and *Y* are so-called *linguistic variables* which take values defined as fuzzy sets on their respective universes of discourse D_X and D_Y. We constructed a particular extension of a fuzzy *if-then* rule in order to take into account its context dependent nature. In other words, we interpret such a rule as modelling a forward causal relationship between the rule-antecedent (*if-part*) and rule-consequent *then*-part which applies in most contexts, but on occasion breaks down in exceptional contexts. Thus the default nature of the rule was revealed and modelled in a novel way by explicitly augmenting the original rule with an exception part, i.e "*if X* is *A then Y* is *B unless Z is C"*. Furthermore, we construct a proper semantic correlate to the augmented rule in terms of a ternary fuzzy relation which satisfies a number of intuitive constraints, each constraint corresponding to a particular mode of inference. Furthermore, we proposed an implementation of *if-then-unless* rules by the use of a *reason-maintenance system*. In particular, we use a *TMS* where the rules are encoded in the so-called *dependency-net*. We verify that the net satisfies the constraints posed by the different modes of inference they represent. The work was done in cooperation with Patrick Doherty and Hans Hellendoorn (Siemens Corporate R & D, Munich, Germany).

Existing non-monotonic formalisms are generally designed to capture a single criterion for determining preference among competing models. As pointed out by Jon Doyle, one must aggregate individual preference orders into a global order to obtain unified non-monotonic formalisms. As long as one considers preference modeling in terms of classical decision theory, any attempt at aggregating different criteria will violate certain commonsense postulates about preference orderings. Fortunately, recent advances in multiple-criteria decision theory permit the use of preference orderings not limited by the classical constraints of complete comparability and transitivity. The advantage is that one is able to model decision scenarios where information is incomplete and inconsistent. In addition, non-conventional preference modeling may provide a potential solution to the aggregation problem discussed by Doyle. One obstacle towards applying non-conventional preference modeling has been the lack of a sufficient axiomatization for non-conventional preference structures. The main contribution of our work is to propose the use of a multi-valued para-consistent logic, WL4, as a means of providing a robust semantic theory for non-conventional preference modeling.

Other research involves a joint-project with ABB Marine Automation on the design of a fuzzy controller for a low-speed diesel ship-engine, and the preparation of a book "Introduction to fuzzy control" in cooperation with the Laboratory for Intelligent Control Systems at Siemens Corporate R & D.

13.7 Non-Monotonic Logics of Action and Change

Erik Sandewall, Tommy Persson, Lennart Staflin, Ulf Söderman.

The central problem in this research is to identify logically correct methods for drawing conclusions about scenarios in systems with inertia and actions. For example, a system consisting of a number of objects with fixed physical location and a robot, may contain *actions* where the robot e.g.picks up an object or puts two objects together. The system is said to be *inert* if changes in the system can only happen due to actions or other well understood reasons. For example, if the robot moves one object in an inert system then the other objects stay put where they are, and do not move by themselves.

A *scenario* for an inert system is a collection of formulas which describe (1) the character of the system as such, in particular what are the effects of each of the possible kinds of actions, (2) what specific *actions* have been performed or will be performed by the agent, and (3) a number of *observations* of the state of the system at some points in time. The purpose of *reasoning* about a scenario is to draw conclusions about the state of the system at some other points in time (before or after the given observations), or about actions which were not stated in the scenario but which can be concluded to have occurred. A common technical term for research on this problem is "reasoning about actions and change".

Reasoning about actions and change includes both *prediction,* where the conclusions refer to timepoints later than the given observations, and *diagnosis* where the opposite is the case. Planning and plan revision can also be seen as a generalized form of reasoning about scenarios. Such reasoning is important for autonomous agents since logical formulae are a very general and powerful way of expressing standing goals, observations, and general knowledge about the agent's environment. (The latter type of knowledge may have been "taught" to the agent by its master).

State of the Art. The state of the art for reasoning about actions and change can be summarized as follows. It is understood that for inert systems the logic must be non-monotonic, meaning that if additional premises are added then some of the conclusions may have to be retracted. A number of different approaches have been proposed. Some are model-preference approaches, such as chronological minimization and causal minimization, where the intended models are selected using circumscription or other semantic criteria. Others are syntactic approaches, where one performs a formula-level transformation on the given premises (e.g. “explanation closure”), and then obtains the classical models for the modified formulae. In most cases the work is concerned with entirely discrete descriptions of the worlds at hand.

The approaches in the literature are usually defended by their intuitive plausibility and their applicability to a limited number of test examples. There have been some recent (1991) results on correctness proofs for non-monotonic logics for action and change, using a syntactic approach.

Systems with actions and inertia have mostly been studied using logic-based methods. In addition there has been work on so-called *qualitative reasoning* (QR) about non-inert systems, i.e. systems with a discrete level of description which go through successive state changes due to the laws of the underlying physics. The QR research has been quite pragmatic, and has emphasized practical implementation aspects while de-emphasizing logical foundations.

Our contributions. We identify three main groups of results for our logic-based research in this area, performed by Erik Sandewall, Tommy Persson, and Lennart Staflin:

- *Improved entailment methods.* In the period 1988-1990 we identified two methods which were important for increasing the generality of the entailment criteria (= the rules for drawing conclusions), namely *occlusion* and *filtering*. Occlusion is important for cases where an action may or may not result in a change. For example, if an object is dropped on the floor then it may or may not break. Often one can not know in advance what will be the case. Filtering is important for cases where observations are given for arbitrary time-points and not only for the initial time of the scenario.
- *Methods for reasoning about hybrid systems, characterized by continuous change except for occasional discontinuities.* In 1989 we showed how the current methods, which had previously only been described for discrete world descriptions, could be generalized in a natural way to also apply for worlds with continuous time and continuous state variables. The established principle of minimization of change in the discrete case, must then be generalized to minimization of discontinuities. This approach was further refined and improved during the following years.
- *Underlying semantics and formal assessment of entailment criteria*. The reliance on informal criteria and test examples as the support for a proposed logic is not sufficient except as a first step. However there has not been any established methodology for doing better than that. During 1991-1992 we have obtained some new results which change the situation dramatically, and which now allows us to obtain *precise statements* and *formal proofs* for the range of applicability of many of the logics for action and change, including both previously proposed and now constructed logics.

The basic innovation is to define an *underlying semantics* for the logic. The underlying semantics can best be understood as a formal description of the interaction between an agent and its environment, including an abstract description of what are the possible ways of performing each action. Different classes of inert systems can then be defined in terms of their properties in this underlying semantics. Some simple examples of system classes are “deterministic systems” (where an action always has the same effect, if the system state when the action starts is given), and “equiduration-al systems” (where the possible duration time of an action can not depend on the system state when the action starts). The various system classes form together an *ontological hierarchy*. In this framework we have then been able to identify, for each of a number of logics, what is the class of systems where the logic is guaranteed to obtain the intended conclusions.

These results will be published in a book from Oxford University Press, and we believe that it can serve as a much-needed firm foundation for continued work in a number of directions in the future.

In addition there has been work by Ulf Söderman, in cooperation with tekn.lic. Jan-Erik Strömberg in the Division of Automatic Control:

- *Modelling of hybrid systems using bond graphs*. This work addresses the same topic as in the second • item above, but using a different methodology. Bond graphs have been used for a long time in mechanical and control engineering in order to characterize the properties of mechanical systems, and can be thought of as an abstraction from a set of differential equations. However this means that classical bond graphs are not applicable to systems with discontinuities – non-linear systems. We have addressed the problem of generalizing bond graphs so that they can also express mode change and discontinuities in a systematic fashion. Some intermediate results have been published; the full report will appear in the near future as Ulf Söderman’s licentiate thesis.

13.8 A High-Level Data Format for Knowledge Exchange

Erik Sandewall

There is currently a lot of interest in standardized notations and formats for knowledge engineering. In the application for automobile co-driver systems in the Prometheus project, there was a special need to develop a common data format for the *exchange* of pieces of knowledge in real time between “intelligent” software modules. This requirement differs in two ways from what is usually being considered. The real-time aspect constitutes an additional difficulty, but at the same time it means that the requirements on expressivity are not as high as in some other cases. For example if some of the participating software processes are rule-based, then the communication language does not need to express additional rules, but only the facts that are being processed by the rules. The rules themselves are assumed to be stationary in their respective modules or processes, and in this context we did not see the need for standardizing the language for expressing them.

We have developed a proposed syntax for such a high-level format, called PAD for “ProArt Datastructure Exchange Format” or “Portable A.I. Datastructure Format”. PAD has a static (timeless) and a dynamic part. The static part of PAD is essentially a character-based or “ASCII” language for expressing frames or multi-level property-lists, with a fairly rich language for the values that may occur in them. The dynamic part allows to express, for each point in time, what changes of properties or sub-properties take place at that time.

An important design choice for PAD has been to make it as independent as possible of actual programming languages and software packages. We first wanted to have a clear and agreeable definition of the data format itself, and only then proceed to the implementation of a library of support software. The first phase of the work has been completed and reported.

13.9 Autonomous Real-Time Systems

Erik Sandewall, Jacek Malec, Magnus Morin, Simin Nadjm-Tehrani, Hua Shu, Per Österling.

There is presently an increasing interest in combining several existing approaches to the design of real-time and real-world computing systems, i.e. systems such as autonomous robots which operate in the physical world under real-time constraints. Such systems have been studied since long in several research contexts, including *artificial intelligence* (intelligent robots, autonomous agents), *formal specification methods* for software (where specification of real-time systems is a particularly important issue), and *real-time operating systems* (addressing in particular the hard-core software kernels).

One can divide the class of problems encountered in the research on autonomous systems into two subclasses: *analysis problems*, dealing with description either of existing, real-world created or engineered systems, or of conceived agents intended to perform complex tasks in the real world, and *synthesis problems*, concerned with designing a system given its description. Our interests span over both areas and include such topics as representation languages and techniques for description of autonomous systems, reasoning about such systems and their behaviour, formal specification and verification techniques for autonomous systems, and software architectures for implementing real-time agents.

The very first question one can ask is: What languages should be used to speak about autonomous real-time systems? Is logic (possibly temporal) a satisfactory, or even preferable tool for all purposes, or should one use some other techniques, such as visual languages, behaviour languages, situated automata models, etc.? In each of those languages the questions of consistency and completeness of description arise, so each such language should be provided with appropriate verification procedures. Another important element is the possibility of expressing real-time related dependencies and properties in the language.We are studying several such languages, mostly logics, but also logic-based visual languages.

In designing the complete autonomous vehicle, or other plan-guided real-world systems, the choice of an adequate software architecture is a crucial issue. It is very important from a conventional software engineering perspective; it is also very important if we hope to combine contributions from A.I., automatic control, sensor technology and others into a coherent system.

Although we feel that the logic-based methods described above could be very useful for the design of a plan-guided system, the software work does not *only* have the goal to implement the given theoretical methods. The emphasis is instead on identifying a software architecture which will satisfy the needs of all the participating technologies, including the logic-based A.I. contributions.

One significant difference is between the "specification" perspective and the "engine" perspective on software. The former case emphasizes the application specific nature of software, where each application needs to be taken through the stages of formal specification, implementation, and verification. Systematic methods for verifying the correctness of a proposed implementation relative to a given specification, and for generating the implementation automatically are then of paramount importance.

In the "engine" perspective one emphasises a software architecture consisting of a limited number of general purpose programs, such as temporal map manager, inference engine of a rule-based system, and so on. The correctness of these software engines is important, but can presumably be established once and for all. In this case the logical and implementational properties of a system need only to be analysed relative to the engine's specification. We are working on establishing algorithms for such analysis.

The group has developed a multi-layer software architecture as an answer to these questions. The lowest layer is 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. Above the process layer engine a supervisory rule layer has also been implemented as an engine. Besides them, there are various tools which make it possible to specify particular applications in high-level languages, to modularize them and also analyse their performance parameters.

The architecture has been implemented for several target systems. The first one is based on a Motorola 68000 processor and is using the $PSOS^+$ real-time operating system. The second is based on a distributed system consisting of a series of nodes connected by the CAN bus, where each node may use arbitrary kind of hardware and software. The implementation is based on an IBM PC-compatible machine with the Micro O'Tool/VDX real-time software environment. A Sun workstation, connected to a target system by a special protocol, is used as the development and simulation vehicle.

The major application area in our research is within the Driver Assistance and Local Traffic Management (DALTM) project within the Swedish RTI programme, a pre-competitive research cooperation project by the Swedish automobile manufacturers and government agencies. 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 DALTM 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 neighbouring cars can be understood to behave. The model of traffic behaviour is needed as a basis for plan recognition, identification of dangerous situations, planning (planning the route, or planning how to handle a danger), etc.

Several approaches to traffic behaviour modelling are being used. They include logic-based approaches along the lines described above, a state-space based approach, and an approach which views each car control/driver support system basically as a reactive agent.

13.10 International activities

RKLLAB's activities have a very international flavour. More than half of the participants in this year's work are foreign nationals. Like the foregoing years, we participated in the Eureka project Prometheus. RKLLAB also participates in two Esprit BRA activities that were approved during 1992, namely the project "DRUMS II" and the working group "Logic and Change".

13.11 Publications

RKLLAB Dissertations 1991-1992:

Christer Bäckström: Computational Complexity of Reasoning about Plans, 1992, No 281.
Patrick Doherty: NML3 - A Non-Monotonic Formalism with Explicit Defaults, 1991, No 258.

International RKLLAB Publications1991-1992:

Douglas R. Busch: An Intuitionistic Three-valued Logic: presented at *Logic Programming in AI First Compulog Net Workshop*, Imperial College, London, March 1992.

Christer Bäckström, Inger Klein: Parallel Non-Binary Planning in Polynomial Time: In *proc. of IJCAI-91*, pp. 268-273, Sydney, Australia, Aug. 24-30, 1991.

Christer Bäckström, Inger Klein: Planning in Polynomial Time: the SAS-PUBS Class: in *Computational Intelligence* 7(3), 1991, pp. 181-197.

Christer Bäckström, Bernhard Nebel: On the Computational Complexity of Planning and Story Understanding: In *proc. of ECAI-92*, pp. 349-353, Vienna, Austria, Aug. 3-7, 1992.

Christer Bäckström: Equivalence and Tractability Results for SAS+ Planning: In *proc. of KR-92*, pp. 126-137, Cambridge, MA, USA, Oct. 26-29, 1992.

Christer Bäckström: Uncertainty and Parallelism in Action Structures: In *notes of ECAI-92 Workshop: Beyond Sequential Planning*, Vienna, Austria, Aug. 2, 1992.

Patrick Doherty, Witold Lukaszewicz: FONML3 - A first order Non-Monotonic Logic with Explicit Defaults: In *proc. of ECAI-92*, pp. 294-298, Vienna, Austria, Aug. 3-7, 1992.

Patrick Doherty, Witold Lukaszewicz: Defaults as First-Class Citizens: In *proc. of the IEEE ISMVL-92*, May-92, Sendai, Japan.

Patrick Doherty: A Constraint-Based Approach to Proof Procedures for Multi-Valued Logics: In *proc. of WOCFAI-91*, Paris, France, July 1-5, 1991. Also to appear in *Studia Logica*.

Patrick Doherty, Witold Lukaszewicz: Distinguishing between Facts and Default Assumptions: In *Non-monotonic Reasoning and Partial Semantics,* W. van der Hoek, J. J. Ch. Meyer, Y. H. Tan and C. Witteveen (eds.), Ellis Horwood, 1992.

Patrick Doherty, Dimiter Driankov: Non-Monotonicity, Fuzziness and Multi-Values: In *proc. of IFSA-91*, Brussels, Belgium, 1991.

Patrick Doherty, Dimiter Driankov: Non-Monotonicity, Fuzziness and Multi-Values: In *Fuzzy Logic: State of the Art,* B. Lower and M. Ruebens (eds.), Kluwer, 1992.

Patrick Doherty, Dimiter Driankov: A Non-Monotonic Fuzzy Logic: In *Fuzzy Logics for the Management of Uncertainty,* L. Zadeh, J. Kacprzyk (eds.), Wiley, 1992, pp. 171-191.

Patrick Doherty, Witold Lukaszewicz: NML3 - A Non-Monotonic Logic with Explicit Defaults: In *J. of Applied Non-Classical Logic*, 2(1), 1992, pp. 9-48.

Patrick Doherty, Dimiter Driankov, Alexis Tsoukias: Partiality, Para-consistency and Preference Modeling: In p*roc. of Int'l. conf. on Economics/Management and Information Technology,* Tokyo, Japan, Aug. 31-Sept. 4, 1992.

Patrick Doherty, Dimiter Driankov, H. Hellendoorn: Fuzzy If-Then-Unless Rules and their Implementation: In *proc. of IPMU-92,* Palma de Mallorca, Spain, July, 1992.

Keith L Downing, L-E Widman: Extending Model-based Diagnosis to Medicine. In *proc. of the Qualitative Reasoning Workshop*, Austin, Texas, USA, 1991.

Keith L Downing: Model-Based Diagnosis of Qualitative Physiological Models. In *proc. of the European Qualitative Reasoning Workshop*, Milan, Italy, 1991.

Keith L. Downing: Physiological Applications of Consistency-Based Diagnosis: To appear in *Journal AI in Medicine*, Dec. 1992.

Keith L. Downing: Consistency-Based Diagnosis in Physiological Domains: In *proc. of AAAI-92*, pp. 558-563, San Jose, CA, USA, Jul. 12-16, 1992.

Keith L. Downing: Extending the GDE Paradigm to Medicine: In *proc. of Applications of AI X: Knowledge-Based System*, Orlando, Florida, Apr., 1992.

Keith L. Downing: Analyzing Genetic Regulatory Systems via Focused Envisionment: In *proc. of the Qualitative Physics Workshop*, Aug.,1992.

Dimiter Driankov, H Hellendorn: Towards a Logic for a Fuzzy Logic Controller: *Proc. ECSQAU-91*, P. Krause and P. Siegel (eds.), *Lecture Notes in Computer Science*, vol. 548, Springer Verlag, pp. 166-171.

Dimiter Driankov: Fuzzy Reasoning and its Applications to Control: In *proc. of ESSCIRC-91*, Milano, Italy, 1991.

Dimiter Driankov and H Hellendorn: Fuzzy Logic with IF-THEN-UNLESS Rules: In *proc. of the 1992 IEEE Int'l Conf. on Fuzzy Systems*, pp. 255-262, San Diego, USA, Mar. 28 - Apr. 1, 1992.

Jacek Malec: Complex Behavior Specification for Autonomous Systems: In *proc. of the IEEE ISIC-92*,Glasgow, Scotland, Aug. 11-13, 1992.

Jacek Malec: How to Pass an Intersection or Automata Theory is Still Useful: In *proc. of SCAI-91*, Roskilde, Denmark, 1991.

Jacek Malec: Process Transition Networks: What are they for? In *proc. of the IEEE International Conference on Systems, Man and Cybernetics*, Charlottesville, VA, Oct. 13-16, 1991.

J. Malec: Applied Knowledge Representation. In *CC-AI: The Journal for the Integrated Study of Artificial Intelligence, Cognitive Science and Applied Epistemology,* Vol. 9, No. 1, pp 9-41, 1992.

Magnus Morin, Simin Nadjm-Tehrani, Per Österling, Erik Sandewall: A Framework for Real-time Hierarchical Control in Autonomous Agents: in *IEEE Software, special issue on real-time systems development*, Sept., 1992.

Simin Nadjm-Tehrani, Per Österling: Characterization of Environment Conditions with Metric Temporal Feature Logic: In *proc. of the 3rd IEEE International Conference on AI, Simulation and Planning in High Autonomy Systems*, Perth, Australia, Jul. 1992.

B. Nebel, C Bäckström: On the Computational Complexity of Temporal Projection, Planning and Plan Validation. Accepted for publication in *the journal Artificial Intelligence.* To appear.

Bernhard Nebel, Christer Bäckström: On the Computational Complexity of Temporal Projection and some Related Problems: To appear in *Artificial Intelligence*.

Erik Sandewall: Causal Qualification and Structure-Based Ramification: In *proc. of the Symposium on Commonsense Reasoning*, Austin, Texas, Jan. 1992.

Erik Sandewall: Parallel Actions, Conditional and Random Outcomes, Surprises, and other Complications when Reasoning about Actions: In *notes of ECAI-92 Workshop: Beyond Sequential Planning*, Vienna, Austria, Aug. 2, 1992.

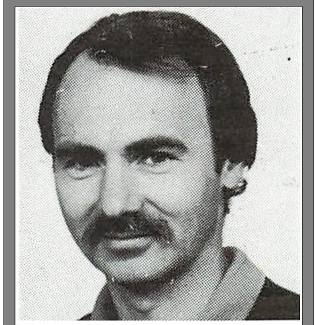
Erik Sandewall: Knowledge-based Systems, Lisp, and Very High Level Implementation Languages, appeared *in The Knowledge Engineering Review*, Vol. 7:2, 1992, pp. 147-155.

Hua Shu, Jacek Malec: From Process Transition Networks to Behaviour Automata: In *proc. of ISIC-91*, Arlington, VA, Aug. 13-15,1991.

Ulf Söderman, Jan-Erik Strömberg: Combining Qualitative and Quantitative Knowledge to Generate Models of Physical Systems: In *proc. of IJCAI-91*, pp. 1158-1163, Sydney, Australia, Aug. 24-30, 1991.

Figure 13-2. *External doctoral students in Systems Development. From the left, Owen Eriksson (University College of Borlänge), Göran Goldkuhl (group leader), Anders Avdic (University College of Örebro), Kenneth Åhlgren (University College of Örebro), Roger Lindqvist (University College of Växjö), Malin Nordström (University College of Östersund), Bengt Andersson (Swedish National Audit Bureau, Stockholm).*

VITS



Göran Goldkuhl
director

14 Information Systems and Work Contexts

Systems development as organizational change
Contextual activity modeling
Information systems architecture
Evaluation of information systems
CASE tools and adaptation of ISD methods

Some of the members in the group for Information Systems and Work Contexts.
Back row: Göran Goldkuhl, Christian Krysander, Dan Fristedt, Stefan Cronholm.
Front row: Annie Röstlinger, Ulrika Laurén, Karin Pettersson, Anna Brolin.

Figure 14-1. *VITS Autumn seminar, November 16-17 with 30 Swedish researchers in Information Systems Development.*

14.1 Introduction

The research group VITS is a newly formed lab within the department. The group was planned during 1990 and established during 1991. VITS research area is information systems development. We are taking a contextual approach emphasizing organizational, social and human aspects of IS and their development.

Some important aspects of our research orientation:

- collaboration with practice
- methods development
- empirical studies, mainly qualitative methods used

14.2 Overview of current research

The research is performed in the following projects:

- Principles for method adaptation into CASE tools
- Contextual evaluation of information systems
- Strategies for information systems architecture
- Governmental inter-organizational information systems
- Change analysis
- Contextual activity analysis of information systems

14.2.1 Principles for method adaptation into CASE tools

The existence of new computerized tools (CASE) for information systems development (ISD) means a computer implementation of the earlier paper and pen methods. Old methods are put into CASE tools. Many marketed CASE tools seem to contain the same old standard methods, like e.g. data modelling and data flow diagramming. How about other methods? If an organisation has a method of their own and want to use a CASE tool - what shall they do? Must they program a tool from scratch?

There exist different software products which make it much easier to build a CASE tool for a specific method. Such tools can be called Meta CASE tools, CASE shells, CASE generators or customizable CASE environments. These tools are design tools for generating CASE tools and in such a CASE tool implements a desired ISD method. Examples of tools with CASE shell properties are Excelerator/Customizer, Ramatic, VSF and MetaEdit.

What are the properties of such CASE shells? How hard is it to implement a method into a CASE tool based on a CASE shell? What parts of a method are appropriate/possible to implement? What is the character of the process of method adaptation (customization)? What are the properties of the customized CASE tools? What are the motives behind using a CASE shell approach? When is such an approach appropriate?

These are important issues for research. In the research project PRIMCASE (Principles for method adaptation into CASE tools) we are studying these issues. We are investigating properties and relations of

- different ISD methods
- different CASE shells (customizable environments)
- method customization processes into CASE tools
- customized CASE tools

The area of interest is depicted in figure 14-2 below.

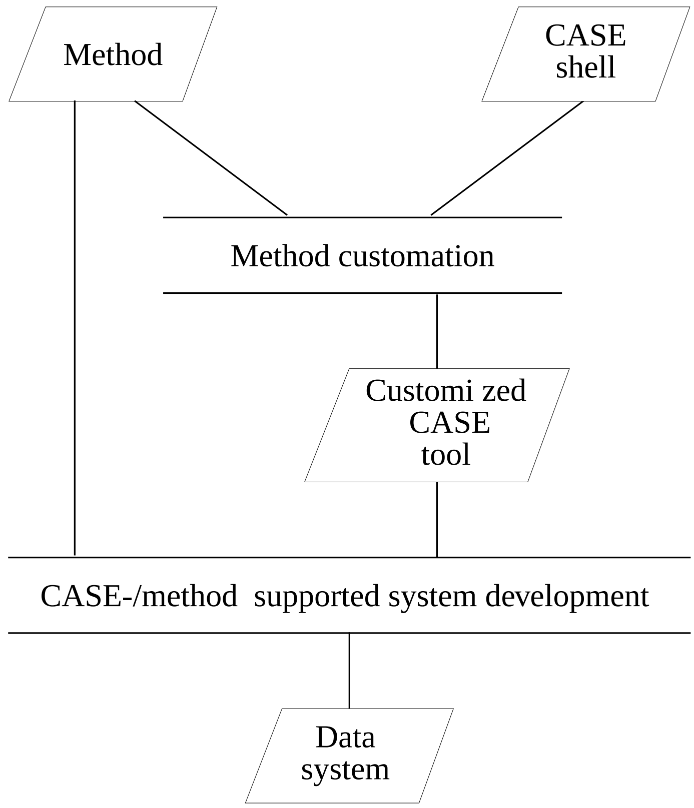


Figure 14-2. *Method customization context. .*

One rationale behind the PRIMCASE project is to safeguard and improve the Scandinavian method tradition. Many approaches in this tradition put emphasis on

- simple methods to enhance user participation
- the social and work character of information systems (activity oriented methods).

This is in contrast to American methods which have a more restricted focus and are more technically oriented. Most marketed CASE tools contain such methods. CASE shells are important to support future use and development of Scandinavian methods. CASE shells are also important to improve the user influence (by systems developers) on their systems development tools.

We have been investigating different CASE shells and their properties. We have also performed some tests with CASE shells, i.e. we have made customizations. Three different CASE shells have been used for three customizations. We have used the same systems development method in all three customizations in order to get comparable tools. We have used our own method for contextual activity modelling of information systems (CONTACT); se below.

The study of CASE shells and customized CASE tools are based on an explicit evaluation model. The model is based on an elaborated conception of ISD work supported by methods. We emphasize information systems development as a modelling and design work. The model consists of several evalution criteria. These criteria have been used when we have evaluated CASE shells and customized CASE tools.

Results from the project has been published in Goldkuhl (1991), Goldkuhl (1992d) and Goldkuhl, Cronholm, Krysander (1992).

The research is funded by The Swedish National Board for Industrial and Technical development (ITYP Program).

Researchers: Göran Goldkuhl (project leader), Stefan Cronholm, Christian Krysander, Dan Fristedt

There is a reference group to this project with industry representatives (CASE users, CASE vendors and method consultants). The project has an established cooperation with the Swedish Institute for Systems development (SISU) and the research group METAPHOR at Jyväskylä University.

14.2.2 Contextual evaluation of information systems

Evaluation of information systems can be performed in different ways. One can study purely technical aspects of information systems. One can also study issues of information quality and information contents. There can be broader studies of the information system; how the system is a functional part of business processes. This kind of evaluation can be characterized as contextual and activity oriented.

The aim of this research project is to develop a methodology for contextual evaluation of information systems. We have delimited our research at the moment to municipal information systems. The contextual methodology consists of

- activity description including the action role of an IS
- analysis of strength, opportunities and potential
- analysis of problems and difficulties
- analysis of goals and goal fulfilment
- analysis based on general quality criteria

The development of methodology is based on our earlier methods for change analysis and activity analysis (see below). We aim at simple and powerful analysis and description techniques in order to be used by non-experts.

We also study evaluation in relation to change of information systems. Evaluation can be performed as a basis for change. Change can mean a larger development of an IS.

It can also mean ordinary maintenance of IS. A contextual analysis of IS change is published in Goldkuhl (1992c).

In May 1992 the VITS group together with the Swedish Computer Association arranged a conference concerning IS change (LISS92). In this conference, among other papers, results from this research project were presented (cf. publication list).

The development of the evaluation methodology will be performed in close relation to action research projects in different municipalities. In these projects VITS researchers participate together with IS users evaluating information systems in a contextual way.

As a prestudy to this research we performed an interview investigation among some municipalities concerning evaluation and change of information systems. This study has been reported in Röstlinger (1992) and Röstlinger & Pettersson (1992).

This research is funded by The Swedish Association of Local Authorities.

Researchers: Annie Röstlinger (project leader) and Anna Brolin.

14.2.3 Strategies for information systems architecture

The topic for this research project is structure and relations between different information systems in an organization (information systems architecture). There are different ways for how to structure IS. One approach is IRM (Information Resource Management). Information is seen as a common resource in the organization. Large shared data bases are established and distinguished from applications. These data bases are independent of organizational structure. The structure of information determines the structure of the IS. Another approach is a functional structuring of the IS. Each IS is allocated to a business function (organizational unit) which has full responsibility for its IS. The information systems in this approach are autonomous, but with a defined exchange of information (a federated approach). This approach has been elaborated by the Swedish researcher and consultant Mats-Åke Hugoson and is in Swedish called VBS (VerksamhetsBaserad Systemstrukturering).

In this research project we study and compare these two strategies (IRM and VBS) for information systems architecture. A preliminary study of these strategies was published in Goldkuhl (1992b). We are performing empirical case studies of organizations using each strategy. There are two main research questions: 1) Are the ideal architectures possible to implement? 2) What properties and consequences (if implemented) do these architectures have? Concerning this question we are interested in IS functionality, IS change, IS transparency and IS responsibility.

This research is funded by The Swedish Council for research in the Humanities and Social Sciences.

Researchers: Göran Goldkuhl (project leader), Karin Pettersson.

14.2.4 Governmental inter-organizational information systems

In one project we are cooperating with Swedish National Audit Bureau concerning governmental inter-organizational information systems. Earlier studies (Andersson & Nilsson, 1992) has shown that there are severe problems in this area: deficiencies in information quality, unclear responsibility and roles between different governmental authorities both in development, use and maintenance.

The project aims at the development of a better conceptual and descriptional model for governmental inter-organizational information systems. Researcher: Bengt Andersson in cooperation with other VITS researchers.

14.2.5 Change analysis

The first phase of ISD is often called feasibility study, pre-study or survey or something like that. Such a phase involves an investigation if the proposed system is considered feasible. Both in method and in practice it seems that IS development often is taken for granted when performing feasibility study.

When an analysis activity (as e.g. feasibility study) is defined as a part of ISD, people probably direct their thinking towards IS solutions. People tend to think in terms of information systems when discussing organizational goals and problems. It is probably hard to drop ISD as the main solution. The problems are already conceived to be of IS character.

We challenge this kind of traditional life cycle ISD initiation. As an alternative we have been working with the concept of change analysis for several years.

The rationale for change analysis can be described in the following way: A development of an information system must be seen as an organizational change. As such it is a possibility among others. Before starting an ISD process there should be a conscious and separate process of investigation and decision. There should not be any IS bias in this process; like system analysts looking for and at organizational problems as possible computerizations. A problem diagnosis should be performed with as little solution bias as possible. Different alternatives should be formulated and evaluated before a choice is made. This is a process of analysing different possible changes in an organization starting from a problematic situation. Change analysis means a phase that is performed before ISD and before any decision on ISD is made. The result from change analysis can be a decision to develop an information system. But it can also result in decision on other kinds of change measures.

A methodology for change analysis was initially developed by Göran Goldkuhl and Annie Röstlinger in the early 80'ies. A book was published in Swedish ("Förändringsanalys" by Studentlitteratur) in 1988. This method has been used in education and practice for many years now. The method is called SIM in Swedish and internationally we call it CONTACT (Contextual activity analysis). The method consists of five analysis areas:

- problem analysis
- goal analysis
- activity analysis
- analysis of change requirement
- determination of change measures

The research is at the moment directed towards gathering empirical experiences from practical use of the method. We have written a paper concerning experiences from problem analysis (Goldkuhl & Röstlinger, 1992).

This research is partly funded by the Faculty of Arts and Sciences, Linköping University.

Researchers: Göran Goldkuhl, Annie Röstlinger, Ulrika Laurén

We have cooperation with the MDA group at the department concerning change analysis. They use parts of this method in their methodological and empirical work.

14.2.6 Contextual activity analysis of information systems

There are different approaches to functional modelling of information systems. To mention two well-known methods: Structured Analysis (SA) with data flow diagramming and the isac approach with activity graphs. These methods are top down approaches (hierarchical decomposition) and using graphical notation.

There are some problems using such methods: Anomalies in hierarchical decomposition. Describing IS as a unified whole. Insufficient description of the dynamics of IS in relation to organizational activities.

Since several years we are working with an alternative to such methods: Contextual activity modelling of information systems. This method is centered around Action Diagrams; a notation for contextual analysis and description of organizational activities and information systems.

Contextual activity modelling emphasizes the information system as part of an activity pattern. IS should give support to users performing different tasks in the organizational activities. One main principle behind Contextual activity modelling is: Computer-based and manual functions should together form a coherent and congruent pattern of activities. This activity pattern is described in Action Diagrams. The main descriptional elements are action/task with doer (actor or IS) action objects (information and material objects). The semantics of Action Diagrams makes possible descriptions of both communication (”information flow”) and handling of material objects (”material flow”). The dynamics of activities are described in terms of Action Logic. Action Logic means modelling of sequences, alternatives, conjunctions, triggers, interruptions, conditions and parallelism. When using Action Diagrams the IS is not described as a unified whole. The IS is described as several functions and in several places according to dynamics of the activities. The different IS functions (actions) are descriptively contextualized. This means that each IS function are described in the specific activity context where it shall ”take place”. This means that IS is not described as a unified whole as in the methods mentioned above.

Contextual activity modelling with Action Diagrams is one part of the change analysis method described above. It is also a main element in the contextual and activity oriented approach to information requirements analysis that we are working with. This method belongs to the language action tradition of ISD.

Contextual activity modelling with Action Diagrams have been described in Goldkuhl (1992a,b) and Goldkuhl & Röstlinger (1991).

Researchers: Göran Goldkuhl, Annie Röstlinger

14.3 The group

Our organizational ambition is to provide a stimulating research environment with a lot of cooperation with practitioners and other researchers. We want to be a source of inspiration for others. So far we have succeeded to interest several persons in other organizations to cooperate in research and participate in graduate studies related to our research.

Group leadership and administrative staff:

Göran Goldkuhl, Ph. D., associate professor
Eva-Britt Berglund, secretary

Employed graduate students:

Anna Brolin, B. Sc.
Stefan Cronholm, B. Sc.
Christian Krysander, M.S c.
Karin Pettersson, B. Sc.
Annie Röstlinger, B. Sc.

Associated graduate students:

Dan Fristedt, B. Sc.
Ulrika Laurén, B. Sc.

Associated external graduate students:

Bengt Andersson, Swedish National Audit Bureau
Anders Avdic, University College Örebro
Owen Eriksson, University College Falun/Borlänge
Roger Lindquist, University College Växjö
Malin Nordström, University College Östersund

14.4 Publications

Papers at research conferences:

Andersson B, **Nilsson P** (1992) How to manage complexity in inter-organizational information systems - some preliminary conclusions. To be presented at IFIP 8.3/8.5 Conference “Decision support in public administratoin”.

Goldkuhl G (1992a) Contextual activity modelling of information systems, in *Proceedings of ”3rd int working Conference on Dynamic Modelling of information systems”*, Noordwijkerhout.

Goldkuhl G (1992b) On the relations between information systems and organizational activities: integration and separation as thought models, in *Proceedings of 15th IRIS* (Information systems Research In Scandinavia), Oslo university.

Goldkuhl G, **Cronholm S**, **Krysander K** (1992) Adaptation of CASE tools to different systems development methods, in *Proceedings of 15th IRIS* (Information systems Research In Scandinavia), Oslo university.

Goldkuhl G, **Röstlinger A** (1992). An import aspect of change analysis. Joint elicitation of problems. To be prented at IFI 8.2 Conference “Information systems development – organizational, social and human aspects”, May, 1993.

Submitted papers:

Cronholm S., **Goldkuhl G.** (1992). Meanings and motives of method custimization in CASE environments – observations and categorizations from an imperical study.

Fristedt D., **Goldkuhl G.** (1992). Possibilities and restrictions of CASE shells – leranings from a comparative study of three customizations in CASE enviroments.

Papers at practice conferences:

Goldkuhl G (1991) Stöd och struktur i systemutvecklingsprocessen, Konferens ”Systemutveckling i praktisk belysning”, Dataföreningen i Sverige.

Goldkuhl G (1992c) Att förändra informationssystem - en verksamhetsinriktad grundsyn, Linköpings informationssystemSeminarium 1992 (LISS 92).

Goldkuhl G, **Röstlinger A** (1991) Att bygga in verksamhetskvalitet i informationssystem, Konferens Sundsvall42, Dataföreningen i Sverige.

Goldkuhl G., **Cronholm S.** (1993). Customizable CASE environments. To be presented at COPE’IT 93, June, 1993.

Röstlinger A (1992) Några problem vid utvärdering och förändring av datorstödda kommunala verksamheter, Linköpings informationssystemSeminarium 1992 (LISS 92).

Other papers:

Goldkuhl G (1992d) Metodanpassning av CASE-verktyg, Institutionen för datavetenskap, Universitetet i Linköping.

Goldkuhl G (1992e) Verksamhetsinriktad forskning om informationssystem: några reflektioner om samspelet IS-forskning - IS-praktik.

Goldkuhl G (1992f) Kunskapande.

Röstlinger A, **Pettersson K** (1992) Utvärdering och förändring av datorstödda kommunala verksamheter - rapport från en förstudie bland kommuner i Östergötland.

15
CENIIT – The Center for Industrial Information Technology

Since1988, the School of Engineering at our University has funded a special research programme on *Industrial Information Technology.* This decision 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 a very important enabling technology for other industries. The term ‘industrial information technology’, that was adopted by the School of Engineering, refers to the second aspect above.

The work in the *Center for Industrial Information Technology, or CENIIT*, started in 1988. Government funding is currently in the order of 8 MSEK per year. The departments for computer science, electrical engineering, physics and mechanical engineering are presently the main participators, but the programme may eventually involve most of the departments in the School of Engineering.

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 use of information technology for product development and production planning*, for instance in connection with computer-aided design and process planning.

In addition, activities in the following research areas may start if funding increases:

- Process and product control for environment and energy optimization, e.g. control of a conventional combustion engine
- Industrial documentation technology, including also issues of hypermedia technology, ‘intelligent handbooks’, natural language processing, machine translation etc.
- Visualization, graphical image technology and multimedia, e.g. for scientific/ technological calculations, cooperative work and virtual realities for training simulation

The total number of projects 1992 is about 16, of which IDA is engaged in 5.

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

The inter-disciplinary research programme of CENIIT is coordinated and planned by a steering committee, with representatives from different departments and also from industry.

The current committee consists of:

Jan-Ove Palmberg, professor, Department of mechanical engineering (chair)
Tore Gullstrand, senior industrial leader
Sture Hägglund, professor, Department of computer and information science
Lars Nielsen, professor, Department of electrical engineering
Kjell-Håkan Närfeldt, Telia Research AB
Lars-Göran Rosengren, Volvo AB
Christer Svensson, professor, Department of Physics and Measurement Technology

The following projects (most of which are further described elsewhere in this report) are currently undertaken as our department's contribution to CENIIT.

15.1 Computer Support for Autonomous Manufacturing

This project started during the spring of 1989 when Dr. Anders Törne was recruited from ABB Corporate Research in Västerås. The project is now one of the core projects within CAELAB (*the Laboratory for Computer Assistance in Engineering*). The research is oriented towards the study of architectures and representations needed in systems for computer support in the engineering process. Presently this incorporates task level programming, real time architectures for supervisory control, programming of autonomous manufacturing environments, and real time systems. The research takes an interdisciplinary approach and involves cooperation with a robotics research group at the department of Physics and Measurement Technology.

Area leader: *Anders Törne*

15.2 Engineering Databases

This area is concerned with basic aspects of database research, which are relevant for database support in design, development and maintenance of (large) technical systems. In particular database support for mechanical engineering, knowledge engineering and software engineering. Central research issues concern effective methods for management of complex structures in the database, including object-oriented approaches and conceptual modelling of database contents.

Important concepts are distribution, heterogeneity, active databases, and databases in real time systems. Applicative projects are also conducted in cooperation with industry; one on object-oriented data modelling and one involving knowledge based techniques. The project provides a base for the Database group within CAELAB (*the Laboratory for Computer Assistance in Engineering*). Professor Tore Risch started his work in the laboratory in march 1992.

Area leader: *Tore Risch*

15.3 Autonomous Real Time Systems

This CENIIT-project started in January 1992. The project considers issues related to the problem of designing an autonomous system capable of acting in the real world. In the work reactivity, predictability and ability to pursue goals are focused and the work is mainly concerned with the specification of behaviour and the software architectural solution.

The work is performed within RKLLAB *(the Laboratory for Representation of Knowledge in Logic)* and is connected to the Prometheus project.

Project and area leader: *Jacek Malec, Erik Sandewall*

15.4 Algorithms for Geometric Contact Detection

This project concerns the development and experimental evaluation of algorithms for contact detection. The problem arises in simulations and control of mechanical systems, e.g., simulations of vehicle collisions and mechanical forming, collision detection in robotics applications and for the avoidance of unintentional penetration in CAD. The project started in july 1992.

The project is a cooperation project between researchers at the department of Mechanical Engineering and ACTLAB *(the Laboratory for Complexity of Algorithms)*.

Area leader: *Per-Olof Fjällström*

15.5 Conceptual Text Representations for Automatic Generation and Translation

The purpose of this project is

- to develop a representation of the content and structure of text which is language-independent and possible to integrate with the logical representation of the objects and relations in the domain from which the topics of the text are taken
- to develop parsers, generators and translation systems that employ such representations.

The project works with restricted forms of texts taken from manuals in Swedish and English. In-depth studies of restricted text types contribute to the general goal while also having potential applications in improved techniques for the production of documentation of various kinds.

The project is conducted within NLPLAB *(the Natural Language Processing Laboratory)*.

Area leader: *Lars Ahrenberg*

15.6 Industrial Software Technology

This project is concerned with the development of methods and technical support for software engineering, in particular industrial-scale software systems. Work in this area was initiated by Bengt Lennartsson, when CENIIT activities started in 1988. One main focus of studies is on empirical accounts of experiences from real-world development processes, in particular aspects of reusability, modifiability, testability, quality assurance, etc. Such studies are carried out in joint projects with companies like NobelTech Systems AB, Ericsson Radio Systems AB and FMV (Försvarets Materielverk).

This project is presently associated with ASLAB *(the Application Systems Laboratory*). During the last year, Bengt Lennartsson has been succeded by Kristian Sandahl as research coordinator. The financial support from CENIIT to Bengt Lennartsson's project terminated by July 1st, 1992 and activities in the area are for the time being carried on with support mainly from the industrial partners.

Area leader: *Kristian Sandahl*

Figure 15-1. *Project leaders within CENIIT.*
Back row: Sture Hägglund (member of the steering committee), Lars Ahrenberg (project leader), Per-Olof Fjällström (project leader), Anders Törne (manager and project leader), Erik Sandewall (project leader), Kristian Sandahl (project leader).
Front row: Tore Risch (project leader), and Jacek Malec (project leader).

16
Educational Programmes

Undergraduate education in Sweden is at the moment organized in study programmes. Each is made up of a set of courses from different departments and subject areas. Figure 16-1 below shows the higher levels of degrees in the Swedish university system.

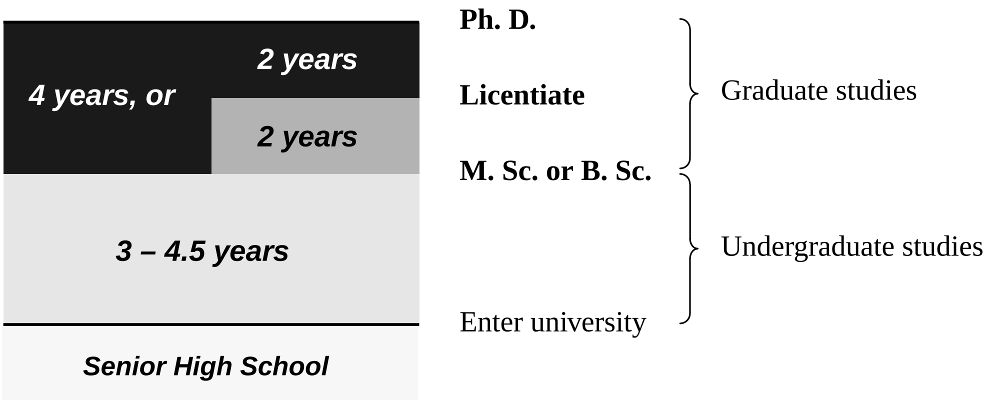


Figure 16-1. *Levels of degrees in Sweden.*

Each undergraduate student is assigned to a *study programme:* e.g. mechanical engineering, computer engineering, computer science, international economy, or system analysis, taking from three years to four and a half. All the students in such a study programme take the same courses (with minor exceptions) during the first two years, and have a free(-er) choice from the third year onwards. All of the programmes which fit into the structure shown by the figure above lead to a degree from where admission to graduate studies is allowed. During recent years a shorter variant of engineering education has also been established. These programmes are only 2 years of length; they are more oriented toward practical work than theoretical studies and does not give admission to graduate studies.

The role of the department in all of these study programmes is to sell and deliver courses to the study programme committees. Those committees have a position between departments and faculty boards and are composed of representatives from departments, student corporations, and industry. The courses given by our department during 1991/92 for the undergraduate education, up to and including the master’s degree level, are listed in Appendix C.

The graduate study programme provides extended studies *from* the level of M. Sc. and B. Sc., *to* licentiate and/or Ph. D. degrees. The graduate students are accepted and cared for by the department. A summary of recent courses and seminars in the graduate study programme is given in Appendix B, together with a presentation of faculty engaged in research and graduate education.

16.1 Graduate education

Graduate studies in the department of Computer and Information Science are organized as a program consisting of courses and project participation. The course program is organized at the department level and consists of *regular courses*, each of which is given approximately every second or third year (if possible), and *occasional courses* which depend on the profile and interests of current faculty and visiting scientists. Thesis projects are always done within or in association with the laboratories or research groups.

Courses and seminars are normally given in English (unless all participants are fluent in Swedish). Licentiate and Ph. D. theses are (with few exceptions) written and defended in English. Thus English is the working language of the graduate study programme.

The programme leads to one of the following degrees:

Licentiate of technology or philosophy. The requirements include 50 points (one point equivalent to one week full time studies) of completed courses and 30 points thesis work. For a licentiate of technology, a master of engineering ('civilingenjör', 4.5 years of study) is normally assumed as a prerequisite.

Doctor of technology or philosophy. The requirements are 80 points courses and 80 points of thesis work. Most of the Ph. D. students take the licentiate degree as an integral part of their doctoral studies.

Although formally not part of the graduate study programme, a *Master of Science* degree is also offered within the department. For this degree, 40 course points are required, where a selection of courses from the undergraduate study programme is also eligible. The thesis work corresponds to 20 points.

About 100 students participates in the graduate programme, and they may choose between about 20 courses given each year. The research committee of the department, headed by prof. Sture Hägglund, is responsible for the organization and implementation of the graduate programme. As executive there is one director of graduate studies (forskningsstudierektor). However, most of the administration and organization rests upon a secretary of research (Lillemor Wallgren), who also organizes courses and activities for graduate students, e.g. a course in Written English, meetings and advice regarding oral presentation techniques. Most graduate students are employed by the department, full time. Their responsibilities comprise, for example, assisting in undergraduate courses and other internal assignments of the laboratories, up to about 15 - 30% of their time. The rest of the time must be spent on courses and their own research.

About 6 - 10 of the students and about 25 % of the teachers in the graduate programme have foreign citizenship or origin, which makes the programme activities very international and this also makes English the language of the programme. On the other hand only about 10 of the students are female and this is definitely a figure to improve for the future.

16.1.1 Fields of study and degree subjects

The program is divided – whether it is a licentiate degree or a Ph. D. – into two main areas:

The *computer science field* including the degree subjects: computer science and computer systems, comprising areas of study such as computer architecture, VLSI, programming languages, artificial intelligence and databases. Studies in this field have an engineering emphasis and a technology orientation. Students are assumed to have a strong background in mathematics and mathematical logic.

The *information science field* including the degree subjects: library and information science, economical information systems, engineering information systems and computational linguistics, comprising areas of study such as system analysis and system analysis tools, specification of demands on and effects of computer technology, cognitive science, natural language processing, engineering databases and information resource management.

The division between these fields is, however, not clear-cut, and many courses may be taken by graduate students from many degree subjects. Each subject has a course profile in which a number of courses are specific and compulsory. Each student chooses courses after consultation with his or her supervisor.

The following degree subjects are presently offered in the department:

Computer Science *(Prof. Erik Sandewall, Prof. Sture Hägglund, Prof. Jan Maluszynski, Prof. Tore Risch)*

Computer Systems (*Acting Prof. Krzysztof Kuchcinski)*

Computational Linguistics (*Associate Prof. Lars Ahrenberg)*

Economic Information Systems (*Prof. Birger Rapp)*

Engineering Information Systems *(Prof. Tore Risch)*

Library and Information Science (*Prof. Sture Hägglund, Prof. Erik Sandewall)*

16.1.2 Admission to graduate studies

Normally a undergraduate degree is required to be admitted, from one of the following areas:

- computer science or computer engineering (CSc, CSy, CL, LIS, EngIS)
- M. Sc. in engineering subject (EIS, EngIS)
- economics (EIS)
- library science (LIS)
- system analysis (LIS, EIS)
- mathematics with emphasis on computer science (CSc, CSy, CL, LIS, EngIS)

- a similar approved qualification, supplemented by at least 40 or 60 points in computer subjects.

Some of these listed degrees are incompatible with certain subjects, as is indicated within parenthesis above. A person with a different academic background may be admitted after supplementary studies and/or may be granted an exemption.

Applications for admission to graduate studies are invited twice a year. A decision is later taken by the department's research committee. Each person admitted as a graduate student is assigned a supervisor, whose duty is to give instructions and advice concerning the further organization of studies. Normally an advisory committee, including two assisting advisors is formed during the first year of graduate studies. If a student later chooses a field of study or speciality different from the one represented by the supervisor, he or she may change to another supervisor.

The task for the advisory committee is to

1. follow the progress of the graduate student and at least once a year assemble for a meeting where the student present his plans and results. The presentation should be followed by a discussion, where the advisors comment upon the thesis project. The student is responsible for scheduling the meeting and inviting the committee.
2. support the main advisor and contribute to an equal treatment and judgement of graduate students in the department. Advisors in the committee may or may not engage in a more frequent and active guidance of the student, depending on individual considerations.

The main supervisor for a graduate student has the full and formal responsibility for approving the thesis subject, courses and other issues in connection with the studies. The advisory committees are consultative.

16.1.3 Teaching faculty

The teaching staff, consisting of those teachers at the department who are graduate course leaders, together with the supervisors and project leaders for graduate students, are responsible for the curriculum of the graduate studies programme. They have, of course, completed graduate education themselves, and the majority of them devote most of their time and attention to research of their own, apart from the graduate education and research supervision.

The teaching staff is complemented by other external teachers, who are employed by another departments or equivalent, and who also lead courses or projects or participate in the planning of the graduate study programmes in computer and information science.

16.1.4 Course Program

The course program is divided into subject areas with an associated group of faculty taking responsibility for courses being given within each area. The teachers are not seldom guest researchers visiting the department for a shorter or longer period. Outside the program, other courses are also given to a small extent, usually due to unforeseen circumstances. The course program is decided every year by the Research Committee. For the last years courses totalling about 80 points have been given each year, i.e., twice as much as a student normally take (“full time”).

The graduate study programme is presently revised - partly due to the expansion of the department.

Course areas and persons to contact:

Area	*Contact person*
Research Methodology	*Sture Hägglund*
Administrative Data Processing and System Analysis	*Göran Goldkuhl*
Economical Information Systems	*Birger Rapp*
Hypermedia	*Toomas Timpka*
Algorithm Analysis and Complexity Theory	*Per-Olof Fjällström*
Artificial Intelligence	*Erik Sandewall*
Natural Language Processing	*Lars Ahrenberg*
Human-Computer Interaction	*Jonas Lövgren*
Philosophy in Computer Science	*Richard Hirsch*
Psychology in Computer Science	*Nils Dahlbäck*
The Mathematical Foundations of Computer Science	*Jan Maluszynski*
Logic	*Erik Sandewall*
Logic Programming	*Jan Maluszynski*
Programming Languages	*Anders Haraldsson*
Computer Architechture/VLSI	*Krzysztof Kuchcinski*
Expert Systems	*Sture Hägglund*
Information Retrieval	*Roland Hjerppe*
Programming Environments	*Peter Fritzson*
Compiler Technology	*Peter Fritzson*
Databases	*Tore Risch*
Distributed Systems	*Johan Fagerström*
Real time Systems	*Anders Törne*
Software Engineering	*Sture Hägglund*

Lists of courses are given in an appendix.

16.1.5 Seminars

Seminars are considered very important for giving the graduate students a broad knowledge in different field of Computer and Information Science. The exposure to different views of research, and different problems in the broad field, nurses the natural curiosity of a good scientist. By observing and participating in the seminars the abilities of the graduate student is strengthend. It is therefore expected from graduate students to go to invited seminars and to licentiate and doctorate dissertations, disregarding research specialities.

A list of seminars is given in chapter Appendix B.

16.2 Undergraduate education at IDA

Undergraduate education accounts for roughly 35% of the total budget of the department, including a very small rate of continuing education and technology transfer activities. (Approx. 25 MSEK together.)

A couple of years ago we went trough a reorganization of the undergraduate teaching activities within the department. The new structure has been effective from July 1990 on, and it has also been evaluated to prove efficient to serve its purpose.

16.2.1 Organization of the undergraduate education

The department is responsible for the formal subject areas computer science (datalogi) and computer systems (datorsystem) in the School of Engineering, and administrative data processing (administrativ databehandling) in the School of Arts and Sciences. The external structure which shows a clear distinction between the university's School of Engineering on the one hand, and it's School of Humanities and Sciences and on the other hand, is however invisible inside the department. We deliver a set of courses, and i principle any teacher could be assigned any course, no matter what study program committee have ordered it.

Under the department board, there is the Undergraduate Teaching Committee (IDUN – IDA's undervisningsnämnd), headed by Olle Willén. The committee is responsible for the contents of courses given by the department and the assignment of teachers to courses. There are representatives from the student unions in this committee as well as teachers and researchers.

Below this committee there are five executive directors of studies (studierektorer), and there is one secretary, Britt-Marie Ahlenbäck, who is in charge of all the common activities concerning undergraduate education. The 'subject areas' that the five executive directors of studies are heading have been composed from the course content point of view. Out of the big offer of courses within the Computer Science and Systems Analysis areas we have tried to find manageable sets of courses with much in common, or well related to each other, to form these subject areas. The subject area Economic Information Systems that was transferred to IDA during the autumn of 1990, has been moved almost unchanged into the IDA framework, but is well integrated into our structure. Much of the responsibilities laying upon the Teaching Committee are delegated to the executive directors. For instance they staff the courses, hire students to assist in training, lead the upgrading of courses, and, of course, acts as the manager for his group of teachers. Each director is also supported by a secretary who handles much of the administration. The organization is presented in Figure 16-2, and the courses of the subject areas are listed in Appendix C. In the next five chapters You will find more detailed descriptions of the contents and efforts of the five subject areas.

Most of the teachers and lecturers have an association both to a research lab and to one, sometimes more, of the five areas for undergraduate education. This means that nearly all courses we give have some connection to the research activities within the department. The Teaching Committee has also started a program which allows those teachers who are not otherwise associated with a research lab to go through courses and direct their interest towards research activities. This is done by reducing the teaching portion of their positions.

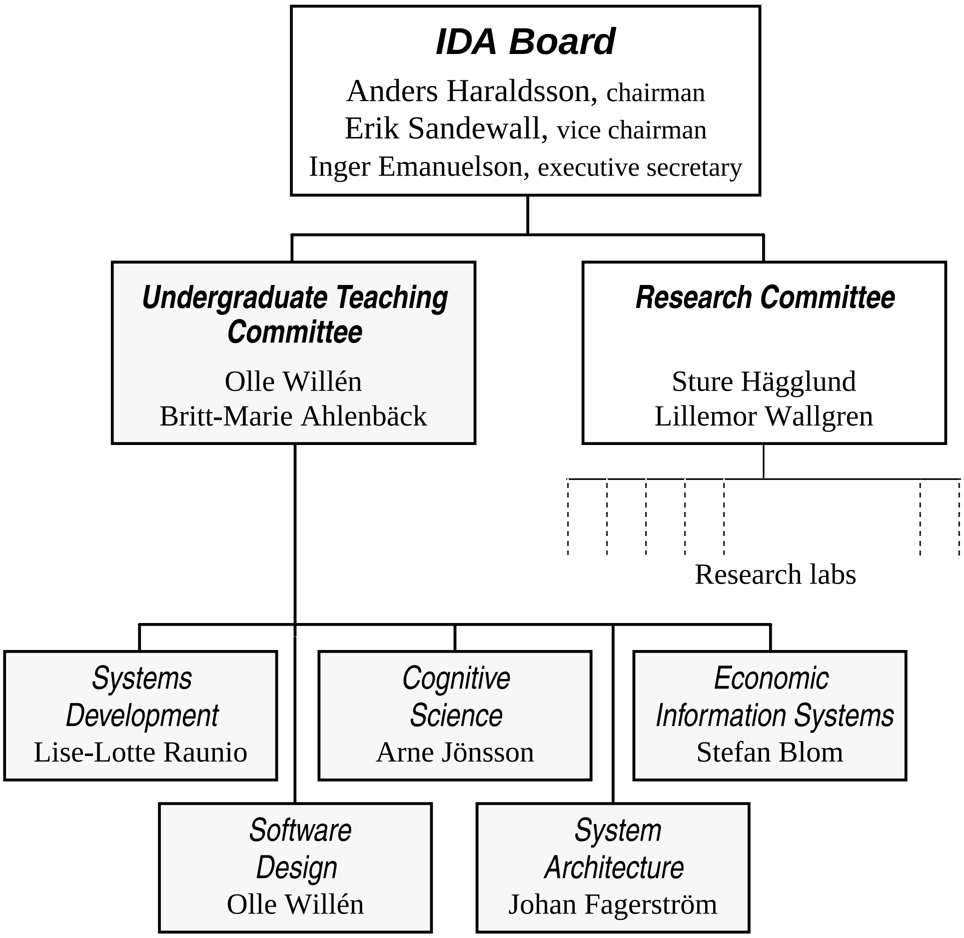


Figure 16-2. *The strategic and executive responsibilities for undergraduate education at IDA 1991/92.*

16.2.2 Undergraduate curricula

Since 1975 Linköping University has had a strong position in undergraduate curricula and teaching in computer science. Linköping is today the only Swedish university offering the three main 3–4.5 years undergraduate study programs in the area of computer science (C, D) and systems analysis (SVL) plus the 2 years computer engineering programme (DI). The volume of other educational activities, such as a continuing education programme in computer science for industry, varies a lot between years, and has for the last years been on a low level.

As the first institute of technology in Sweden we started the D study line (Computer Engineering – ’Datatekniklinjen’) in 1975 as a four-year (now converted to 4.5 years) programme leading to a Master of Engineering. It was the first full and specialized programme in computer science, specialized on software and hardware. The programme was introduced in 1982 at all other Swedish institutes of technology. Linköping University was also the first to introduce the I-line (Industrial and Management Engineering – 'Industriell ekonomilinjen') in 1969. This programme has also been copied by other institutes of technology, more than a decade and a half after it was introduced in Linköping, however.

About 30% 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 now.

A new computer science programme, the C-line (Computer Science – 'Datavetenskapliga linjen') was started in 1982. It is a four-year programme leading to a Master of Science degree. The number of students accepted annually is 30. The programme is also given at Uppsala and Umeå Universities. At Linköping this programme is organized within 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 logic, 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

Thus, a strong characteristic of the C-line is its broad coverage of cognitive science combined with computer science. It is quite clear that these students develop a different 'culture', and in particular a more solid basis for graduate research in computer science, than students from our other lines do. While certainly our other lines will continue to be of very high importance, the computer science line has provided a significant addition.

The D-line has also passed through changes. The new computer science base with discrete mathematics, logics and programming beginning with LISP instead of Pascal is introduced there. An advantage is that the students from the C- and D-lines get the same basis and we expect a large number of students from the D-line to be better prepared to specialize their studies in both more theoretical computer science areas and in artificial intelligence. In the D-line there are possibilities for specialization during the last years. In particular a new set of courses prepare students for software engineering assignments, i.e. the process of developing and maintaining large software systems in an industrial environment, as well as qualitative aspects of software products.

The set of courses that are available in the other programmes has been extended, and many of the courses have been improved. Technically, this has often been done by making new courses from the computer science curriculum available to other lines as well.

Industrial and Management Engineering, the I-line, was introduced in 1969 on an initiative from industry and business. It has steadily grown and adapted to meet new demands ever since. The transfer to IDA of the subject area Economic Information Systems including a full professorship, Birger Rapp, and a dozen researchers, teachers, and graduate students, will strengthen the links between IDA and the I-line. Today the students can specialize in either electrical or mechanical engineering. A new branch towards economic control, the controller function, and information management, would be welcome. The subject area Economic Information Systems is a natural bridge between computer science and its applications in business, industry, and organizations.

In the mechanical engineering programme there is a specialization that combines mechanical and computer engineering. We believe that especially research in artificial intelligence will be significant within that specialization.

The SVL programme, (Systems analysis – 'Systemvetenskapliga linjen'), has recently been modernized and ranges over three years of full-time studies. It aims at professional activities of design, evaluation and implementation of computer-based information systems. ADP-systems analysis dominates the program but nevertheless great importance has been attached to other subjects in order to give the program the necessary breadth and also to ensure that the students will become aware of the complexity of the community where computers can be used.

The first two years of the SVL programme constitute a common core of basic studies for all students. Within the subject of ADP-systems analysis there are courses in systems development and systems theory as well as courses in programming and computer science. The courses about systems development and systems theory deal with formal methods and prototyping. For the programming courses Pascal has been chosen as the main language but, other languages are taught as well. Within the field of computer science the students take courses in database design, development of interactive systems, communication, evaluation of computer systems, programming methodology, etc. Other subjects given within the common core of basic studies within the SVL programme are:

- business economics and management, to get basic knowledge about the organization of corporations and public services and their 'commonday' routines.
- human factors, industrial and social psychology, including ergonomics, work environment, co-determination and participative management, group dynamics etc.

There are also courses in practical Swedish language for professional use, social science, mathematics and statistics. The second year ends with about five months of on-the-job training.

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

- development of computer programs and program systems (program development) aimed at program development, methodology, and technology.

- development of information systems (systemering), aimed at methodology for design and evaluation of information systems.

Both SVL specializations end with a term-paper reporting the development and implementation of an individual project.

The transfer of the subject area Economic Information Systems to IDA offers new opportunities for the development also of the SVL programme.

From 1993/94 on the structure of education in the School of Arts and Sciences will take a new and quite different form, and this will of course also effect the SVL programme. The purpose in short is to break down the lengthy, cohesive and compulsory programmes into one basic block followed by a wider degree of free choices of courses and finished by a deepening in the specific subject. The efforts to design this new structure have already started, but will continue over the year to come.

As a response to the demand for a non theoretic, more practically oriented engineering education a number of shorter engineering study programmes have recently been introduced into the Swedish universities. They are two years of length and lead to the non graduate degree of ’University Certificate in Engineering’. You can discover a slight tendency of trying to prolong these studies with one year, but in Linköping such a possibility have not yet been formalized. All six of these programmes that are located to Linköping University have at least one basic programming course included. The goal of the DI-line (Computer Engineering - ’dataingenjörslinjen’) is to educate students to get very handy with practical systems programming, i e to perform service and maintain the software of computer installations. Thus their main programming language is C, and the students practice this language in a lot of different courses. The studies are not oriented towards hardware, and so IDA give about 65% of all the courses in this programme.

Though we give courses in a vast number of different programmes, be believe that the most important education from the computer and information science point of view are the C, D, SVL, and DI programmes. Students within these study lines spend much of their time, sometimes most of their time, taking IDA-courses, so we strongly feel that we have a great responsibility for their education and personal development.

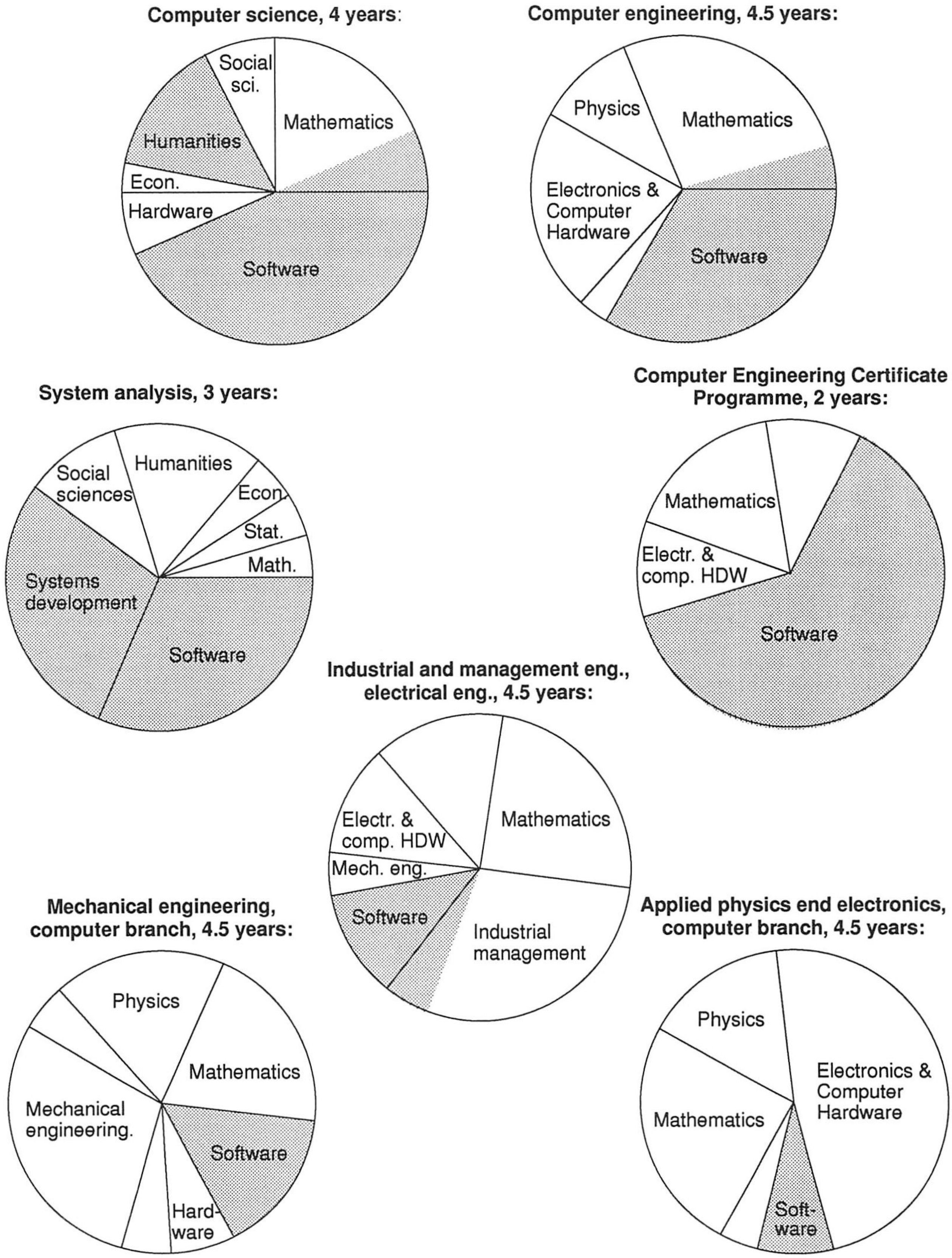


Figure 16-3. *The department share (shadowed) of the different study programmes.*

16.3 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 often given as academic courses and give academic credits after normal examination. They are organized for half-times studies and are given in such a way that the participants are free for studies 2 days a week with one full-day teaching and one day for reading and exercises of their own.

During the years up to 1990 we were engaged in a lot of courses of this kind, which we delivered to the Swedish industry. E g a 25-points programme in AI/expert systems was given to ASEA Brown Boveri and to Volvo, and a large Programming Methodology block was given at LM Ericsson.

For the last two years the demand for this kind of activity has been very low. We believe that during the ’bad times’ the industry in general have been looking not for academic continuing education, but rather for practical education, and in this area we can not compete. However we continuously try to expose our competence and possibilities in a paper which has been written for that purpose.

There has anyhow been some small scale projects in this field:

The county employment board (länsarbetsnämnden) in Östergötland ordered a series of courses for a handful of immigrants with an engineering education as background. At IDA we gave two of these courses, namely *Programming and Computer Science, basic course* (5 points), and *Management Control* (3 points).

We also gave a 3 days course for 10 people at Saab Missiles in Linköping called *VHDL Hardware Description Language*. The course introduced basic VHDL concepts for hardware description. The structure modelling, timing modelling as well as behavioural modelling techniques were discussed. Finally, the course had one day laboratories to give a practical inside to the field.

Systems Development

17 Subject area: Systems Development

Lise-Lotte Raunio
director

Back row: Mikael Johansson, Eva-Britt Berglund, Andreas Björklind, Mikael Lindvall, Peter Fritzson.
Front row: Stefan Cronholm, Lisbeth Björklund, Göran Goldkuhl, Roland Hjerppe, Lise-Lotte Raunio, Eva-Chris Svensson, Birgitta Franzén, Karin Pettersson, Annie Röstlinger.
(All teachers engaged in the undergraduate programme in Appendix C).

17.1 Overview

The subject area Systems Development mainly involves methodologies for activity-oriented development of information systems. One method, 'Change Analysis', is intended to provide students with a thorough knowledge of the impact of change in an organisation: before any change is undertaken it is necessary to analyze the consequences of a change, eg. on the staff, working routines, duties, organizational structure. When it has been decided what to change there are methods how to evaluate and develop information systems as well. In respect of CASE-tools, on which we give courses, this subject area belongs to the 'upper Case'. But we also give very popular project courses dealing with the whole process of systems analysis, data modelling, program design and implementation of a working system.

The area consists of about 30 courses of 1200 students.

Half of them are courses in the SVL programme, in the ADP-systems analysis section. This programme covers most of the subject area.

Then in order (about a third) we have self-contained courses (courses are chosen for one term at a time) in ADP-systems analysis. These subjects are similar to the SVL courses given at the Department of Computer and Information Science . Students may take these kinds of courses up to 60 points and attain a B. A. degree majoring in ADP.

This year the Laboratory for Library and Information Science (LIBLAB) has started another self-contained course, Electronic Media; techniques, use and consequences, 10 points. The first step in plans to increase the scope of this subject has been successfully realized within the introduction of the Information Resources in Society course.

Another research laboratory, People, Computers and Work (MDA), is currently developing the course, Information Systems in Health and Medical Care, which will be given as an independent course next spring.

The remaining courses are mainly given to the students majoring in Computer Science and Computer Engineering. They deal with Systems Development, CASE and a practical Software Engineering course focusing on the management of a programming project.

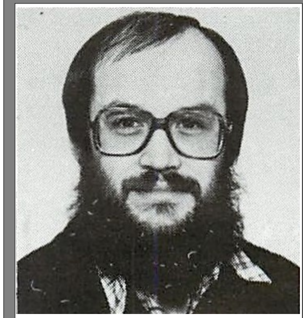
More annual statistics about the area:

- About 5000 student points are taken
- 6.5 full-time teachers are involved in the courses
- The students spend 23 000 schedule hours at the terminal (PC and workstation)

Figure 17-1. *Lecture in session.*

Software Design

18 Subject area: Software Design



Olle Willén
director

From back: Martin Sjölin, Thomas Padron-McCarthy, Erik Stoy, Christian Krysander, Stefan Cronholm.
Andreas Björklind, Jonas Wallgren, Anders Haraldsson, Andreas Kågedal.
Torbjörn Jonsson, Jonas Persson, David Andersson, Martin Sköld.
Per-Olof Fjällström,Barbara Ekman, Katarina Sunnerud, Ulf Nilsson, Olle Willén, Rober Bilos.
(All teachers engaged in the undergraduate programme in Appendix C).

Figure 18-1. *Marking examination papers. Janne Petersson, Jonas Persson, David Andersson och Niclas Wahllöf.*

18.1 Overview

All students undertaking any of the programmes at the School of Engineering have to take a *basic programming course* within their first year of studies (sometimes the second year). This also applies to the students attending the Computer Engineering Certificate, the Systems Analysis (SVL), the ADP, the statistics and the mathematics programmes. These basic courses are of varying length and use different programming languages, but all of them are given by the subject area of Program Design. The number of students attending these courses is appr. 800 each year, and this figure is one of the characteristics for this area, i e education of great volume at primary levels.

Another characteristic is the fact that all the *programming languages* that are taught by IDA, with different objectives in different study programmes, belong to courses within this area. The first language that students learn is in most cases Pascal, but LISP (the C- and D-lines) and C (the DI-line) are also used for this purpose. Of course we also offer a number of courses where Ada is the main objective, but this is usually done in a context where imperative languages as a paradigm are studied. Other languages introduced to support this view are C (with C++) and Simula. Specialized courses have the intention of teaching Logic Programming in PROLOG, Object Oriented Programming in C++, Administrative Programming in COBOL and even to give an overview of Fortran and Algol 68.

Those student that go on from this introductory level usually have to take a course in *Datastructures and Algorithms* as their next one. This is definitely a prerequisite for all programmes that lead into further depth of Computer Science. A set of such courses are also included in the area of Program Design, but they are of different nature, more or less mathematically oriented, depending on the level and the purpose of the education in question. For those students who are deeply engaged in this subject there is a possibility to take a course in Design and Analysis of Algorithms at the very end of the studies.

At IDA, we regard *project work* as an important means of exercising program development and the practical use of programming languages. Performed in groups, project work also has the advantage of training students to work in a social context and experience the difficulties and possibilities of cooperation. Many of the basic courses within the Program Design area has smaller project works as a natural and integrated ingredient. Besides a number of courses are especially designed to take care of this matter, and they also include certain aspects of Software Engineering in the small scale. Additionally we offer a course treating the subject of Software Quality, which enjoys an increasing interest even from outside the university.

To summarize, the area of Program Design give more than 30 courses a year, and these are mainly related to the fields of programming languages and paradigms, data structures, and program development. To give these courses we require a teaching staff corresponding to 14 full time annual positions and 2000 hours of training supervised by assisting students. While taking our courses the students also consume approximately 55.000 hours of terminal (work station) resources a year.

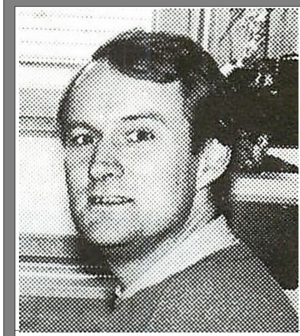
The selection of courses is always changing and developing, and almost every year some courses are excluded from the various programs while new ones are built up and introduced. This year e.g. Ada will replace Pascal as the first imperative language given in the C-programme, and efforts have been made to create a new course with this contents. The change that will have the greatest impact is however the planned idea of giving the students in the Y-programme LISP as their first programming language instead of Pascal. For this purpose an "experimental" course was given last year. This experiment will be of great help when we now are about to start the demanding development of the new course.

In 1991/92 the first year of the 2 years Computer Engineering Programme was given for the first time. All four Computer Science courses in this programme lie within the Program Design area, and since three of them did not exist before, we spent much of our efforts to build them up. Now these courses, Programming in C, Datastructures and Algorithms (in C), and Program Development Project (in C), have been successfully implemented, and will be given again in the years to come.

Of course some minor modifications to existing courses have also been made. The most remarkable one is perhaps that Pascal is no longer taught in full extent in the C and D programmes, but is merely integrated as one introductory detail among many others in the very first IDA-course (Introduction to Computer Science and Computer Equipment).

Cognitive Science

19
Subject area: Cognitive Science



Arne Jönsson
director

From back: Nils Dahlbäck, Richard Hirsch, Patrick Doherty, Jonas Löwgren, Arne Jönsson, Erik Sandewall, Åke Thurée, Sture Hägglund, Jalal Maleki, Dimiter Driankov, Ivan Rankin, Stefan Svenberg Hans Holmgren, Lisbeth Linge.
(All teachers engaged in the undergraduate programme in Appendix C).

Figure 19-1. *Sun computer lab.*

19.1 Overview

The subject area Cognitive Science includes courses in the various fields of cognitive science such as artificial intelligence, linguistics and cognitive psychology. The courses are tailored according to the objectives and backgrounds of the participants. Courses are provided for univerisity students and for companies taking part in our continuing eduaction programme. We cater for students from technical and philosophical faculties at introductory and advanced levels.

Cognitive Science plays an important role at the department both in education and research as well as in co-operation with industry. The teachers come mainly from the natural language processing laboratory, the application systems laboratory and the laboratory for representation of knowledge in logic.

Currently some twenty courses are given in the following areas:

- **Linguistics/Computational Linguistics**. Courses include linguistics, computational linguistics and computational models for natural language processing.

  A new lecturer was employed with responsibilities for the linguistics/computational linguistics courses.
- **Cognitive psychology**. This sub-area includes an introductory course on cognitive psychology and an advanced course on cognitive models for communication.
- **Human-computer interaction**. This is another important application area were we provide courses meeting different objectives.

  The courses in human-computer interaction given to students with a background in computer technology or in systems development, have been more closely connected. The course given to the student with a background in systems development has been extended and now includes more laboratory work.
- **Logic**. Both an introductory course to logic and more AI-oriented knowledge representation courses are given.

  The course in fundamental logic have been developed and new lectures were added to provide more training for the students.
- **Artificial intelligence.** There are three introductory AI-courses, differing in prerequisites and possible continuations. Continuations include in-depth courses in AI-programming and knowledge representation.
- **Expert systems.** Knowledge acquistion and expert system design are covered in a variety of courses
- **Databases**. Five different courses ranging from technically detailed database courses to courses on data modelling are given.

  A new course in technical databases will be given the acedemic year 92/93 which required new development. This development was carried out by Jalal Maleki and Thomas Padron-McCarthy together with professor Tore Risch at the new research laboratoy in technical databases.

Figure 19-2. *Student office of the department.*

System Architecture

20 Subject area: System Architecture

Johan Fagerström
director

Back row: Lars Strömberg, Johan Fagerström, Anders Törne, Björn Fjellborg, Staffan Bonnier.
Front row: Gunilla Lingenhult, Jan Maluszynski, Krzysztof Kuchcinski.
(All teachers engaged in the undergraduate programme in Appendix C).

TEMPUS Project

Krzysztof Kuchcinski has got financing for participation in TEMPUS project JEP-2754. The financing has been granted by Linköping University (50.000 SEK) and VHS (148.000 SEK). The project is coordinated by the **University of Amsterdam**, the Faculty of Mathematics and Computer Science and personally by Prof. Dr. ir. F.C.A. Groen. The aim of the project is to restructure the curricula for Computer Science, Electronics Engineering, and Control Engineering on the **Technical University of Timisoara**, Rumania.

Other partners involved in the project are **Fluke and Philips**, Eindhoven, The Netherlands, and **Universität Freiburg**, Freiburg, Germany.

20.1 Overview

Courses given by the subject area System Architecture are mainly concentrated around computer systems, compiling and computer architecture. We give courses both to students from technical and philosophical faculities and to Swedish industry.

The total number of students is approximiately 1100, mostly from late-year civil engineering programs. Our teachers mainly come from the Programming Environments laboratory, the Computer Aided Design laboratory and the Computer Assistance in Engineering laboratory. Most of our courses include a large part of pratical work, in particular computer network and operating system courses.

Currently we give more than twenty courses dividided into the following areas:

Parallel programming and operating systems

We offer four courses of varying complexity for students wanting to learn how to build and understand complex parallel systems, in particular operating systems. One course is specialized towards real-time systems, another towards technical support of operating systems.

Compilers and interpretors

This area includes two traditional courses on how to construct a compiler and three courses on related theory: formal languages and automata theory, formal specification, and term rewriting systems.

Computer networks and distributed systems

Computer networks play an increasingly important role in computer systems, our network course is therefore very popular (is has well over 100 students even though it is not mandatory). As an add-on we offer a course on distributed systems, in particular distributed operating systems.

Computer architecture

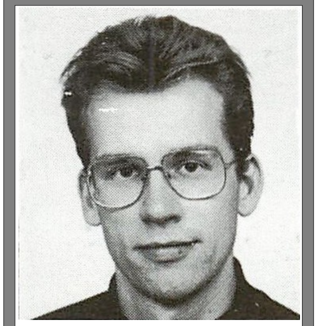
We give two advanced computer architecture courses, one based on modern (parallel) architectures and one project-oriented course, where the students design a complete system (including software) based on Inmos Transputers. We also give a course on computer aided design of electronics.

In addition to these courses, we also give courses on discret simulation techniques, computers in measurements and C-programming.

Figure 20-1. *Students' working area.*

Economic
Information Systems

21
Subject area: Economic Information Systems



Stefan Blom
director

Back row: Carl-Gunnar Hansson, Lars Mattsson, Torbjörn Näslund, Eva Elfinger, Birger Rapp, (Maria Olsson, Camilla Sjöström), Mehran Noghabai, (Anette Stolt).
Front row: (Peter Carlsson), Jonas Lind, Björn Helander, Jörgen Andersson, Anna Moberg, Jaime Villegas.
(Persons named within parenteses are not members of the teaching staff. All teachers engaged in the undergraduate programme in Appendix C).

Figure 21-1. *A view of the university campus.*

21.1 Overview

Economic Information Systems provides courses for different programmes at the Institute of Technology as well as at the undergraduate programmes Business Administration and System Analysis. Specially designed courses in Management control are also produced.

Most courses are produced for the MSc programme Industrial Engineering and Management. All students undergoing the programme attend a course in *Computers – Tools and Applications* within their first year. This introductory course shows the use of computers in commerce, industry and administration and explains the principles which govern the operation of modern computers. It also discusses the importance of information to enterprises of all kinds and describes the issues of linking dispersed users of data processing with networks.

Within the second year the students attend a compulsory course in *accounting and budgeting*. This course discusses different aspects of management control, i.e. product costing, budgeting, cost accounting and different techniques of analysing and evaluating business activities. A similar course is also compulsory for the programme of Mechanical Engineering

In 1991/92 Economic Information Systems was introduced as one of five profiles of specialization within the programme of *Industrial Engineering and Management*.

The profile consists of three courses which studies the role and usage of computer-based information systems for supporting decision making at different organizational levels. It also discusses planning and control of different business functions both theoretically and practically. The student learns to participate in the system development process for different kinds of information system applications. Important parts are information analysis and system design for in-house development as well as evaluation and implementation of purchased ready-made software. Methodologies for strategic information systems planning for establishing policies and priorities for development of individual applications are assessed and practical procedures for control, review, and maintenance of information systems projects are applied. Parts of these courses are also available for the undergraduate programme of Business Administration and Systems Analysis.

The profile can also be combined with a course in office information systems which discusses the organizational communication and the role of advanced information technology in new organizational networks.

During the last year the development and change of courses produced by the subject area have been increased, with new contents, workbooks and laboratory sessions in almost all courses.

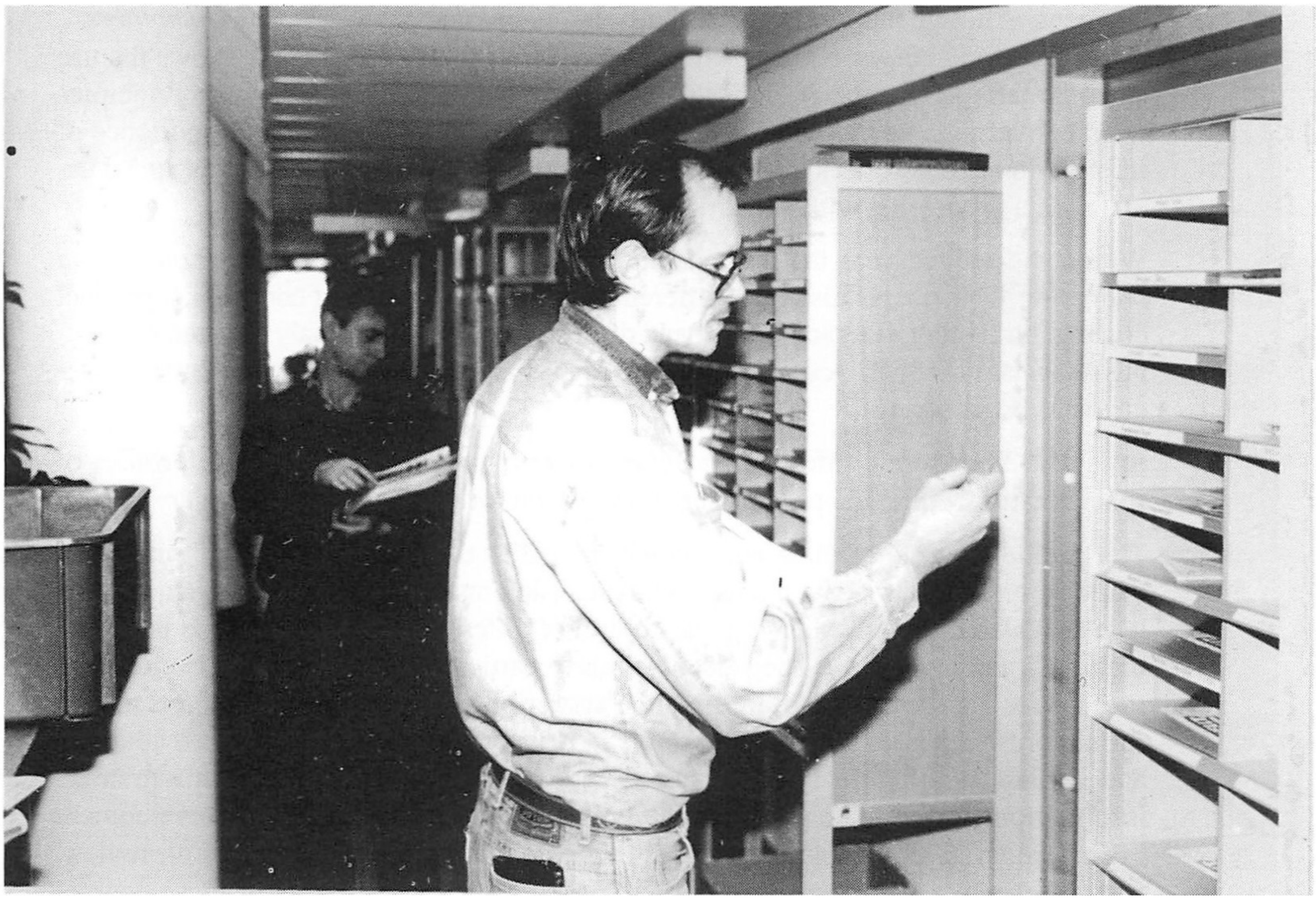


Figure 21-2. *Nils Nilsson, porter of the department, delivering mail*

Appendix A
Department Organization

The Department of Computer and Information Science (IDA) at Linköping University covers now four teaching subjects (computer science, economic information systems, telecommunication and computer systems, and administrative data processing).

The Department was formed in 1983, bringing together groups previously in the Mathematics and the Electrical Engineering departments. In September 1990 another group, Economic Information Systems, came from the Department of Production Economics.

A considerable flexibility was allowed when the internal organization and routines were to be decided. The basic idea was to build research within the department upon vital, autonomous, and cooperating research groups, each with a distinct leader and about five to ten more teachers, researchers, and employed graduate students. From the beginning there were four such groups or laboratories. Today there are thirteen.

Formally all significant administrative decisions, such as the annual budget are taken by the Department Board. Important and general issues regarding research or undergraduate studies are treated by the Research Committee or the Undergraduate Teaching Committee respectively. Running economy and personnel issues are handled by the administrative manager, who is also the leader for the group providing administrative services.

The lab leader is responsible for supervision and guidance of the work in his/her group, and also for writing grant proposals and reports to funders. Each lab also takes responsibility for maintaining competence in its area of research and some related areas, and to make it available to the rest of IDA in graduate courses and seminars, as well as in the undergraduate course program. The set of labs is designed to provide a sufficiently wide basis for a vital computer science department and also to give the necessary spectrum required for the undergraduate courses given by the department. At the same time it is important that research is sufficiently focused and that a group can achieve critical size in its area of specialization.

The department budget for the fiscal year 1992/93 balances at 65 MSEK. (One SEK is at present approximately 0.15 USD.) The resources for undergraduate education supplied by the university amount to 22,7 MSEK, and corresponding resources for research and graduate education are 15,2 MSEK. The research activities are thus heavily dependent on external sources, where the *Swedish National Board for Industrial Technical Development* (NUTEK) is the main contributor (92/93 9,4 MSEK). Additional funds are provided by the *Swedish Research Council for Engineering Sciences*, (92/93 4,6 MSEK), the *Swedish Council for Planning and Coordination of Research* (FRN), (92/93 1,5 MSEK) and the *Swedish Council for Research in the*

Humanities and Social Sciences, HSFR (92/93 1,3 MSEK). Through European co-operation with ESPRI, AIM, COMETT, etc., we have budgeted 1,4 MSEK for 1992/93, but we expect that the outcome of approved proposals will amount to something in the order of MSEK 5. We also have external funds through co-operation with external accounting firms (1992/93 2,6 MSEK). Totally our external funds amount to 27,2 MSEK.

Figure A-1. *(Departmental "Coffee Break") Every Tuesday the whole department meets for 45 minutes to receive and exchange information about current developments.*

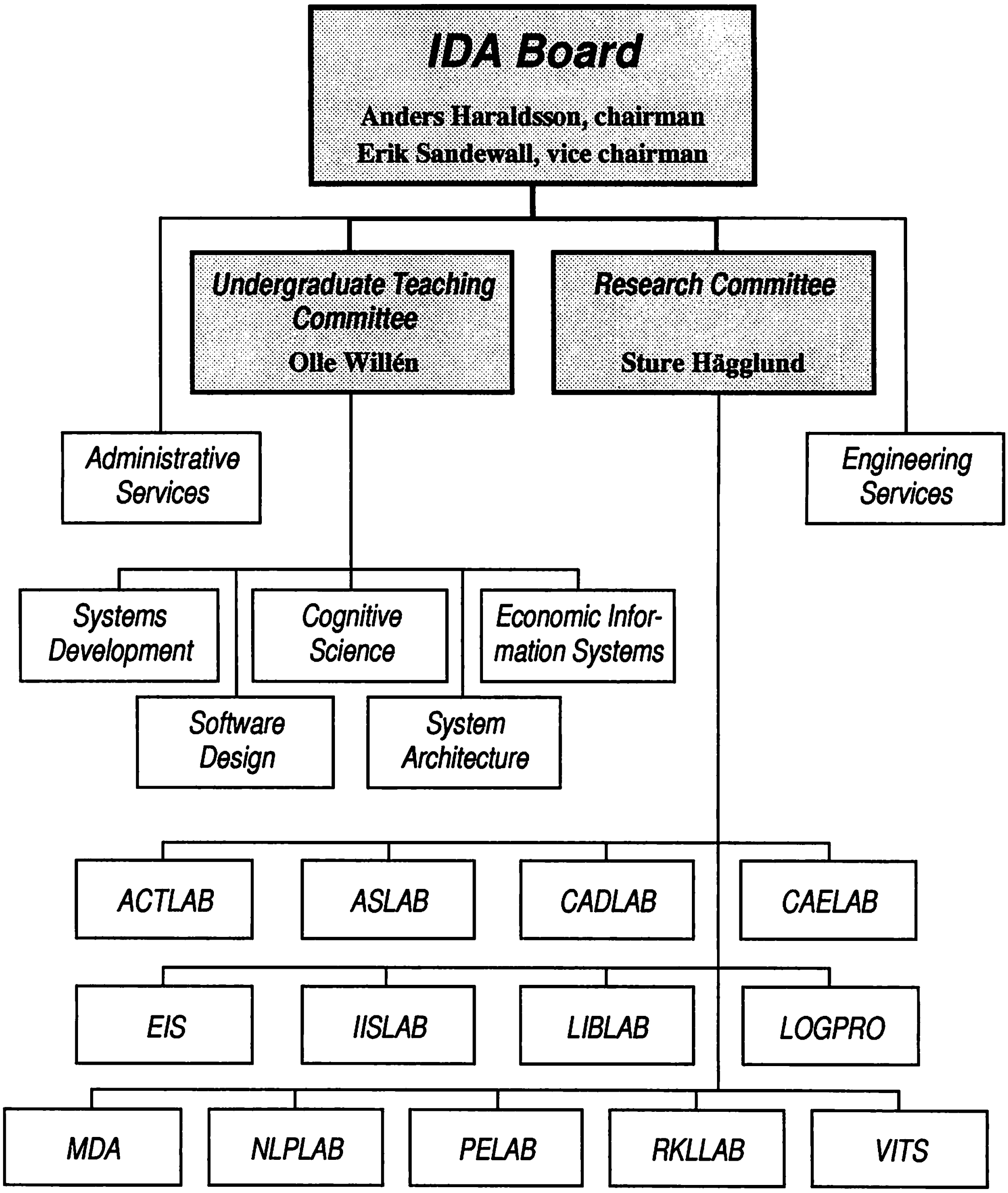


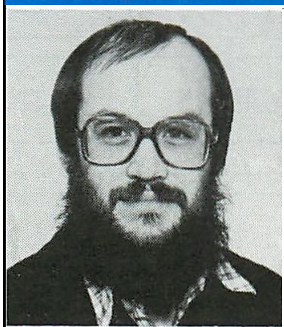
Figure A-2. *Organization of the department 1992/93.*

IDA Board

Anders Haraldsson
chairman

Erik Sandewall
vice chairman

Undergraduate Teaching Committee



Olle Willén

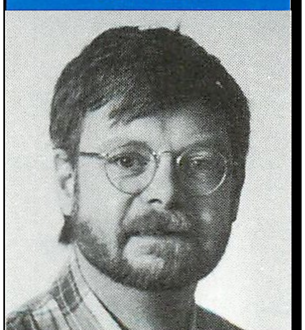
Research Committee

Sture Hägglund

Administrative Services

Inger Emanuelson

Technical Services



Anders Aleryd

A.1 The Department Board

The Department Board is chaired by Anders Haraldsson, with Birgitta Franzén as secretary and the items are mostly prepared and submitted by Inger Emanuelson. Annually the board delegates to two committees all issues about research and graduate studies, and about undergraduate education, respectively. The board also handles items related to both committees, normally by approving their coordinated proposals.

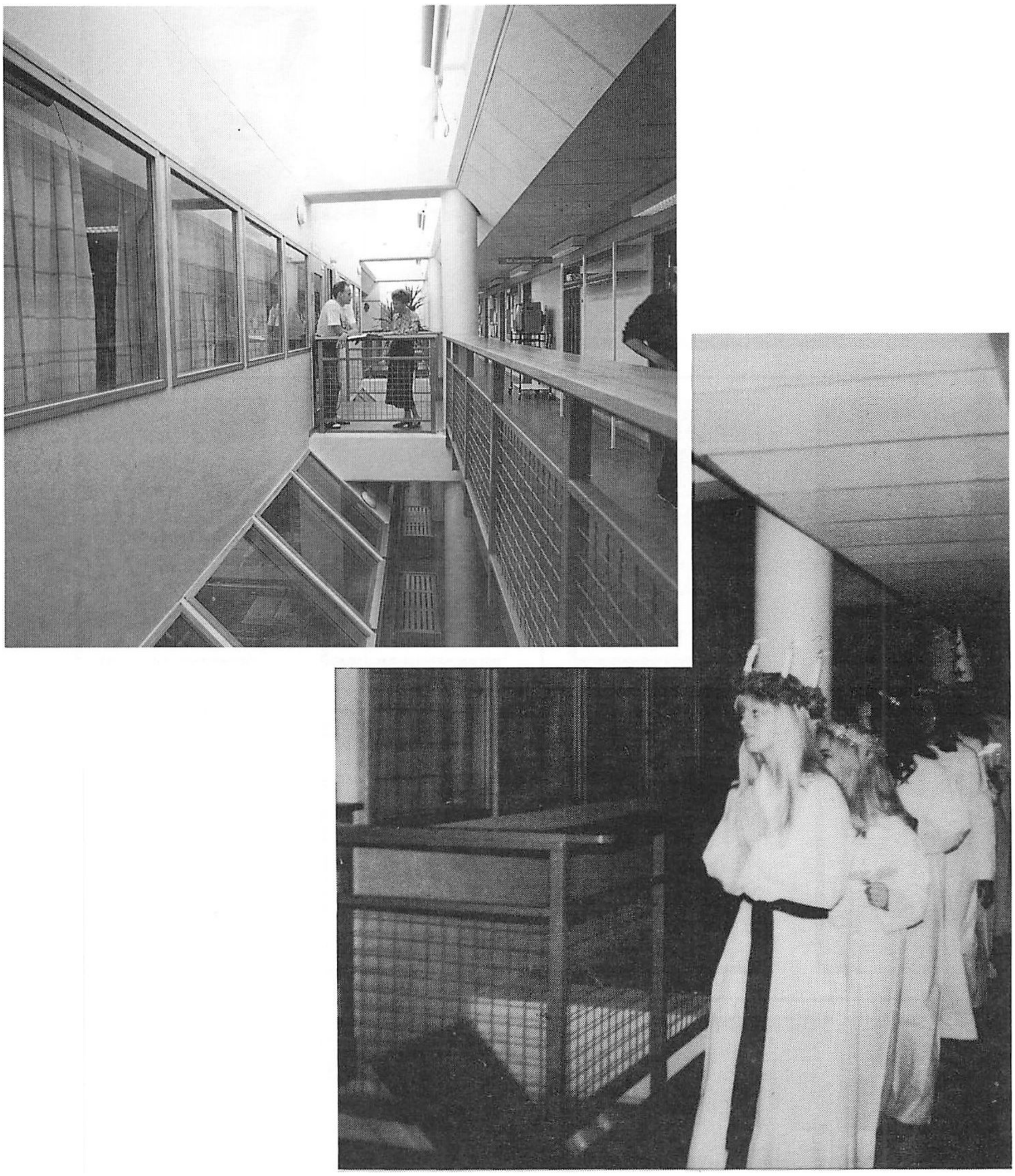
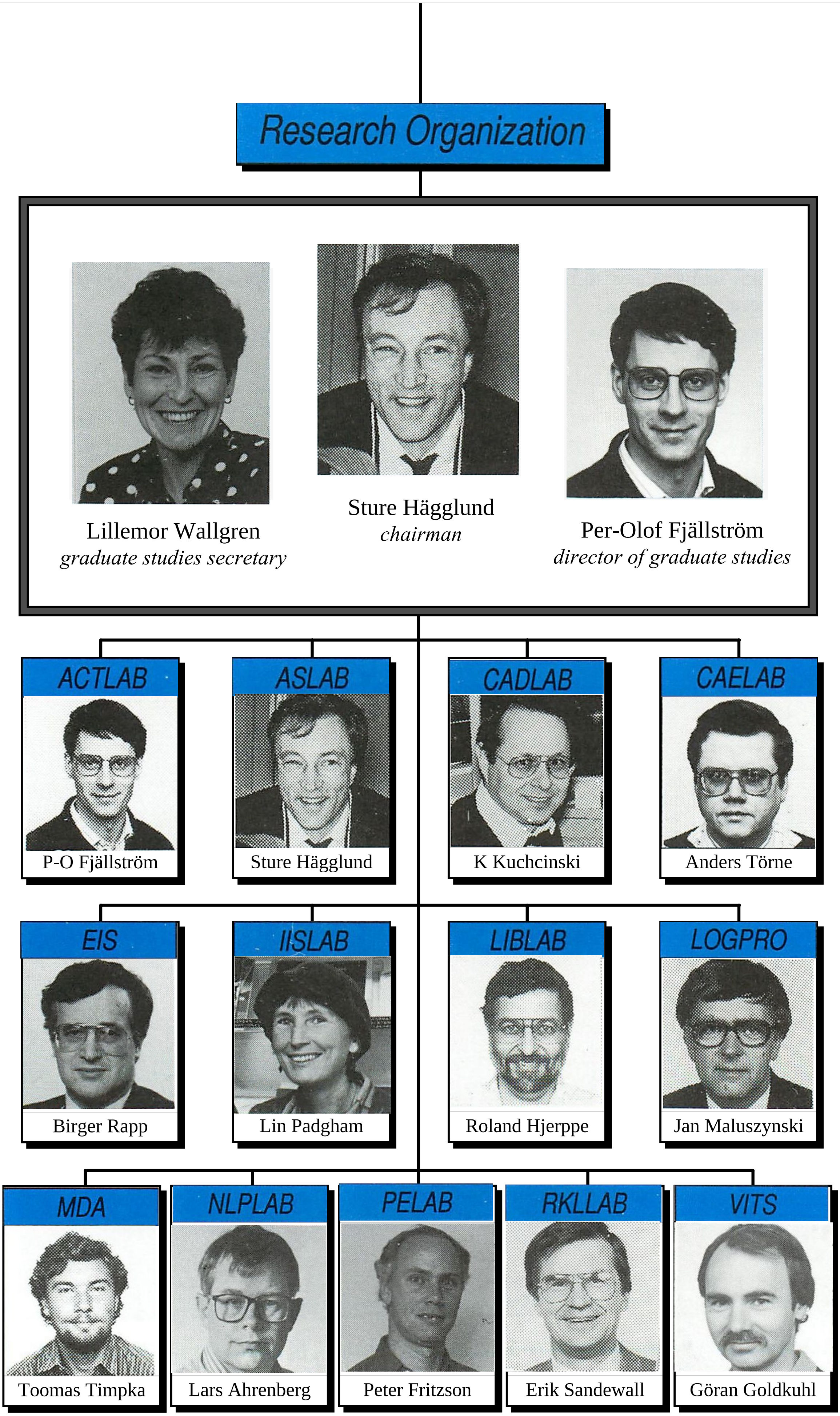


Figure A-3. *Above, interior from building E. Below, Lucia Day celebrations, 1992.*

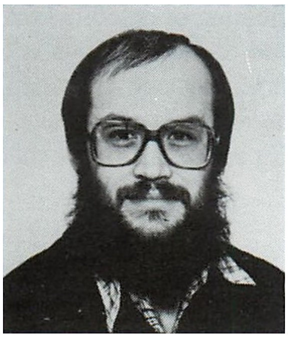

Research Organization
Lillemor Wallgren
graduate studies secretary
Sture Hägglund
chairman
Per-Olof Fjällström
director of graduate studies
ACTLAB
P-O Fjällström
ASLAB
Sture Hägglund
CADLAB
K Kuchcinski
CAELAB
Anders Törne
EIS
Birger Rapp
IISLAB
Lin Padgham
LIBLAB
Roland Hjerppe
LOGPRO
Jan Maluszynski
MDA
Toomas Timpka
NLPLAB
Lars Ahrenberg
PELAB
Peter Fritzson
RKLLAB
Erik Sandewall
VITS
Göran Goldkuhl

A.2 The Research Organization

The Research Committee, headed by Sture Hägglund and with Lillemor Wallgren as secretary, handles research activities and graduate education. The committee suggests the annual budget for each research lab, based on the grant situation, and can also modify the lab structure by merging, splitting, creating or deleting labs and appointing lab leaders. Admission of doctorate students and master students is co-ordinated 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.

Figure A-4. *Meeting of the Research Committee.*

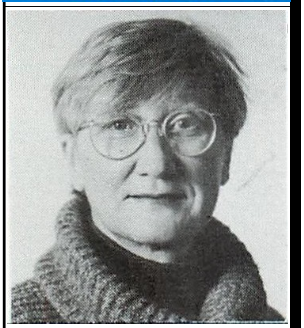
Undergraduate Teaching Organization



Olle Willén
chairman

Britt-Marie Ahlenbäck
undergraduate studies secretary

System Development

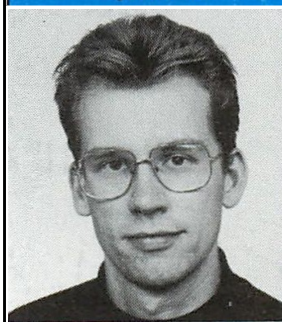


Lise-Lotte Raunio

Cognitive Science

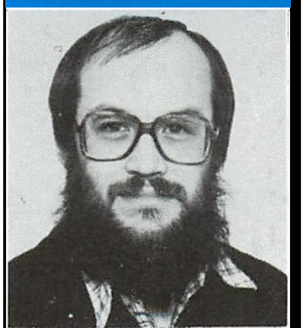
Arne Jönsson

Economic Information Systems



Stefan Blom

Program Design



Olle Willén

System Architecture

Johan Fagerström

A.3 The Undergraduate Teaching Organization

The Undergraduate Teaching Committee, headed by Olle Willén and with Britt-Marie Ahlenbäck as secretary, is responsible for the organization of undergraduate courses and continuing education for industry. Most of the teachers and lecturers are also members of the research labs and the decision about teaching load for each individual, in terms of percentage, is taken annually in conjunction with the budget negotiation process.

A.4.1 Subject areas

Under the Undergraduate Teaching Committee there are five subject areas, each with an executive director of studies, and there is one secretary, Britt-Marie Ahlenbäck, for undergraduate education.

The executive responsibility for undergraduate studies is taken by the directors of studies. We have five directors, Stefan Blom (Economic Information Systems), Johan Fagerström (Computer Architecture), Arne Jönsson (Cognitive Science), Lise-Lotte Raunio (Systems Development) and Olle Willén (Software Design). They are all responsible for the study programs within their area of knowledge.

Figure A-5. *Course meeting. Tommy Hoffner and Tommy Olsson.*

A.5 Administrative services

The administrative group under Inger Emanuelson is responsible for administrative services including economic and staff services. The secretaries are members in the administrative group as well as members in one or more educational or research group. In that way we are more prepared to step in for each other but everyone still has her/his own tasks.

Administrative office personnel

Inger Emanuelson	administrative manager.
Britt-Marie Ahlenbäck	general educational secretary.
Ingalill Andersson	substitute secretary.
Eva-Britt Berglund	secretary of the subject area System Development.
Barbara Ekman	secretary of the subject area Software Design, and the Laboratory for Library and Information Science.
Eva Elfinger	secretary of the subject area Economic Information Systems, and the group for Economic Information Systems.
Anne Eskilsson	secretary (on leave, returning in 1993).
Birgitta Franzén	secretary of the Laboratory for Applications Systems.
Madeleine Häger	office assistant (on leave).
Anne-Marie Jacobson	secretary of the Technical Support Group.
Eva Johansson	office assistant and secretary of the group for Economic Information Systems.
Carita Lilja	staff assistent, and the department registrar.
Lisbeth Linge	secretary of the subject area Cognitive Sciences, and the Laboratory for Natural Language Processing.
Gunilla Lingenhult	secretary of the subject area Computer Architecture, and the laboratories of Computer Assistance in Engineering, Logic Programming, Programming Environments, and the project for Industrial Software Technology.
Bodil Mattsson Kihlström	secretary of laboratories for Complexity of Algorithms, Computer Aided Design, and Intelligent Information Systems.
Nils Nilsson	department porter.
Lise-Lott Svensson	secretary of the laboratory for Representation of Knowledge in Logic.
Siv Söderlund	secretary of the group for People, Computers and Work.
Lillemor Wallgren	general research secretary.
Lena Wigh	secretary (on leave).

Administrative Services

Administrative Services

Inger Emanuelson
administrative manager

B-M Ahlenbäck

IngaLill Andersson

Eva-Britt Berglund

Barbara Ekman

Eva Elfinger

Anne Eskilsson

Birgitta Franzén

Madeleine Häger

AM Jacobson

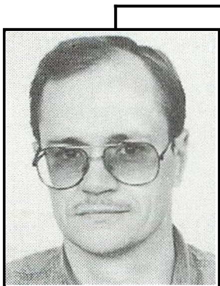
Eva Johansson

Carita Lilja

Lisbeth Linge

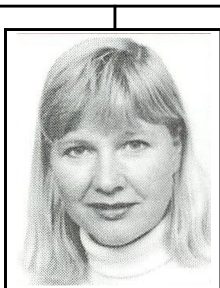
Gunilla Lingenhult

Bodil Mattsson



Nils Nilsson

Lise-Lott Svensson



Siv Söderlund

Lillemor Wallgren

Lena Wigh

A.6 Technical services

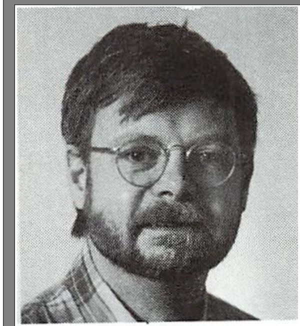
The system support group under Anders Aleryd is responsible for computer systems and services, as well as for all kinds of technical equipment at the department. Computer resources and other equipment are normally not reserved for a specific group or project, but shared as far as possible and supported at the department level. This allows a good economy for support costs and effective use of the facilities, although projects needing exclusive access to a particular piece of equipment of course can be granted that right for a specific period of time.

Technical services personnel

Anders Aleryd	Managing engineer, member of the IDA board.
Anne-Marie Jacobson	Secretary.
Mats S Andersson	Senior research engineer, system manager research facilities.
Leif Finmo	Research engineer, system manager for Sun systems and hardware facilities.
Arne Fäldt	Senior research engineer, system manager for Sun systems and Editing and Publishing program environment.
Lars Mattsson	System manager for PC’s and AS/400 systems.
Bernt Nilsson	Research engineer, system manager for Sun systems and LISP systems.
Björn Nilsson	Senior research engineer, system manager for Sun systems, administration environment.
Peter Nilsson	Research engineer, system manager for Sun systems, educational systems.
Rolf Nilsson	Research engineer, system manager for PC’s systems.
Göran Sedvall	Engineer, manager of computer networks and hardware.
Jaime Villegas	System manager for PC’s and IBM AS/400 systems.

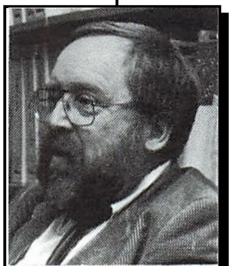
Technical Services

Technical Services

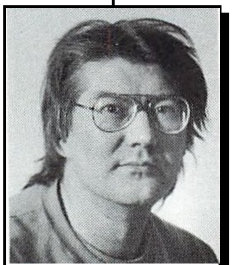


Anders Aleryd
managing engineer

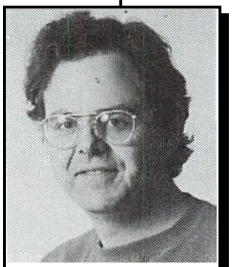
A-M Jacobsson



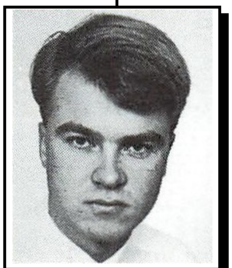
Mats S Andersson



Leif Finmo

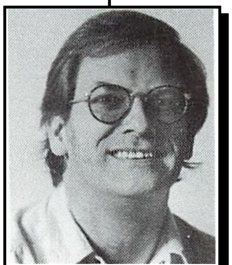


Arne Fäldt



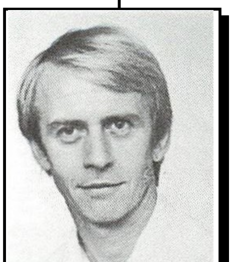
Lars Matsson

Bernt Nilsson



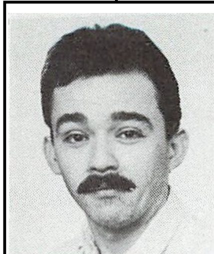
Björn Nilsson

Peter J Nilsson



Rolf Nilsson

Göran Sedvall

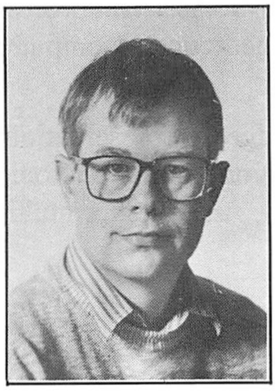


Jaime Villegas

Figure A-6. *Administrative and technical staff.*

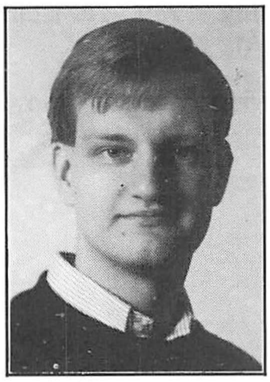
Appendix B
Graduate Study Program

B.1 Faculty presently engaged in graduate study program.



Lars Ahrenberg, Ph. D., Uppsala 1987. Associate professor (*docent, högskolelektor*) of computational linguistics. Group leader, NLPLAB. Previous affiliations Uppsala and Göteborg.

Syntax, semantics and pragmatics of natural language; natural language understanding; natural language interfaces; text generation.

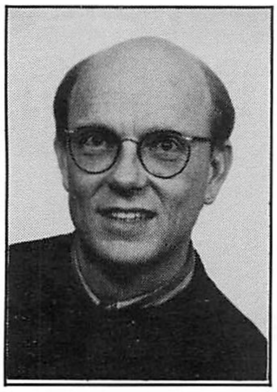


Christer Bäckström, Ph. D., Linköping 1992. Research associate (*forskarassistent*).

Planning and reasoning about plans, algorithms and complexity for AI problems, representation and reasoning about knowledge.

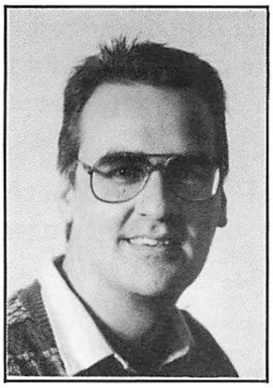
Douglas Busch, Ph. D., Rockefeller 1973. Associate professor (*högskolelektor*) of logic and theoretical computer science. Previous affiliation Mcquarie Univ., Sydney, Australia. On partial leave.

Application of theories from formal logic to problems in theoretical computer science and artificial intelligence; semantics of logic programs; philosophical questions in artificial intelligence.



Nils Dahlbäck, Ph. D., Linköping 1992. Assistant professor (*högskolelektor*), cognitive Science.

Natural language processing, especially empirically based computational models of discourse. Cognitive aspects of discourse coherence in man and machine. Philosophy of mind and its consequences for empirical theories in cognitive science.



Patrick Doherty, Ph. D., Linköping 1991. Assistant professor (*högskolelekto*r), logic and theoretical computer science.

Logical approaches to knowledge representation; reasoning with incomplete information, non-monotonic reasoning; reasoning with uncertainty, fuzzy logic; multi-valued and partial logics.

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

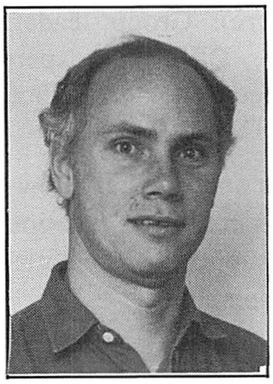
Reasoning under uncertainty, many-valued logics, knowledge-based plan-recognition, decision support systems.

Johan Fagerström, Ph. D., Linköping 1988. Assistant professor (*högskolelektor*), computer science. Director of undergraduate studies in System Architecture.

Distributed systems, parallel systems, operating systems.

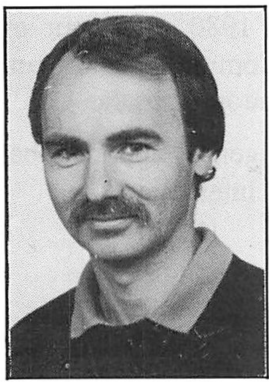
Per-Olof Fjällström, Ph. D., Stockholm 1985. Associate professor *(högskolelektor)*, theoretical computer science. Previous affiliation KTH and IBM. Group leader, ACTLAB.

Computational geometry, analysis of algorithms, data structures.



Peter Fritzson, Ph. D., Linköping 1984. Associate professor (*docent, högskolelektor*), computer science. Group leader, PELAB. Previous affiliations Sun MicroSystems USA 1985-86, Linköping Univ. Hospital 1976-79.

Programming environments, scientific computing, debugging tools, incremental compilation technology, compiler generation, compilers for parallel hardware.

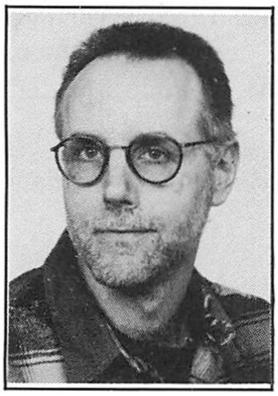


Göran Goldkuhl, Ph. D., Stockholm 1980. Associate professor (*docent, högskolelektor*), administrative data processing. Group leader VITS. Previous affiliations Stockholm, Göteborg.

Theories/methods on problem formulation, activity analysis, IS design and evaluation; ISD methods and customization of CASE tools; Humanistic science traditions and qualitative research methods.

Anders Haraldsson, Ph. D., Linköping 1977. Associate professor (*högskolelektor*), computer science. Department chairman. Previous affiliation Uppsala.

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



Richard Hirsch, Ph. D., Gothenburg 1989. Lecturer (*docent, högskoleadjunkt*) of general and computational linguistics. Thesis supervision in the laboratory for natural language processing. Previous affiliation Gothenburg.

Syntax, semantics, and pragmatics of natural languages; discourse analysis; argumentation theory; philosophy of language.

Roland Hjerppe. Researcher (*forskare*). Group leader, LIBLAB. Previous affiliation KTH, DFI and expert mission Tanzania. Visiting Distinguished Scholar at Office of Research, OCLC Inc. in Columbus, Ohio, 1988-89.

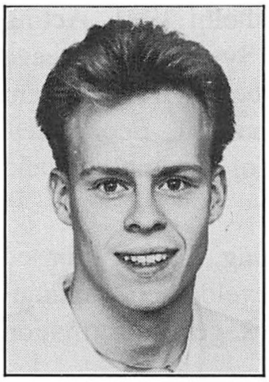
Library science and systems, hypertext and -media, knowledge organization and information retrieval, citation analysis and bibliometrics, computer support for personal and cooperative activities, virtual environments.

Sture Hägglund, Ph. D., Linköping 1980. Professor of knowledge-based systems. Research Committee chairman. Group leader, ASLAB. Previous affiliation Uppsala.

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

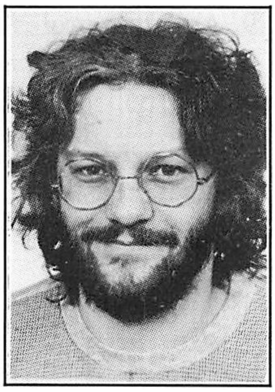
Krzysztof Kuchcinski, Ph. D., Gdansk 1984. Acting professor of computer systems. Group leader, CADLAB. Previous affiliation Technical Univ. of Gdansk, Poland.

Computer architecture, computer-aided design of digital systems, VLSI, test generation methods.



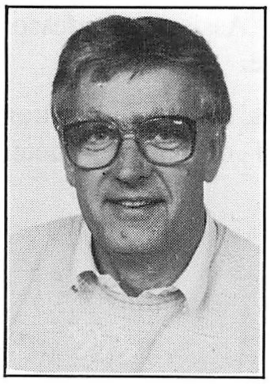
Jonas Löwgren, Ph. D., Linköping 1991. Assistant professor *(högskolelektor)*, human computer interaction.

Human Computer Interaction; User Interface Management Systems; Usability issues in information systems development; Expert Critiquing Systems.



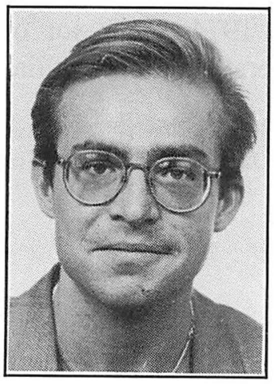
Jacek Malec, Ph. D., Wroclaw 1987. Research associate (*forskarassistent*). Previous affiliation Technical Univ. of Wroclaw, Poland.

Artificial Intelligence: knowledge representation, planning, reactive systems, autonomous systems architecture, dynamic scene description.



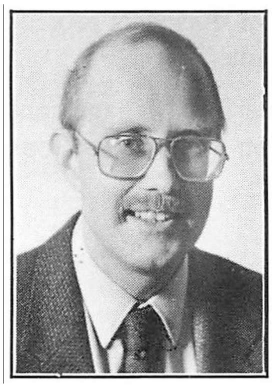
Jan Maluszynski, Ph. D., Warszawa 1973. Professor of programming theory. Several previous affiliations. Group leader, LOGPRO.

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



Ulf Nilsson, Ph. D., Linköping 1992. Assistant professor (*högskolelektor*), logic programming. On leave 1991–1992 to SUNY at Stony Brook, USA.

Logic programming and deductive databases; Evaluation strategies for query processing; program transformation and abstract interpretation.



Anders G. Nilsson, Econ. Dr., Stockholm 1991. Acting professor of Economic Information Systems. On leave from the Institute for Business Process Development (Institute V) and Stockholm School of Economics. Previous affiliations Royal Institute of Technology and Stockholm University.

Business modelling, strategy planning, activity based development, information systems development, maintenance management, application packages, information management.

Lin Padgham, Ph. D. , Linköping 1989. Assistant professor *(högskolelektor),* computer science. Previous affiliation Univ. of Oregon, USA, and Tektronix. Group leader, IISLAB.

Inheritance, default reasoning, taxonomical reasoning, object-oriented systems.

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

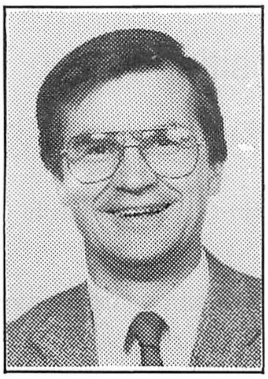
Automated synthesis of digital systems, formal description of hardware, VLSI, computer-aided design, computer architecture.

Birger Rapp, Econ. Dr., Stockholm 1974, Professor of Economic Information Systems. Several previous affiliations. On leave 1992-93.

Accounting systems, economic control, IT and organisation, production, economics.

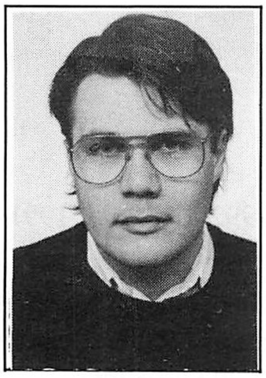
Tore Risch, Ph. D., Uppsala 1978. Professor of Engineering Databases. Previously at Uppsala University, IBM Almaden Research Lab. (San Jose, CA), Stanford Research Institute, Syntelligence Inc. (Sunnyvale, CA), HP Laboratories (Palo Alto, CA), and Stanford University.

Database support for engineering and scientific applications, e.g., object-oriented databases, heterogeneous data bases, active databases, and real-time databases.



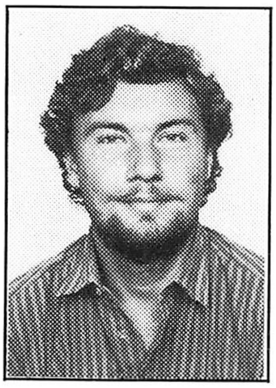
Erik Sandewall, Ph. D., Uppsala 1969. Professor of computer science. Vice-Chairman of the department. Research Committee Chairman (1983-1990). Group leader, RKLLAB. Several previous affiliations.

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



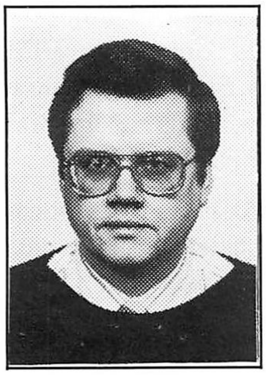
Kristian Sandahl, Ph. D., Linköping 1992, Research associate (*forskarassistent).*

Knowledge engineering, industrial software technology.



Toomas Timpka, MD., Stockholm 1983, Ph. D., Linköping 1989. Associate professor (*docent, högskolelektor*), computer and information science. Group leader, MDA.

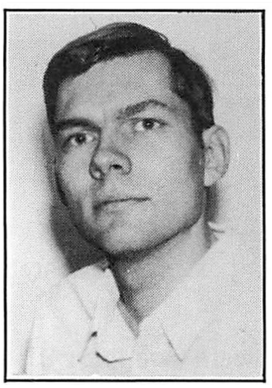
Hypermedia, computers and society, human-computer interaction, systems development.



Anders Törne Ph. D., Uppsala 1980. Associate professor (*högskolelektor)*, computer support in automation. Group leader, CAELAB. Previously at ABB Corporate Research, Västerås. Director of the graduate study programme. CENIIT manager from 1992.

Computer support for generation, transformation, and use of information in manufacturing processes. Architectures for processing control and supervision. Robot programming.

B.2 Guest researchers and Ph. D.'s in transition engaged in graduate study program

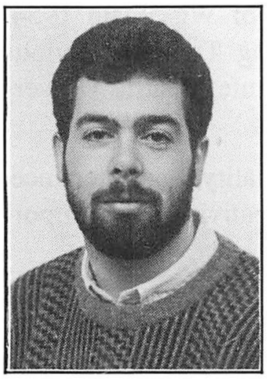


Roland Bol, Ph. D., Amsterdam 1991. Guest researcher. Previous affiliation CWI, Amsterdam.

Logic programming, non-monotonic reasoning, deductive databases, process algebra.

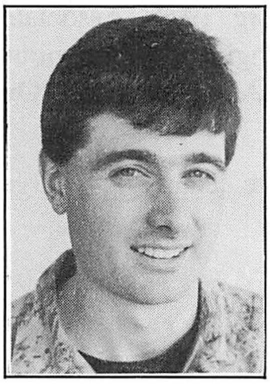
Staffan Bonnier, Ph. D., Linköping 1992. On leave 1991-92 to Ecole Normale Supérieure, Paris, France.

Logic and functional programming languages, extended unification and types for logic programs. More recently automatizing inductive proofs of program correctness.



Brant Cheikes, Ph. D., Univ. of Pennsylvania, 1991. Guest researcher.

Natural language processing and cooperative dialogue; architectures for response-planning systems, esp. intelligent help systems; simulator-based training systems. Applications of knowledge-based systems technology for operator training in the process industry.



Keith Downing, Ph. D., University of Oregon 1990. Guest researcher.

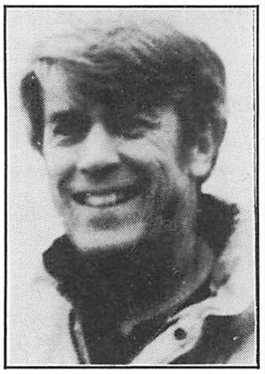
Model-based diagnosis, qualitative physics, reason maintenance, artificial intelligence in medicine. Applications of model-based simulation and diagnosis to physiological domains.

Wlodek Drabent, Ph. D., Warszawa 1985. Assistant professor (*högskolelektor*). On leave from Institute of Computer Science, Polish Academy of Sciences.

Logic programming, programming languages semantics.

Henrik Eriksson, Ph. D., Linköping 1991. On leave at Stanford University, Stanford, California 1991 – 93.

Knowledge-based systems, knowledge acquisition, software development environments, software reuse.

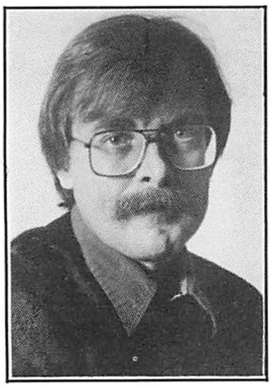


Robert L. Glass, M. Sc., University of Wisconsin 1954. Guest professor. President, Computing Trends. Previous affiliation Software Engineering Institute, Seattle University, The Boeing Company, USA.

Software engineering: software quality, maintenance, design, technical communication, creativity. The importance of the application domain.

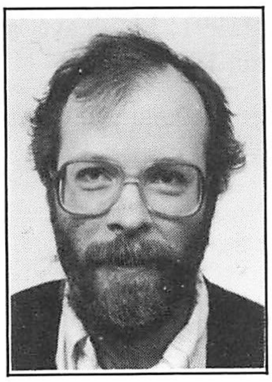
Bengt Lennartsson, Ph. D., Göteborg 1974. Associate professor (*högskolelektor*), software engineering. Previous affiliation Luleå. Group leader, PELAB until 1988. On leave 1992 .

Real-time systems, industrial software technology, large scale software development.



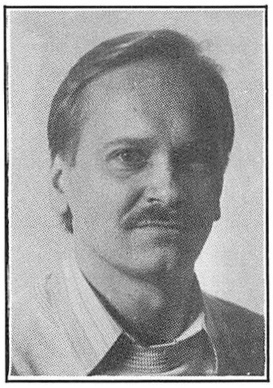
Witold Lukaszewicz, Ph. D., Warsaw University 1979. Guest professor. On leave from the Institute of Informatics, Warsaw University, Poland.

Knowledge representation, non-monotonic reasoning, programming methodology.



James M. Nyce, Ph. D., Brown 1987. Guest professor, computer and information science. Previous affiliation Brown.

Work and knowledge (medicine and academia); tradition, innovation and technology; hypertext and visual resource development paths.



Jukka Paakki, Ph. D., University of Helsinki, 1991. Associate professor of computer science (University of Jyväskylä, Finland). Guest researcher.

Programming paradigms, language design and implementation, attribute grammars, logic programming.

Ralph Rönnquist, Ph. D., Linköping 1992. Lecturer (*högskoleadjunkt*) computer science.

Formal representation techniques and advanced computation methods applied for information systems.

Nahid Shahmehri, Ph. D., Linköping 1991. Lecturer (*högskoleadjunkt*), computer science.

Programming theory, programming languages, debugging tools, compiling technology.

B.3 Graduate Study Course Program 1991–92

Basic and Occasional Graduate Courses:

Constraint Logic Programming (Jan Maluszynski)
Computer-Supported Cooperative Work (Toomas Timpka)
Compiling Functional Languages (Peter Fritzson)
Datasäkerhet (Birger Rapp)
Doktrinhistoria (Birger Rapp)
Human-Computer Interaction (Jonas Löwgren)
Information Retrieval I: Traditional and Newer Approaches (Roland Hjerppe)
Intelligent Autonomous Systems Architectures (Jacek Malec)
Introduction to Parallel Algorithms (Per-Olof Fjällström)
Introduction to Research Methodology in Computer Science (Sture Hägglund)
Kunskapsutveckling om informationssystem (Göran Goldkuhl)
Logic for Ida-ites (Patrick Doherty)
Metodik för Systemutveckling (Göran Goldkuhl)
Non-Monotonic Reasoning (Witold Lukaszewicz)
Operating Systems (Johan Fagerström)
Parallel Computers:Architecture & Programming (Kris Kuchcinski, Zebo Peng)
Parallel Execution Models (Johan Fagerström)
Process Algebra (Jan Maluszynski)
Real-Time Systems (Anders Törne)
Risker och Nyföretagande (Birger Rapp)
Representation of Knowledge about Dynamic Systems (Erik Sandewall)
Semiotics: History, Basic Concepts and Computer Applications (Richard Hirsch)
Software Engineering Design and Methodology (Lin Padgham)
Teorier och Strategier för Informationssystem (Göran Goldkuhl)
Verksamhets- och Informationsbehovsanalys (Göran Goldkuhl)

B.4 Graduate Study Course Program 1992–93 as planned

Basic and Occasional Graduate Courses:

Business Modelling (Anders G. Nilsson)
Cognitive Psychology (Nils Dahlbäck)
EIS classics (Anders G. Nilsson)
Data Structures and Graph Algorithms (Per-Olof Fjällström)
Fuzzy Logic and Control (Dimiter Driankov)
Generation of Incremental Environments (Peter Fritzson)
Information och verksamhet (Göran Goldkuhl)
Introduction to Research Methodology in Computer Science (Sture Hägglund)
Introduction to Systems Theory (Jacek Malek)
Natural Language Processing (Lars Ahrenberg)
Petri Nets and Formal Description of Parallel Systems (Zebo Peng, Kris Kuchcinski)
Principles of Modern Database Systems (Tore Risch)
Redovisningsteori (Jan Dahlgren, Birger Rapp)
Vetenskaplig Metodik (Birger Rapp)
Advanced Logic (Douglas Busch)
Argumentation Theory: Part I Philosophical Foundations (Richard Hirsch)
Automated Debugging (Peter Fritzson, Mariam Kamkar)
Computability and Complexity Theory (Per-Olof Fjällström)
Computers and Computer Science: A Historical Approach (James Nyce)

Inductive Learning Methods (Nada Lavrac), possibly
Introduction to Neural Nets (Kris Kuchcinski)
Intelligent Information Systems (Lin Padgham)
Knowledge Acquisition (Kristian Sandahl)
Kvalitativ undersökningsmetodik (Göran Goldkuhl)
Object-Oriented Programming (Jukka Paakki)
Structured Operational Semantics (Ulf Nilsson)
The Role of Maintenance in Software Engineering (Robert Glass)
Svensk och internationell externredovisning (Rolf Rundfeldt, Birger Rapp)

B.5 Seminars

Seminars Study Year 1991–1992

March	14	Prof. Harold W Lawson, Cy-Clone - *An Approach to the Engineering of Resource Adequate Cyclic Real-Time System.*
	26	Prof. Gerhard Fischer, Univ. of Colorado - *Reducing the Power of High-tech Scribes*
April	4	Prof Mark A. Musen, Stanford University - *Generation of Model-Based Knowledge - Acquisition Tools from Reusable Components*
May	14	F Fukumoto, Y Tanaka, ICOT, Japan - *A Parallel Parsing System Based on Restricted Dependency Grammar*
	15	Janis Barzdins, Univ. of Lettland - *Inductive Synthesis of Programs: Old and New Approaches*
August	19	Prof. Kevin Ryan, Limerick Univ., Dublin - *Software Reguirements Capture and*
	21	*ESPRIT Past, Present and Future* and
	23	*Re-Engineering Software*
Sept.	25	Dr. Ron Loui, Washington University - *From where Defeasible Reasoning is coming to where Defeasible Reasoning is going*
Oct	1	Dr. Bo Hedberg, Stockholms univ. - *Imaginary Organization - A Challenge to Business Administration*
	14	Prof. Mihai Draganescu, Academia Romana - *The Functional Potentiability of Information*
	22	Prof John-Jules Meyer, Free University, Amsterdam - *Epistemic Logic for Computer Science* and
	23	*On Defaults and Counterfactuals*
	25	Dr. Stanley B Zdonik, Brown Univ., Providence, RI - *Object-Oriented Data-base Systems: Fact or Fiction*
	31	Mikhail Augustson, Latvia Univ. - *FORMAN - A Program Formal Annotation Language*
Nov	12	Jan Olsson, IBM, Stockholm - *An Architecture for Diagnostic Reasoning Based on Causal Models*
	14	Dr. Peter Struss, Institut fuer Informatik, Muenchen - *Diagnosis Tutorial* and
	15	*Modeling for Diagnosis*
Dec	12	Hilde Ade, Katholieke Univ, Leuven, Belgien - *Using CLINT's declarative bias in Plotkin lgg-framework*
	12	Prof Mary-Lou Soffa, Univ of Pittsburg - *A Framework for Generalized Slicing*

1992

Febr	18	Dr. Jerome Lang, Universite Paul Sabatier - *Possiblistic Logic*
March	4	Dr. Lars Borin, Uppsala univ - *How to make a Computer Learn Morphology*
	12	Jerker Wilander, Softlab, Linköping - *Concurrency and Locking protocols*
	20	Prof. Saumya Debray, Univ of Arizona - *On the Complexity of Dataflow Analysis of Logic Programs, A Portable and Efficient Sequential Implementation of Janus*
April	4	Prof. Reiner W Hartenstein, Kaiserslautern Univ - *Xputers: New Machine Principles and Compilation Methods for High Performance*
	6	Dr Michael Georgeff, Australian Artificial Intelligence Institute, Carlton, Victoria - *The Design of Rational Computational Agents*
	10	Prof Bonnie Lynn Webber, Univ of Pennsylvania - *Instructing Animated Agents*

	12	Prof. James Nyce, Brown Univ, Providence, RI - *If you can't see it - Agenda and Practice in Visualization Efforts*
	13	Sten Henriksson, Lunds Univ, Tord-Jöran Hallberg, Linköping Univ - *Vi som var med. Historieskrivningen utifrån universitets- resp industrihåll*
	18	Dr Tarabanis, IBM T. J. Watson Research Center - *Sensor Planning and Modeling for Machine Vision Tasks*
	21	Dr. Dmitry Apraksin, The State Academy of Management - *Management Decision Making by means of Intellectual Text Processing*
	25	Jerker Wilander, Softlab, Linköping - *Schema evololution i O-O Data-bases*
May	19	Prof. James Nyce, Brown Univ, Providence, RI - *What should a computer science department teach about its own history and the history of computing?*
	20	Dr. Bryan Lyles, Xerox Palo Alto Research Center - *ATM for Local Area Networks*
	22	Docent Katuscia Palamidessi, Univ of Pisa - *Concurrent Constraint Logic Programming*
	25	Nigel Horspool, Univ of Victoria - *Static Analysis of PostScript Code*
	25	Dr. Sharma Chakravarthy, Univ of Florida - *Event Specification in Sentinel - An Object-Oriented Active DBMS*
	25	Dr. John Fox, Imperial Cancer Research Fund, London - *Qualitative Frameworks for Decision Support: Lessons from Medicine*
June	5	Dr. Mats Carlsson, SICS, Kista - *Logikprogrammering med bivillkor i Boolesk Algebra*
	16	Dr. Peter Eklund, Univ of Adelaide - *Fuzzy Control: Fact or Fallacy?*
August	4	Dr. Pierre Deransart, INRIA Rocquencourt - *A proof method of partial completeness and weak completeness for normal logic programs* and
	5	*Formal specification by Logic Programmin*g
	11	Prof Barry Silverman, George Washington Univ - *Smart vs Dumb Expert Systems*
	11	Prof Halina and Teodor Przymusinski - *Stationary Default Extensions*
	13	Maria Lee, CSIRO, Sydney - *Modelling with Context-Dependent Causality*
	26	Prof. Carl-Gustaf Jansson, Stockholm Univ - *Machine Learning*
Sept	8	Dr. Jerry Feldman, ICSI, Berkeley - *The role of connectionism in artificial intelligence (or vice versa)*
	10	Rudolf K Keller, CRIM, Montreal and McGill Univ. - *User Interface Engineering in Perspective*
	17	Dr. Bart Selman, AT & T Bell Lab., Murray Hill - *Randomized Local Search for Satisfiability Testing*
	17	Francis Lorenz, Lorenz Consulting, Belgien - *Reflections about Representation Methods*
	22	Dr. Görel Hedin, Lund Univ - *Efficient Incremental Semantic Analysis with Door Attribute Grammars*
	29	Frank Tip, CWI, Amsterdam - *The ASF+SDF meta-environment*
Oct	13	Jordan Zlatev, Stockholm Univ - *Connectioist Modelling of Semantics and the Nature of Polysemy*
	15	Prof. Berndt Brehmer, Uppsala - *Dynamiskt och fördelat beslutsfattande*
Nov	20	Dr. Stephen Murphy, NASA Center for Intelligent Robotic System for Space Exploration, Troy, N.Y. - *Development of an integrated multiarm robotic system* and
	23	*Software aspects i large scale robotics integration* and
	23	*Issues in force control of coordinated robots*
Dec	10	Prof. Yvonne Waern, Linköping University -*CSCW i ledningssystem: Några tvärvetenskapliga synpunkter på datorstöd för samarbete*

Appendix C
Courses and Teachers in the Undergraduate Education 1992/93

IDA is responsible for computer and information science courses in the School of Engineering as well as in the School of Arts and Sciences. There are about 130 such undergraduate courses with a total of almost 5000 students.

C.1 Undergraduate courses in the School of Engineering and in the School of Arts and Sciences

The study programs in the **School of Engineering** lead to a Master of Engineering or a Master of Science degree (for the C-program) and run over 4–4.5 years. The annual intakes to the study programmes are:

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

The 2 year Engineering Programs lead to a University Certificate and the annual intakes are:

Computer Engineering (DI), 30 students per year
Electrical Engineering (EI), 60 students per year
Mechanical Engineering (MI), 30 students per year

There are also single subject courses given as part-time and evening courses, and external courses given directly to companies and organizations.

In the **School of Arts and Sciences** the program for Systems Analysis (systemvetenskapliga linjen, SVL) ranges over 3 years. The annual intakes is 30 students.

In the list below "ADB" means single subject courses in administrative data processing, and "Frist" means single subject courses in general. "FL" means the study programme for Science of public administration (förvaltningslinjen), and "SL" the study programme for Statistics (statistikerlinjen).

The courses given by IDA are divided into five undergraduate study areas

System Development: Lise-Lotte Raunio, director of undergraduate studies

Course name in Swedish	Course name in English	Examiner
Systemutveckling, teori och tillämpn. (C4, D4)	System Development	Christian Krysander
CASE (C4, D4)	CASE for Dev and Maintenance	Peter Fritzson
Programutv.metodik o progr.projekt (D4)	Meth of Prog Dev and Project	Christian Krysander
Ledarskap (C4, D4)	Leadership	Christian Krysander
Administrativt utvecklingsarbete (FL2)	Organizational Development	Pia Arendell
Systemutvecklingsprojekt I (SVL1)	System Development, project I	Stefan Cronholm
Datorisering och arbete (SVL2)	Ergonomics and Man-Machine Syst	Hans Holmgren
Databaser och datamodellering (SVL2)	Databases and Data Models	Ivan Nilsson
Systemutredningsmetoder (SVL2)	Methods for System Development	Eva-Chris Svensson
Informationshanteringssystem (SVL2)	Fourth Generation Languages	Rolf Nilsson
Systemutvecklingsprojekt I (SVL1)	System Development, project I	Stefan Cronholm
Systemutvecklingsprojekt II (SVL2)	System Development, project II	Ivan Nilsson
Förändringsanalys (SVL2)	Change Analysis	Annie Röstlinger
Datorstödd systemutveckling (SVL3)	Comp. Aided System Development	Göran Goldkuhl
Systemvetenskaplig grundkurs (SVL1)	Introduction to System Science	Hans Holmgren
Utv. och förändring av informationssyst. (SVL1)	System Development	Annie Röstlinger
Samhällsvetenskaplig metodik (SVL3)	Methods in Social Science	Göran Goldkuhl
Teorier o strategier för info system (SVL3)	Theories and Strategies on Inf Syst	Göran Goldkuhl
Introd. till systemutveckling (ADB1)	Introduction to System Development	Pia Arendell
Datorisering av administration, helfart (ADB1)	Computers in Office	Pia Arendell
Datorisering av administration, halvfart (ADB1)	Computers in Office	Pia Arendell
Systemutvecklingsprojekt A, helfart (ADB1)	System Development Project A	Lise-Lotte Raunio
Systemutvecklingsprojekt A, halvfart (ADB1)	System Development Project A	Stefan Cronholm
Elektro media, tekn, användn o konsekv (Frist)	Electronic Media	Roland Hjerppe
Samhällets informationsresurser (Frist)	Information Resources in Society	Roland Hjerppe
Förändringsanalys o projektadm (ADB2)	Change Analysis and Proj. Managem.	Lise-Lotte Raunio
Experimentell systemutveckling (ADB2)	Experimental System Development	Ivan Nilsson
Informationssystem i hälso- o sjukvård (Frist)	Information System for Health Care	Toomas Timpka
4G-språk (SL3)	Fourth Generation Languages	Stefan Cronholm
Praktik (SVL2, SVL3)	Practical Work	Lise-Lotte Raunio
Examensarbete (SVL3)	Degree Project	Hans Holmgren/ Eva-Chris Svensson

Software Design: Olle Willén, director of undergraduate studies

Course name in Swedish	Course name in English	Examiner
Programvarukvalitet (C4, D5)	Software Quality	Christian Krysander
Orientering, datateknik och datorutrustn. (C1, D1)	Introd. CS and Computer Equipment	Katarina Sunnerud
Programmering i ADA(C4, I4, M4, Y4, SVL)	Programming in Ada	Olle Willén
Ada och programspråk (C3, C4, D3, D4)	Ada and Programming Languages	Tommy Olsson
Imperativa språk (C1)	Imperative Programming Languages	Nahid Shahmehri
Programmering Y, gk (Y2)	Introd. to CS and programming	Torbjörn Jonsson
Programmering I, gk (I2)	Introduction to Programming	Torbjörn Jonsson
Datastrukturer (Y4, M4, M5, I3)	Data Structures	Tommy Olsson
Konstruktion och analys av algoritmer (C4, D5)	Design and Analysis of Algorithms	Per-Olof Fjällström
Objektorienterad programmering (C3, C4, D4)	Object-oriented Programming	Johan Fagerström
Logikprogrammering (C3, D4)	Logic Programming	Ulf Nilsson
Programmering i Pascal (D2)	Programming in Pascal, D	Rober Bilos
Projektarbete i programmering, p2 (D2)	Programming Development Project	Torbjörn Jonsson
Projektarbete i programmering, p4 (D2)	Programming Development Project	Mariam Kamkar
Projektarbete i programmering (C2)	Programming Development Project	Mariam Kamkar

Datastrukturer och algoritmer (C2, D3)	Data Structures and Algorithms	Jan Petersson
Programutvecklingsmetodik M (M3)	Methodology of Program Devel.	Olle Willén
Data- och programstrukturer (C2, D4)	Data and Program Structures	Eric Stoy
Datorsystem och programmering (M1)	Computer Systems and Programming	Rober Bilos
Programmering i inkrementellt system (C1)	Programming in Incremental System	Anders Haraldsson
Programmering i inkrementellt system (D1))	Programming in Incremental System	Anders Haraldsson
Programmering i C (DI1)	Programming in C	Thomas P.-McCarthy
Datastrukturer och algoritmer (DI1)	Data Structures and Algorithms	Tommy Olsson
Programmeringsprojekt (DI1)	Program Development Project	Lars Strömberg
Ada och programspråk (DI2)	Ada and Programming Languages	Olle Willén
Programmering i ADA (SVL3)	Programming in ADA	Olle Willén
Inl progr och datatekn, Pascal (SVL1)	Programming and CS, basic course	Stefan Cronholm
Progr för adm tillämpn, Cobol (SVL1)	Programming for Adm System	Stefan Cronholm
Datastrukturer o algoritmer (SVL2)	Data Structures and Algoritms	Andreas Björklind
Inl progr och datatekn, Pascal, helfart (ADB1)	Programming and CS, basic course	Thomas P.-McCarthy
Inl progr och datatekn, Pascal, halvfart (ADB1)	Programming and CS, basic course	Katarina Sunnerud
Progr för adm tillämpn, Cobol, helfart (ADB1)	Programming for Adm System	Stefan Cronholm
Progr för adm tillämpn, Cobol, halvfart (ADB1)	Programming for Adm System	Stefan Cronholm
Grundkurs i programmering (SL1)	Elementary Programming	Mariam Kamkar
Programutvecklingsmetodik (SVL3)	Methodology of Progr Development	Olle Willén

Cognitive Sciences: Arne Jönsson, director of undergraduate studies

Course name in Swedish	Course name in English	Examiner
Databaser (I4, I3, M4, Y4)	Databases	Åke Thurée
Människa-datorinteraktion ((C3,D4)	Human-Computer Interaction	Jonas Löwgren
Artificiell intelligens D (D3)	Artificial Intelligence	Arne Jönsson
AI-programmering (C3, C4)	AI Programming	Jalal Maleki
Logik, grundkurs (C1, D2)	Logic, introductory course	Erik Sandewall
Logik för AI (C4)	Logic for AI	Douglas Busch
Kognitiv psykologi (C1, D4, Y4)	Cognitive Psychology	Nils Dahlbäck
Databehandling av naturligt språk (C4)	Natural-Language Processing	Stefan Svenberg
Artificiell intelligens och LISP (I4, M4, Y4)	Artificial Intelligence and LISP	Åke Thurée
Databasteknik (C3, D4)	Database Technology	Jalal Maleki
Artificiell intelligens (C2)	Artificial Intelligence	Jalal Maleki
Exp.system–metodik o verkt. (C4,D4,I4,M4,Y4)	Expert Systems – Methods and Tools	Sture Hägglund
Datalingvistik (C2)	Computational Linguistics	Richard Hirsch
Lingvistik grundkurs (C1, D4, Y4)	Introduction to Linguistics	Richard Hirsch
Kogn vetenskapliga kommunikationsmod (C3)	Cognitive Science Communic. Models	Nils Dahlbäck
Tekniska databaser (DI2)	Technical Databases	Jalal Maleki
Människa-datorinteraktion (SVL3)	Human Computer Interaction	Hans Holmgren
Expertsystem och kunskapsteknik (SVL3)	Expert Systems and Knowledge Eng.	Sture Hägglund
Datalagring (ADB2)	Data Structures	Hans Holmgren
Databasmetodik (SL2)	Data Base Methodology	Jalal Maleki

System Architecture: Johan Fagerström, director of undergraduate studies

Course name in Swedish	Course name in English	Examiner
Systemprogramvara (Y4)	System Software	Lars Strömberg
Processprogrammering (C3, D3, Dx3)	Concurrent Programming	Johan Fagerström
Kompilatorer och interpr. (D4, I4, Y4, SVL)	Compilers and Interpreters	Nahid Shahmehri
Programmeringsteori II (C4)	Programming Theory II	Johan Boye
Kompilatorkonstruktion (C3, D4)	Compiler Construction	Rober Bilos

Programmeringsteori (C3)	Programming Theory	Nahid Shahmehri
Realtids- och processprogr. (I4, M4, Y4, DI2)	Real Time and Concurrent Progr.	Anders Törne
Distribuerade system(C4, D4)	Distributed Systems	Rober Bilos
Formella språk och automatateori (C2, D4)	Formal Languages and Autom. Theory	Johan Boye
Diskret simuleringsteknik (D4, I4, Y4)	Discrete Simulation	Zebo Peng
Datornät (C4, D5, I4, Y5)	Computer Networks	Lars Strömberg
Datorarkitektur (C4, D4, Y4)	Advanced Computer Architecture	Krzysztof Kuchcinski
Datorstödd elektronikkonstruktion (C4, Y4)	Computer Aided Design of Electr.	Krzysztof Kuchcinski
Datorarkitektur, projektkurs (D5, Y5)	Adv. Computer Architecture Project	Lars Viklund
Operativsystem o systemprogramvara (DI2)	Operating Systems a System Progr	Johan Fagerström
Datornät och distribuerade system (DI2)	Comp. Networks a Distr. Systems	Lars Strömberg
Datorarkitektur (DI2)	Advanced Computer Architecture	Krzysztof Kuchcinski
Datorsystem och programmering (EI1, MI1)	Intr. to Computer Science and Progr.	Peter Johannesson
Mätdatorsystem (EI2)	Computers in Measurements	Peter Johannesson
Programmering, fk (EI2)	Programming, advanced course	Peter Johannesson
Datorteknik (ADB2)	Elementary Computer Architecture	Katarina Sunnerud
Datorteknik (SVL2)	Elementary Computer Architecture	Zebo Peng

Economic Information Systems: Lennart Ohlsén, director of undergraduate studies

Course name in Swedish	**Course name in English**	**Examiner**
Industriell Ekonomi DK (M4)	Industr. Economics and Management	Lennart Ohlsén
Industriell Ekonomi P (M4)	Industr. Economics and Management	Lennart Ohlsén
Industriell Ekonomi (M2)	Industr. Economics and Management	Lennart Ohlsén
Redovisning och budgetering (I2)	Accounting and Budgeting	Jörgen Andersson
Redovisning och budgetering, ök (I2, I3, I4)	Accounting and Budgeting, adv. course	Lennart Ohlsén
Informationssystem, grundkurs (I3)	Informations Systems, basic course	Mehran Noghabai
Tekn o ekon utvärd. av datorsystem (I4,D4,M4)	Techn. Econom. Eva.l of In.f Syst.	Lars Mattsson
Ekonomiska informationssystem, ak (I4, D4)	Economic Information Systems	Bengt Savén
Ekonomiska informationssystem, fk (I4, D4)	Economic Inf Systems, adv. course	Björn Helander
Modellbyggnad (I3)	Modelling	Thomas Bennet
Kontorsinformationssystem (I4, M4)	Office Information Systems	Kristina Larsen
Informationssystem och MPS (SVL3)	Information Systems and MPS	Bengt Savén

C.2 Teaching staff

The following persons from IDA have been responsible for one or more courses:

Jörgen Andersson, B. Sc.
Pia Arendell, B. Sc.
Thomas Bennet, B. Sc.
Rober Bilos, Ph. D.
Andreas Björklind, M. Sc.
Johan Boye, M. Sc.
Douglas Busch, Ph. D.
Stefan Cronholm, B. Sc.
Johan Fagerström, Ph. D.
Per-Olof Fjällström, Ph. D.
Peter Fritzson, Ph. D.
Göran Goldkuhl, Ph. D.
Anders Haraldsson, Ph. D.
Björn Helander, B. Sc.
Richard Hirsch, Ph. D.
Roland Hjerppe, M. Sc.
Hans Holmgren, M. Sc.
Sture Hägglund, Ph. D.
Peter Johannesson, M. Sc.
Torbjörn Jonsson, M. Sc.
Arne Jönsson, Tech. Lic.
Mariam Kamkar, Tech. Lic.
Krzysztof Kuchcinski, Ph. D.
Christian Krysander, M. Sc.
Karin Larsen, B. Sc.
Jonas Löwgren, Ph. D.
Jalal Maleki, Ph. Lic.
Lars Mattsson, M. Sc.
Ivan Nilsson, B. Sc.
Rolf Nilsson, B. Sc.
Ulf Nilsson, Ph. D.
Mehran Noghabai, M. Sc.
Thomas Padron McCarthy, M. Sc.
Jan Petersson, M. Sc.
Lennart Ohlsén, M. Sc.
Tommy Olsson, M. Sc.
Zebo Peng, Ph. D.
Ivan Rankin, Ph. Lic.
Lise-Lotte Raunio, M. Sc.
Annie Röstlinger, B. Sc.
Erik Sandewall, Ph. D.
Bengt Savén, M. Sc.
Nahid Shahmehri, Ph. D.
Eric Stoy, M. Sc.
Lars Strömberg, Tech. Lic.
Katarina Sunnerud, M. Sc.
Stefan Svenberg, M. Sc.
Eva-Chris Svensson, M. Sc.
Toomas Timpka, M. D.
Åke Thurée, M. Sc.
Anders Törne, Ph. D.
Olle Willén, B. Sc.
Lars Viklund, M. Sc.

Figure C-1. *Barbara Ekman, course secretary.*

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 over 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.

Table D–1. *Computer resources, November 1992.*

250	Sun workstations
13	Sun file servers
4	HP workstations
1	IBM AS/400 MiniMainframe
80	Macintoshes
60	IBM PC compatibles
2	VAX 3200
20	Laser printers

Our main computer resources are the Sun SPARCstations and fileservers.

Economic Information Systems group, who joined the department in October 1990, have just acquired an IBM/AS400.

In addition there are lots of smaller computers (MicroVax, PDP-11s, Macintoshes and other PC’s of various kinds) and around 20 laser printers.

There is also special purpose equipment for specific research projects.

The schematic pictures on the following pages shows the local network and the accessible computer systems.

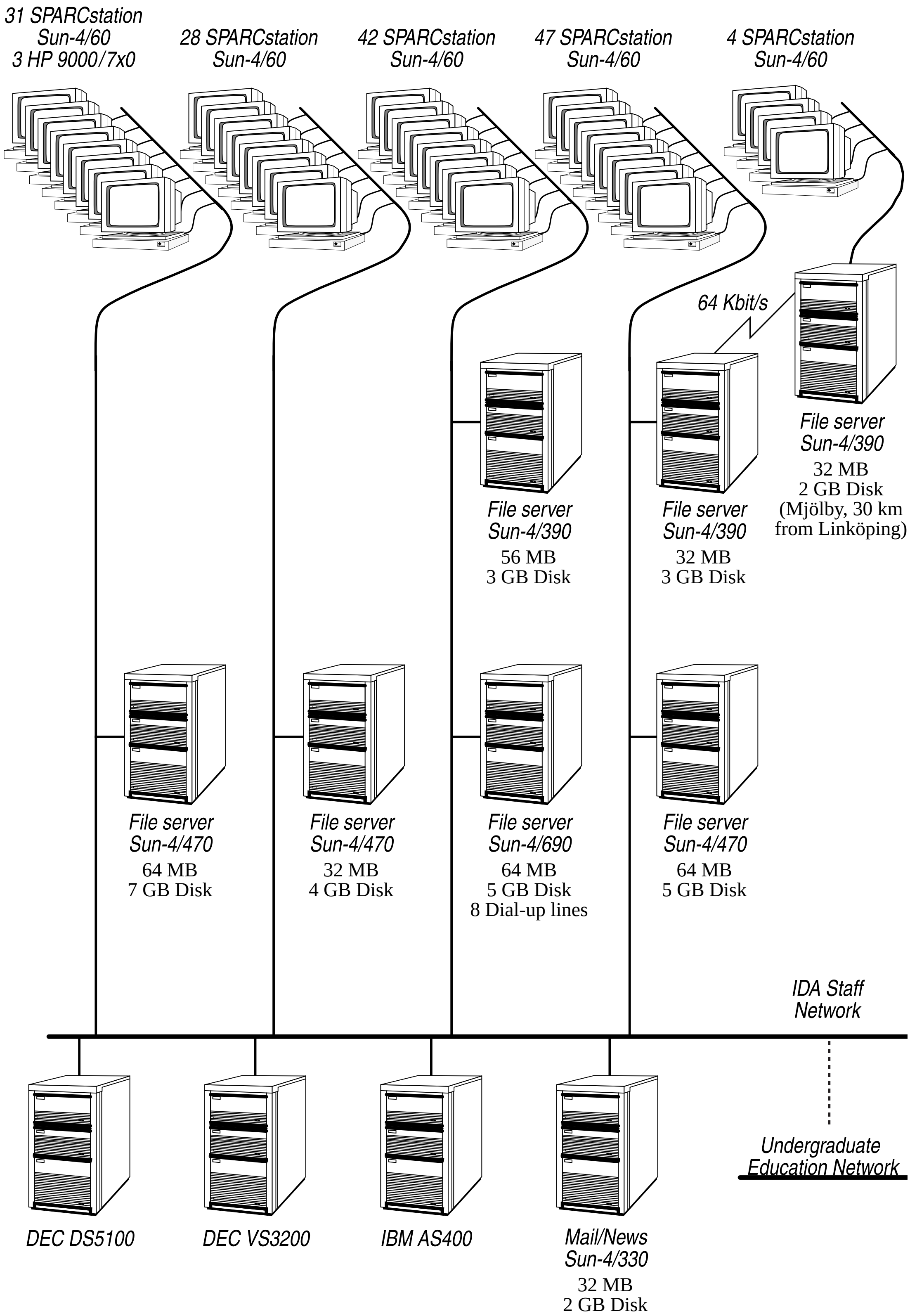


Figure D-1. *Main computer resources for the department staff.*

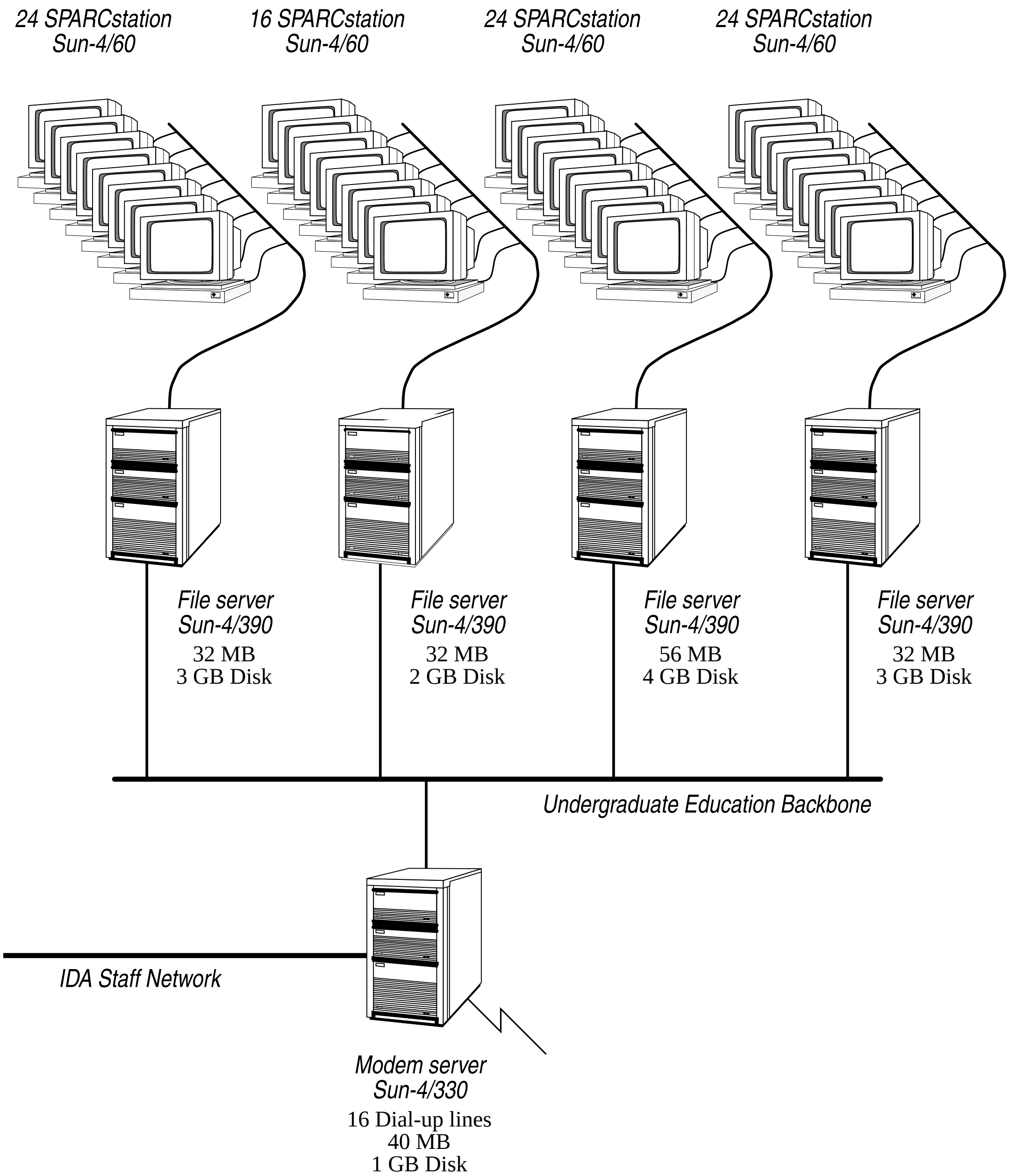


Figure D-2. *Main computer resources for the undergraduate education.*

Figure D-3. *Interior from computer room. Jaime Villegas at the IBM AS400.*

Appendix E
Publications

E.1 Dissertations

Linköping Studies in Science and Technology. Dissertations.

No 14 **Anders Haraldsson:** A Program Manipulation System Based on Partial Evaluation, 1977.
No 17 **Bengt Magnhagen:** Probability Based Verification of Time Margins in Digital Designs, 1977.
No 18 **Mats Cedwall:** Semantisk analys av processbeskrivningar i naturligt språk, 1977.
No 22 **Jaak Urmi:** A Machine Independent LISP Compiler and its Implications for Ideal Hardware, 1978.
No 33 **Tore Risch:** Compilation of Multiple File Queries in a Meta-Database System, 1978.
No 51 **Erland Jungert:** Synthesizing Database Structures from a User Oriented Data Model, 1980.
No 54 **Sture Hägglund:** Contributions to the Development of Methods and Tools for Interactive Design of Applications Software, 1980.
No 55 **Pär Emanuelson:** Performance Enhancement in a Well-Structured Pattern Matcher through Partial Evaluation, 1980.
No 58 **Bengt Johnsson, Bertil Andersson:** The Human-Computer Interface in Commercial Systems, 1981.
No 69 **H. Jan Komorowski:** A Specification of an Abstract Prolog Machine and its Application to Partial Evaluation, 1981.
No 71 **René Reboh:** Knowledge Engineering Techniques and Tools for Expert Systems, 1981.
No 77 **Östen Oskarsson:** Mechanisms of Modifiability in Large Software Systems, 1982.
No 94 **Hans Lunell:** Code Generator Writing Systems, 1983.
No 97 **Andrzej Lingas:** Advances in Minimum Weight Triangulation, 1983. (Out of stock)
No 109 **Peter Fritzson:** Towards a Distributed Programming Environment based on Incremental Compilation, 1984.
No 111 **Erik Tengvald:** The Design of Expert Planning Systems. An Experimental Operations Planning System for Turning, 1984.
No 155 **Christos Levcopoulos:** Heuristics for Minimum Decompositions of Polygons, 1987.
No 165 **James W. Goodwin:** A Theory and System for Non-Monotonic Reasoning, 1987.
No 170 **Zebo Peng:** A Formal Methodology for Automated Synthesis of VLSI Systems, 1987.
No 174 **Johan Fagerström:** A Paradigm and System for Design of Distributed Systems, 1988.
No 192 **Dimiter Driankov***:* Towards a Many-Valued Logic of Quantified Belief, 1988.
No 213 **Lin Padgham:** Non-Monotonic Inheritance for an Object Oriented Knowledge Base, 1989.
No 214 **Tony Larsson:** A Formal Hardware Description and Verification Method, 1989.
No 221 **Michael Reinfrank:** Fundamentals and Logical Foundations of Truth Maintenance, 1989.
No 239 **Jonas Löwgren: Knowledge-Based Design Support and Discourse Management in User Interface Management Systems**, 1991.
No 244 **Henrik Eriksson: Meta-Tool Support for Knowledge Acquisition**, 1991.
No 252 **Peter Eklund:** An Epistemic Approach to Interactive Design in Multiple Inheritance Hierarchies, 1991.
No 258 **Patrick Doherty:** NML3 - A Non-Monotonic Formalism with Explicit Defaults, 1991.
No 260 **Nahid Shahmehri:** Generalized Algorithmic Debugging Technique, 1991.
No 264 **Nils Dahlbäck:** Representations of Discourse, Cognitive and Computational Aspects, 1992.
No 265 **Ulf Nilsson:** Abstract Interpretations and Abstract Machines: Contributions to a Methodology for the Implementation of Logic Programs, 1992

No 270 **Ralph Rönnquist:** Theory and Practice of Tense-bound Object References, 1992.
No 273 **Björn Fjellborg:** Pipeline Extraction for VLSI Data Path Synthesis, 1992.
No 276 **Staffan Bonnier:** A Formal Basis for Horn Clause Logic with External Polymorphic Functions, 1992.
No 277 **Kristian Sandahl:** Developing Knowledge Management Systems with an Active Expert Methodology, 1992.
No 281 **Christer Bäckström:** Computational Complexity of Reasoning about Plans, 1992.
No 292 **Mats Wirén:** Studies in Incremental Natural-Language Analysis, 1992.

Dissertation by IDA member published elsewhere.

Lars Ahrenberg: Interrogative Structures of Swedish: Aspects of the Relation between Grammar and Speech Acts. (Reports from Uppsala University Department of Linguistics No. 15, 1987).

E.2.1 PhD Dissertation Abstracts

No 252
AN EPISTEMIC APPROACH TO INTERACTIVE DESIGN IN MULTIPLE INHERITANCE HIERARCHIES

Peter Eklund

The thesis explores the advantages of a marriage between a “mixed dialogue” interaction metaphor and belief logics and in particular how the two can be used for multiple inheritance hierarchy design. The result is a design aid which produces critiques of multiple inheritance hierarchies in terms of their logical consequences. The work draws on a number of theoretical issues in artificial intelligence, namely belief logics and multiple inheritance reasoning, applying “belief sets” to dialogue and using multiple inheritance hierarchy design as a specific application.

The work identifies three design modes for the interface which reflect the intuitions of multiple inheritance hierarchy design and conform to an existing user modeling framework. A major survey of multiple inheritance hierarchies leads to the allocation of a precise inheritance semantics for each of these design modes. The semantics enable a definition of entailment in each, and are in turn used to determine the translation from inheritance networks to belief sets.

The formal properties of belief sets imply that when an ambiguous inheritance network is encountered more than one belief set must be created. Each belief set provides an alternative interpretation of the logical consequences of the inheritance hierarchy. A “situations matrix” provides the basic referent data structure for the system we describe. Detailed examples of multiple inheritance construction demonstrate that a significant design aid results from an explicit representation of operator beliefs and their internalization using an epistemic logic.

No 258
NML3 - A NON-MONOTONIC FORMALISM WITH EXPLICIT DEFAULTS

Patrick Doherty

The thesis is a study of a particular approach to defeasible reasoning based on the notion of an information state consisting of a set of partial interpretations constrained by an information ordering. The formalism proposed, called NML3, is a non-monotonic logic with explicit defaults and is characterized by the following features: (1) The use of the strong Kleene three-valued logic as a basis. (2) The addition of an explicit default operator which enables distinguishing tentative conclusions from ordinary conclusions in the object language. (3) The use of the technique of preferential entailment to generate non-monotonic behavior. The central feature of the formalism, the use of an explicit default operator with a model theoretic semantics based on the notion of a partial interpretation, distinguishes NML3 from the existing formalisms. By capitalizing on the distinction between tentative and ordinary conclusions, NML3 provides increased expressibility in comparison to many of the standard non-monotonic formalisms and greater flexibility in the representation of subtle aspects of default reasoning.

In addition to NML3, a novel extension of the tableau-based proof technique is presented where a signed formula is tagged with a set of truth values rather than a single truth value. This is useful if the tableau-based proof technique is to be generalized to apply to the class of multi-valued logics. A refutation proof procedure may then be used to check logical consequence for the base logic used in NML3 and to provide a decision procedure for the pro-positional case of NML3.

A survey of a number of non-standard logics used in knowledge representation is also provided. Various formalisms are analyzed in terms of persistence properties of formulas and their use of information structures.

No 260

GENERALIZED ALGORITHMIC DEBUGGING TECHNIQUE

Nahid Shahmehri

This thesis presents a novel method for semi-automatic program debugging: the *Generalized Algorithmic Debugging Technique*, GADT. The notion of *declarative algorithmic debugging* was first introduced for logic programming. However, this is the first algorithmic debugging method based on the principle of declarative debugging, which can handle debugging of programs written in an *imperative language* including *loops* and *side-effects*. In order to localize a bug, the debugging algorithm incrementally acquires knowledge about the debugged program. This knowledge is supplied by the user. The algorithm terminates when the bug has been localized to within the body of a procedure or an explicit loop.

The generalized algorithmic debugging method uses *program transformation* and *program flow analysis* techniques to transform the subject program to a largely side-effect free internal form, which is used for bug localization. Thus, this method defined two views of a program: (1) *the user view* which is the original program with side-effects and (2) *the transformed view* which is the transformed side-effect free version of the original program. Transparent program debugging is supported by keeping a mapping between these two views. The bug localization algorithm works on the transformed version, whereas user interactions are defined in terms of the user view.

We have presented a *general* technique which it is not based on any ad-hoc assumptions about the subject program. The *flexibility* of this method has made it possible to further improve the bug localization algorithm by employing a number of other techniques, i.e. *program slicing* and *test database lookup*, thus increasing the degree of automation provided by GADT. These extensions are topics for ongoing research projects and further work.

A survey and evaluation of a number of automated debugging systems and the techniques behind these systems is also presented. We have introduced several criteria for comparing these techniques with GADT. A prototype implementation of the generalized algorithmic debugging technique has been done to verify its feasibility, and to provide feedback for further refinement of the method.

No 264

REPRESENTATIONS OF DISCOURSE: COGNITIVE AND COMPUTATIONAL ASPECTS

Nils Dahlbäck

This work is concerned with empirical studies of cognitive and computational aspects of discourse representations. A more specific aim is to contribute to the development of natural language interfaces for interaction with computers, especially the development of representations making possible a continuous interactive dialogue between user and system.

General issues concerning the relationship between human cognitive and computational aspects of discourse representations were studied through an empirical and theoretical analysis of a psychological theory of discourse coherence, Johnson-Laird’s theory of mental models. The effects of previous background knowledge of the domain of discourse on the processing of the types of texts used in previous work was demonstrated. It was argued that this demonstration does not invalidate any of the basic assumptions of the theory, but should rather be seen as a modification or clarification. This analysis also suggested that there are principled limitations on what workers in computational linguistics can learn from psychological work on discourse processing. While there is much to be learned from empirical investigations concerning what kinds of knowledge is used during the participation in dialogues and in the processing of other kinds of connected discourse, there is less to be learned concerning how this is represented in detail. One specific consequence of this position is the claim that computational theories of discourse are in principle theories only of the processing of discourse in computers, as far as the detailed representational account is concerned.

Another set of studies used the so-called *Wizard of Oz*-method, i.e. dialogues with simulated natural language interfaces. The focus was on the dialogue structure and the use of referring and anaphoric expressions. The analysis showed that it is possible to describe the structure of these dialogues using the LINDA-model, the basic feature of which is the partitioning of the dialogues in a number of initiative-response (IR) units. The structure can be described using a simple context free grammar. The analysis of the referring expressions also shows a lack of some of the complexities encountered in human dialogues. The results suggests that it is possible to use computationally simpler methods of dialogue management than what has hitherto been assumed, both for the dialogue management and the resolution of anaphoric references.

No 265

ABSTRACT INTERPRETATION AND ABSTRACT MACHINES: CONTRIBUTIONS TO A METHODOLOGY FOR THE IMPLEMENTATION OF LOGIC PROGRAMS

Ulf Nilsson

Abstract: Because of the conceptual gap between high-level programming languages like logic programming and existing hardware, the problem of compilation often becomes quite hard. This thesis addresses two ways of narrowing this gap --- program analysis through *abstract interpretation* and the introduction of intermediate languages and *abstract machines*. By means of abstract interpretations it is possible to infer program properties which are not explicitly present in the program --- properties which can be used by a compiler to generate specialized code. We describe a *framework* for *constructing* and *computing* abstract interpretations of *logic programs with equality*. The core of the framework is an abstract interpretation called the *base interpretation* which provides a model of the run-time behaviour of the program. The model characterized by the base interpretation consists of the set of all reachable computation states of a transition system specifying an operational semantics reminiscent of SLD-resolution. This model is in general not effectively computable, however, the base interpretation can be used for constructing new abstract interpretations which *approximate* this model. Our base interpretation combines both a simple and concise formulation with the ability of inferring a wide range of program properties. In addition the framework also supports efficient computing of approximate models using a *chaotic iteration strategy*. However, the framework supports also other computation strategies.

We also show that abstract interpretations may form a basis for implementation of *deductive databases*. We relate the *magic templates* approach to *bottom-up evaluation* of deductive databases with the base interpretation of C. Mellish and prove that they not only specify isomorphic models but also that the computations which lead up to those models are isomorphic. This implies that methods (for instance, evaluation and transformation techniques) which are applicable in one of the fields are also applicable in the other. As a side-effect we are also able to relate so-called “top-down” and “bottom-up” abstract interpretations.

Abstract machines and intermediate languages are often used to bridge the conceptual gap between language and hardware. Unfortunately --- because of the way they are presented -- it is often difficult to see the relationship between the high-level and intermediate language. In the final part of the thesis we propose a methodology for *designing abstract machines* of logic programming languages in such a way that much of the relationship is preserved all through the process. Using *partial deduction* and other transformation techniques a source program and an interpreter are “compiled” into a new program consisting of “machine code” for the source program and an abstract machine for the machine code. Based upon the appearance of the abstract machine the user may choose to modify the interpreter and repeat the process until the abstract machine reaches a suitable level of abstraction. We demonstrate how these techniques can be applied to derive several of the control instructions of Warren’s Abstract Machine, thus complementing previous work by P. Kursawe who reconstructed several of the unification instructions using similar techniques.

No 270

THEORY AND PRACTICE OF TENSE-BOUND OBJECT REFERENCES

Ralph Rönnquist

Abstract: The work presented in this thesis is a study of a formal method for representation of time and development. It constitutes a formalisation of the conception that change and development is attributed to objects, which then occur in time structures of versions. This conception is taken as the foundation for a formal temporal logic, LITE, which is then defined in syntax, semantics and interpretation.

The resulting logic is studied with respect to how it captures temporal aspects of developments. In particular the way apparently atemporal formulas convey implicit synchronisations between object versions is studied. This includes the temporal implications of verification, of specifying database invariances, and the intuitions regarding propagation of change for composite objects.

The logic is also applied and discussed for a few particular process characterisation tasks. In this logic, processes are generally characterised in terms of how data changes rather than which actions are performed. As a result, the same characterisation can be used for both sequential and parallel execution environments.

The conceptualisation of development and the formal semantics is further utilised for introducing temporal qualifications in a terminological logic. The theoretical issues in terminological logics are relatively well understood. They therefore provide an excellent testbed for experimenting with the usefulness of the LITE temporal logic.

No 273

PIPELINE EXTRACTION FOR VLSI DATA PATH SYNTHESIS

Björn Fjellborg

Abstract: An important concern in VLSI design is how to exploit any inherent concurrency in the designed system. By applying pipelining, a high degree of concurrency and efficiency can be obtained. Current design tools for automatic pipeline synthesis exploit this by pipelining loops in the design. However, they lack the ability to automatically select the parts of the design that can benefit from pipelining. Pipeline extraction performs this task as a first step of pipeline synthesis. This thesis addresses the problem of pipeline extraction from a general perspective, in that the search for pipelines is based on detecting potential for hardware sharing and temporal overlap between the individual tasks in a design. Thus loops appear as an important special case, not as the central concept. A formalism for reasoning about the properties underlying pipelinability from this perspective has been developed. Using that, a series of results on exactly what mutual dependencies between operations that allow a pipelined schedule with static control sequence to be constructed are proven. Furthermore, an evaluation model for designs with mixed pipelined and non-pipelined parts has been formulated. This model and the formalism's concept of pipelinability form the basis for a heuristics-guided branch and bound algorithm that extracts an optimal set of pipelines from a high-level algorithmic design specification. This is implemented in the pipeline extraction tool PiX, which operates as a preprocessor to the CAMAD VLSI design system. The extraction is realized as transformations on CAMAD's Petri net design representation. For this purpose, a new model for representing pipeline constraints by Petri nets has been developed. Preliminary results from PiX are competitive with those from existing pipeline synthesis tools and also verify a capability to extract cost-efficient pipelines from designs without apparent pipelining properties

No 276

A FORMUL BASIS FOR HORN CLAUSE LOGIC WITH EXTERNAL POLYMORPHIC FUNCTIONS

Staffan Bonnier

Abstract: Horn clause logic has certain properties which limit its usefulness as a programming language. In this thesis we concentrate on two such limitations:

(P1) Horn clause logic has no support for the (re-) use of external software modules. Thus, procedures which are more easily solved in other kinds of languages still have to be encoded as Horn Clauses.

(P2) To work with a predefined structure like integer arithmetic, one has to axiomatize it by a Horn clause program. Thus functions of the structure are to be represented as predicates of the program.

When extending the Horn clause formalism, there is always a trade-off between general applicability and purity of the resulting system. There have been many suggestions for solving one or both of these problems. Most of the solutions are based on one of the following two strategies:

(a) To allow new operational features, such as access to low-level constructs of other languages.

(b) To introduce new language constructs, and to support them by a formal semantics.

In this thesis a solution to problems (P1) and (P2) is suggested. It combines the strategies of (a) and (b) by limiting their generality: We allow Horn clause programs to call procedures written in arbitrary languages. It is assumed however that these procedures compute typed first-order *functions.* A clean declarative semantics is obtained by viewing the procedures as a set χ of equations. This set is completely determined by two parameters. The *types* of the procedures, and the *input-output* relationship they induce.As a first step towards an operational semantics, we show how the computation of correct answers can be reduced to solving equations modulo χ. For the purpose of solving such equations a *type driven narrowing* algorithm (TDN) is developed and proved complete. TDN furthermore benefits from the assumption that polymorphic functions are parametrical. Still TDN is impractical since it may create infinitely branching search trees. Therefore a finitely terminating version of TDN (FTDN) is considered. Any unification procedure satisfying the operational restrictions imposed on FTDN is necessarily incomplete. When only monomorphic types of infinite size are present, we prove however that FTDN generates a complete set of answers whenever such a set is generated by *some* procedures satisfying the restrictions. A necessary condition for TDN and FTDN to work properly is that the set of equations to be solved is well-typed. We therefore give a sufficient condition on programs and goals which ensures that only well-typed sets of equations are generated.

No 277

DEVELOPING KNOWLEDGE MANAGEMENT SYSTEMS WITH AN ACTIVE EXPERT METHODOLOGY

Kristian Sandahl

Knowledge Management, understood as the ability to store, distribute and utilize human knowledge in an organization, is the subject of this dissertation. In particular we have studied the design of methods and supporting software for this process. Detailed and systematic description of the design and development processes of three case-study implementations of Knowledge Management software are provided. The outcome of the projects is explained in terms of an *Active Expert development methodology,* which is centered around support for a domain expert to take a substantial responsibility for the design and maintenance of a Knowledge Management system in a given area of application.

Based on the experiences from the case studies and the resulting methodology, an environment for automatically supporting Knowledge Management was designed in the KNOWLEDGE-LINKER research project. The vital part of this architecture is a *knowledge acquisition tool*, used directly by the experts in creating and maintaining a knowledge base. An elaborated version of the Active Expert development methodology was then formulated as the result of applying the KNOWLEDGE-LINKER approach in a fourth case study. This version of the methodology is also accounted for and evaluated together with the supporting KNOWLEDGE-LINKER architecture.

No 281

COMPUTATIONAL COMPLEXITY OF REASONING ABOUT PLANS

Christer Bäckström

The *artificial intelligence (AI) planning problem* is known to be very hard in the general case. Propositional planning is PSPACE-complete and first-order planning is undecidable. Many planning researchers claim that all this *expressiveness* is needed to solve real problems and some of them have abandoned theory-based planning methods in favour of seemingly more efficient methods. These methods usually lack a *theoretical foundation* so not much is known about the *correctness* and the *computational complexity* of these. There are, however, many applications where both provable correctness and efficiency are of major concern, for instance, within *automatic control*.

We suggest in this thesis that it might be possible to stay within a well-founded theoretical framework and still solve many interesting problems tractably. This should be done by *identifying restrictions* on the planning problem that improve the complexity figure while still allowing for interesting problems to be modelled. Finding such restrictions may be a non-trivial task, though. As a first attempt at finding such restrictions we present a variant of the traditional STRIPS formalism, the *SAS^+ formalism*. The SAS^+ formalism has made it possible to identify certain restrictions which define a *computationally tractable* planning problem, the *SAS^+-PUS problem*, and which would not have been easily identified using the traditional STRIPS formalism. We also present a polynomial-time, sound and complete algorithm for the SAS^+-PUS problem.

We further prove that the SAS^+ formalism in its unrestricted form is *equally expressive* as some other well-known formalisms for propositional planning. Hence, it is possible to compare the SAS^+ formalism with these other formalisms and the complexity results carry over in both directions.

Furthermore, we analyse the computational complexity of various subproblems lying between unrestricted SAS^+ planning and the SAS^+-PUS problem. We find that most planning problems (not only in the SAS^+ formalism) allow instances having *exponentially-sized minimal solutions* and we argue that such instances are not realistic in practice.

We conclude the thesis with a brief investigation into the relationship between the *temporal projection* problem and the *planning* and *plan validation* problems.

E.3 Licentiate Of Engineering Theses

Linköping Studies in Science and Technology. Theses

No 17 **Vojin Plavsic:** Interleaved Processing of Non-Numerical Data Stored on a Cyclic Memory. (Available at: FOA, Box 1165, S-581 11 Linköping, Sweden. FOA Report B30062E)

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 Architecture based on Asynchronous Processes, 1985.
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 Behavior of the Greedy Triangulation, 1986.
No 104 **Shamsul I. Chowdhury: S**tatistical 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, 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 User Acceptance of Expert Systems, 1987.
No 139 **Christer Bäckström:** Reasoning about Interdependent Actions, 1988.
No 140 **Mats Wirén:** On Control Strategies and Incrementality in Unification-Based Chart Parsing, 1988.
No 146 **Johan Hultman:** A Software System for Defining and Controlling Actions in a Mechanical System, 1988.
No 150 **Tim Hansen:** Diagnosing Faults using Knowledge about Malfunctioning Behavior, 1988.
No 165 **Jonas Löwgren:** Supporting Design and Management of Expert System User Interfaces, 1989.
No 166 **Ola Petersson:** On Adaptive Sorting in Sequential and Parallel Models, 1989.
No 174 **Yngve Larsson:** Dynamic Configuration in a Distributed Environment, 1989.
No 177 **Peter Åberg:** Design of a Multiple View Presentation and Interaction Manager, 1989.
No 181 **Henrik Eriksson:** A Study in Domain-Oriented Tool Support for Knowledge Acquisition, 1989.
No 184 **Ivan Rankin:** The Deep Generation of Text in Expert Critiquing Systems, 1989.
No 187 **Simin Nadjm-Tehrani:** Contributions to the Declarative Approach to Debugging Prolog Programs, 1989.
No 189 **Magnus Merkel:** Temporal Information in Natural Language, 1989.
No 196 **Ulf Nilsson:** A Systematic Approach to Abstract Interpretation of Logic Programs, 1989.
No 197 **Staffan Bonnier:** Horn Clause Logic with External Procedures: Towards a Theoretical Framework, 1989.
No 203 **Christer Hansson:** A Prototype System for Logical Reasoning about Time and Action, 1990.
No 212 **Björn Fjellborg:** An Approach to Extraction of Pipeline Structures for VLSI High-Level Synthesis, 1990.
No 230 **Patrick Doherty:** A Three-Valued Approach to Non-Monotonic Reasoning, 1990.
No 237 **Tomas Sokolnicki:** Coaching Partial Plans: An Approach to Knowledge-Based Tutoring, 1990.
No 250 **Lars Strömberg:** Postmortem Debugging of Distributed Systems, 1990.
No 253 **Torbjörn Näslund:** SLDFA-Resolution - Computing Answers for Negative Queries, 1990.
No 260 **Peter D. Holmes:** Using Connectivity Graphs to Support Map-Related Reasoning, 1991.
No 283 **Olof Johansson:** Improving Implementation of Graphical User Interfaces for Object-Oriented Knowledge-Bases, 1991.
No 298 **Rolf G Larsson:** Aktivitetsbaserad kalkylering i ett nytt ekonomisystem, 1991.
No 318 **Lena Strömbäck:** Studies in Extended Unification-Based Formalism for Linguistic Description: An Algorithm for Feature Structures with Disjunction and a Proposal for Flexible Systems, 1992.
No 319 **Mikael Pettersson:** DML-A Language and System for the Generation of Efficient Compilers from Denotational Specification, 1992.
No 326 **Andreas Kågedal:** Logic Programming with External Procedures: an Implementation, 1992.
No 328 **Patrick Lambrix:** Aspects of Version Management of Composite Objects, 1992.

No 333 **Xinli Gu:** Testability Analysis and Improvement in High-Level Synthesis Systems, 1992.
No 335 **Torbjörn Näslund:** On the Role of Evaluations in Iterative Development of Managerial Support Systems, 1992.
No 348 **Ulf Cederling:** Industrial Software Development - a Case Study, 1922.
No 352 **Magnus Morin:** Predictable Cyclic Computations in Autonomous Systems: A Computational Model and Implementation. 1992.

E.4 Master of Engineering Theses

No 1 **Patrick Lambrix:** Management of Historical Information of Composite Objects, 1992.

E.5 External Publications since 1990

Refereed papers published in books, journals or international conference proceedings.

L. Ahrenberg: A Grammar Combining Phrase Structure and Field Structure. In *Papers presented to the 13th International Conference on Computational Linguistics*, Helsinki 20–25 Aug., 1990, Vol. 2 pp. 1–6, 1990.

L. Ahrenberg, A. Jönsson, N. Dahlbäck: Discourse Representation and Discourse Management for a Natural Language Dialogue System. In *Papers from the 2nd Nordic Conf. on Text Comprehension in Man and Machine*, Täby April 25–27, 1990, pp 1-14.

L. Ahrenberg, S. Svenberg: Conceptual text representation for multi-lingual generation and translation. In *Proc. of the Scandinavian Conference on Computational Linguistics*, Bergen, 28-30 Nov. 1991. Bergen, Norwegian Computing Centre for the Humanities,1992.

L. Ahrenberg: On the integration and scope of segment-based theories of discourse. To appear in *Papers from the Third Nordic Conference on Text Comprehension in Man and Machine*, Linköping, 21-23 April, 1992.

N. Andersson, P. Fritzson: Comparative Evaluation and Industrial Application of Code Generator Generators. In *Proc. of the Nordic Workshop on Programming Environment Research*, Tampere, Jan. 8-10, 1992.

E. Arborelius, S. Bremberg, T. Timpka: What is going on when the general practitioner doesn't grasp the consultation? *Family Practice* 1990, in press.

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

E. Arborelius, S. Bremberg, T.Timpka: What is going on when the general practitioner doesn't grasp the consultation? *Family Practice* 1991;8:3-9.

E. Arborelius, T. Timpka, J. Nyce: Patients comments on video-recorded consultations: The "good" GP and the "bad". *Scand. J Soc. Med* 1992, in press.

M Aug.on, P Fritzson: PARFORMAN - An assertion language for specifying behaviour when debugging parallel applications. In *Proc. of the Euromicro Workshop on Parallel and Distributed Processing*, Gran Canaria, Jan., 1993.

C. Bäckström: Logical modelling of simplified geometrical objects and mechanical assembly processes. In Su-shing Chen, editor, *Advances in Spatial Reasoning*, volume 1, pages 35-62, Norwood, NJ, 1990, Ablex.

C. Bäckström: Equivalence and Tractability Results for SAS+ Planning. In Proc. of *KR-92*, pp 126 - 137, Morgan Kaufmann, Cambridge, MA., USA, Oct., 1992.

C. Bäckström: Uncertainty and Parallelism in Action Structures. In *Proc. of Workshop: Beyond Sequential Planning at ECAI '92*, Vienna, Austria, 1992.

C. Bäckström, B. Nebel: On the Computational Complexity of Planning and Story Understanding: In *Proc. of ECAI-92*, pp 349 - 353, Wiley, Vienna, Austria, Aug. 3 -7, 1992.

C. Bäckström, I Klein: Planning in polynomial time. In G Gottlob and W Nejdl, ed, *Expert Systems in Engineering: Principles and Applications. International Workshop*, volume 462 of *Lecture Notes in Computer Science*, pages 103 - 118, Vienna, Austria, Sept. 1990, Springer Verlag.

C. Bäckström, I. Klein: Planning in polynomial time; Short paper. In Mary L Emrich et al., ed, *Methodologies for Intelligent Systems: Selected Papers*, pages 125 - 129. International Center for the Application of Information Technology, Knowville, TN, USA, Oct., 1990. Paper presented at the ISMIS'90 conference.

C Bäckström, I. Klein: Parallel non-binary planning in polynomial time. In Ray Reiter and John Mylopoulos, ed., *Proc. of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, pages 268-273, Sydney, Australia, Aug., 1991. Morgan Kaufman.

C Bäckström, I Klein: Planning in polynomial time: The SAS-PUBS class. *Computational Intelligence*, 7(3):181-197, Aug., 1991.

A. Björklind: Wide Area Information Server — a Prototype for the Personal Information Tool of the Future? In: *Teknologi och Kompetens, Proc. of the 8th. Nordic Conference for Information and Documentation*, Helsingborg, Sweden, May 19-21, 1992. Adler et. al. Ed. TLS, Stockholm, 1992 p.105-110.

A. Björklind: Document architecture for public service case processing and archival search. In: *Teknologi och Kompetens, Proc. of the 8th. Nordic Conference for Information and Documentation*, Helsingborg, Sweden, May 19-21, 1992. Adler et. al. Ed. TLS, Stockholm, 1992 p.65-70.

A. Björklind, L. Björklund, R. Hjerppe: Libraries of the Future - Real and Virtual. Presented at *15th International Essen Symposium*, Essen Oct. 12-15.

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

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

L. Björklund: Document description in the future. In: *Teknologi och Kompetens" Proc. of the 8th. Nordic Conference for Information and Documentation,* Helsingborg, Sweden, May 19-21, 1992. Adler et. al. Ed. TLS, Stockholm, 1992 p.71-74.

L. Björklund: The need for inventing user needs. Paper presented at *the Anglo-Nordic Seminar 1992: The Effects of Digitisation on Library and Information Services*, Edinburgh, Sept., 17-20. In press.

S. Bonnier, U. Nilsson, T. Näslund: A Simple Fixed Point Characterization of Three-Valued Stable Model Semantics. *Information Processing Letters*, 40:73-78, 1991.

S. Bonnier, J. Wallgren: Logic Programs with Polymorphic Types: A Condition for Static Type Checking. In *Proc. of the 3rd. International Workshop on Conditional Term Rewriting Systems*. Springer-Verlag, LNCS, 1992. To appear.

D.R. Busch: An Expressive Three-Valued Logic with Two Negations. Accepted for *the Seventh International Symposium on Methodologies of Intelligent Systems, ISMIS*, Trondheim, Norway, June 15-18, 1993. Proceedings will be published in *Lecture Notes in Artificial Intelligence* Series by Springer Verlag.

D.R. Busch: An Intuitionistic three-valued logic. Presented at *Logic Programming in AI, First Compulog-Network Workshop*, Imperial College, London, March 23 - 24, 1992, published by Imperial College, Dept. of Computing, London

D. R. Busch: Sequent Formalizations of Three-Valued Logic with Two Negations. In *Proc. of the Nordic Workshop and Summer School on Partial Semantics*, Linköping (invited paper), May, 1992, D Driankov, P. Doherty (eds.) Ellis Horwood.

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

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

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

N. Dahlbäck, A. Jönsson: An empirically based computationally tractable dialogue model. In *Proc. of the 14th Annual Meeting of the Cognitive Science Society*, Bloomington, Indiana, 1992.

N. Dahlbäck: Pronoun usage in NLI-dialogues. In *Papers from the Third Nordic Conference on Text Comprehension in Man and Machine,* Linköping, 21-23 April, 1992.

N. Dahlbäck, A. Jönsson, L. Ahrenberg: Wizard-of-Oz studies - Why and How. To appear in *Proc. from the International Workshop on Intelligent User Interfaces*, Orlando, Florida, Jan., 1992.

P. Deransart, J. Maluszynski: What kind of Grammars are Logic Programs. In *P. Saint-Dizier and S. Szpakowicz, editors, Logic and Logic Grammars for Language Processing*. Ellis Horwood, 1990.

P. Doherty: Preliminary Report: NM3 – A Three-Valued Non-Monotonic Formalism, *V, Z. Raz et. al. Methodologies for Intelligent Systems,. ISMIS* 90, 1990.

P. Doherty, D. Driankov: A non-monotonic fuzzy logic. Presented at IFSA'91 – The World Congress on Fuzzy Logic.

P. Doherty: Nm3 - a three-valued cumulative non-monotonic formalism. In *the Proc. of Logics in AI, European Workshop JELIA'90*, Amsterdam, vol. 478 of *Lecture Notes in Artificial Intelligence*, pp 196-211. Springer-Verlag, 1990.

P Doherty: A constraint-based approach to proof procedures for multi-valued logics. In *Proc. of the 1st World Conference on Fundamentals of Artificial Intelligence, WOCFAI,* July, 1991. Also published as Technical Report LiTH-IDA-R-91-13. To appear in a special issue of Studio Logica, 1993.

P Doherty, W Lukaszewicz: NML3 - A Non-Monotonic Logic with Explicit Defaults. In *Journal of Applied Non-Classical Logics*, 2(1):9-48, 1992.

P. Doherty, W. Lukaszewicz: Fonml3 - A first order non-monotonic logic with explicit defaults. In *Proc. of the European Conference on Artificial Intelligence, ECAI-92*, Vienna, Austria, Aug., 1992.

P. Doherty, W. Lukaszewicz: Defaults as first-class citizens. In *Proc. of the 22nd International Symposium on Multiple-Valued Logic, IEEE,* May, 1992, Sendai, Japan.

P. Doherty, W. Lukaszewicz: Distinguishing between facts and default assumptions. In W. van der Hoek, J.-J. Ch. Meyer, Y.H. Tan, and C. Witteveen, eds, *Non-Monotonic Reasoning and Partial Semantics*, Ellis Horwood Workshops, chapter 1, pages 1-19, Ellis Horwood Ltd., 1992.

P. Doherty, D. Driankov, H. Hellendoorn. Fuzzy if-then-unless rules and their implementation. In *Proc. of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'92,* Palma de Mallorca, July 6-10, 1992.

P. Doherty, D. Driankov, A. Tsoukias. Partial logics and partial preferences. In *Proc. of the International Conference on Economics/Management and Information Technology 92*, pages 525 - 528, Tokyo, 1992.

P. Doherty, D. Driankov: FNML3 - Integrating Fuzzy and Non-Monotonic Logic. Extended version of IFSA'91 abstract in refereed collection of articles. Will appear in the book: *Fuzzy Logic: State of the Art* (Kluwer Academic Publishers) in 1993.

K. L. Downing: Consistency-Based Diagnosis in Physiological Domains. In *Proc. of AAAI-92,* San Jose, Cal, July, 1992.

K. L. Downing: Analyzing Genetic Regulatory Systems via Focused Envisionment. Accepted to *Qualitative Physics Workshop*, Aug., 1992.

K. L. Downing: Extending the GDE Paradigm to Medicine. Presented at *Applications of AI X: Knowledge-Based System*, Orlando, Florida, April, 1992.

K. L. Downing: Physiological Applications of Consistency-Based Diagnosis. Accepted for publication in *Journal AI in Medicine*, Feb., 1993.

W. Drabent, M. Martelli: Strict Completion of Logic Programs. *New Generation Computing*, 1990, pp 69 - 79, 1991.

W. Drabent: What is failure? or: Constructive negation by fail answers. In *ICCI Proc.*. IEEE Press, 1992.

D. Driankov: Inference with single fuzzy conditional proposition. *Int. Journal of Fuzzy Sets and Systems* vol. 24, No 1 pp. 51–63.

D. Driankov, H. Hellendorn: Towards a Logic for a Fuzzy Logic Controller. In *Proc. ECSQAU-91*, P. Krause and P. Siegel (eds.), *Lecture Notes in Computer Science*, vol. 548, Springer Verlag, pp 166-171.

D. Driankov: Fuzzy reasoning and its applications to control. In *Proc. ESSCIRC-91*, Milano, Italy.

D. Driankov, H. Hellendorn: Fuzzy Logic with If-Then-Unless Rules. In Proc. *of IEEE International Conference on Fuzzy Systems,* San Diego, USA, March, 1992.

D Driankov, P. Doherty: A non-monotonic fuzzy logic. In L. A. Zadeh and J Kacprzyk, editors, *Fuzzy Logic for the Management of Uncertainty*, chapter 9, pages 171 - 190. J Wiley and Sons, 1991.

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

P. Eklund: Negotiating Conceptual Structures: inheritance hierarchies. *Proc of the 5th Annual Workshop on Conceptual Structures*, pp 281-297. Stockholm 1990.

P. Eklund: A Framework for Negotiating Conceptual Structures. In Nagle, T., Nagle, N., Gerholz, L., and Eklund, P., editors, *Conceptual Structures: Current research and practice*. Ellis-Horwood. 1992.

P. Eklund: Negotiating Inheritance Taxonomies in Conceptual Structures. In *Proc. of 3rd Scandinavian Conference on AI*, SCAI91.

P. Eles, K. Kuchcinski, Z. Peng, M. Minea: Compiling VHDL into a High-Level Synthesis Design Representation. In *Proc of EURO-DAC'92*, pp 604 - 609, Hamburg, Sept. 7-10, 1992.

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

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

H. Eriksson: Domain-oriented Knowledge Acquisition Tool for Protein Purification Planning. *Journal of Chemical Information and Computer Science*, 32(1):90–95. 1992.

H. Eriksson, M. A. Musen: Conceptual Models for Automatic Generation of Knowledge-Acquisition Tools. In Wetter, T., Althoff, K.-D., Boose, J., Gaines, B., Linster, M., and Schmalhofer, F., editors, *Current Developments in Knowledge Acquisition: EKAW'92*, pages 14–36. Springer-Verlag, Berlin, Germany. 1992.

H. Eriksson, M. A. Musen: Meta-Tools for Software Development and Knowledge Acquisition. In *Proc. of the AAAI Workshop on Automating Software Design*, pages 43–47, San Jose, CA. 1992.

H. Eriksson, K. Sandahl, J. Brewer, B. Österlund: Reactive Planning for Chromatography. *Chemometrics and Intelligent Laboratory Systems: Laboratory Information Management,* (13):185 – 194. 1991.

H. Eriksson, K. Sandahl, G. Forslund, B. Österlund: Knowledge-Based Planning for Protein Purification. *Chemometrics and Intelligent Laboratory Systems: Laboratory Information Management,* (13):173 – 184. 1991.

H. Eriksson: A Survey of Knowledge Acquisition Techniques and Tools and their Relationship to Software Engineering. In *Journal of Systems and Software*, in press. 1992.

H. Eriksson, Y. Shahar, S. W. Tu, A. R. Puerta, M. A. Musen: Task Modeling with Reusable Problem-Solving Methods. In Boose, J. H. and Gaines, B. R., editors, *Proc. of the Seventh Banff Knowledge Acquisition for Knowledge-Based Systems Workshop*, Banff, Canada, 1992.

H. Eriksson: Specialized Knowledge Acquisition Tool Support Compared to Manual Development - A Case Study. In *Proc. of the 7th Conf. on AI Applications*, Miami Beach, FL, 1991.

J. Fagerström, P. Fritzson, J. Ringström, M. Pettersson: A Data-Parallel Language and its Compilation to a Formally Defined Intermediate Language. In *Proc of ICCI'92 -the 4:th International Conference on Computing and Information,* Toronto, May 28-30, 1992.

B. Fjellborg: A General Strategy for Pipeline Extraction from High-Level Design Descriptions, *poster-presentation at European Design Automation Conference (EDAC'90),* Glasgow, Scotland, 12–15 March 1990.

B. Fjellborg: A High-Level Synthesis Tool for Exploiting Pipelines in Special-Purpose Systems, *IFIP Working Conference on Logic and Architecture Synthesis*, Paris, May 30 – June 1, 1990.

B. Fjellborg: A Petri Net Model for Pipeline Scheduling, *Sixteenth Symposium on Microprocessing and Microprogramming, EUROMICRO'90,* Short Notes Session, Amsterdam, Aug. 27–30, 1990.

B. Fjellborg: A High-Level Synthesis Tool for Pipeline Extraction. 3rd Swedish Workshop on Computer Systems Architecture, Aspenäs, 6–8 June 1990.

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

P-O Fjällström: Polyhedral Approximation of Bivariate Functions. Accepted in *Computer-Aided Design*.

L. Fribourg, H. Olsen: A Unifying View of Structural Induction and Computation Induction for Logic Programs. In the *Proc. of LOPSTR 92 - Workshop on Logic Program Synthesis and Transformation*, Manchester, July 2- 3, 1992. To appear.

D. Fritzson, P. Fritzson, L. Viklund, J. Herber: Object-Oriented Mathematical Modelling - Applied to Rolling Bearings. In *Proc. of SCAFI-92 - Studies in Computer Algebra for Industry,* Amsterdam, Nov. 5-6, 1992. Also submitted to the Journal of Computers and Structures.

P. Fritzson, T. Gyimothy, M. Kamkar, N Shahmehri: Generalized Algorithmic Debugging and Testing. In *PLDI'91, Conference on Programming Language Design and Implementation,* Toronto, Ontario, Canada June 24–28, 1991. In the *Journal ACM LOPLAS - ACM Letters of Programming Languages and Systems.*

P. Fritzson, L. Viklund, J. Herber, D. Fritzson: Industrial Application of Object-Oriented Mathematical Modeling and Computer Algebra in Mechanical Analysis. In *Proc. of TOOLS EUROPE'92,* Dortmund, Germany, March 30 - April 2, 1992. Published by Prentice Hall.

P. Fritzson, M. Auguston, N. Shahmehri: Using Assertions in Declarative and Operational Models for Automated Debugging. Accepted 1992 to the *Journal of Systems and Software*. An abridged version in *Proc. of the fifth International Conference on Software Engineering & its Applications,* Toulouse, Dec. 7-11, 1992.

P. Fritzson, **D. Fritzson**: The Need for High-Level Programming Support in Scientific Computing - Applied to Mechanical Analysis. *Computers and Structures*, Vol. 45, No. 2, pp 387 - 395, 1992.

P. Fritzson, D Fritzson, L Viklund, J Herber: Transformation of Equation-Based Real-World Models to Efficient Code. Applied to Machine Elements Geometry. In *Proc. of the 1st National Swedish Symposium on Real-Time Systems,* Uppsala, Aug. 19-20, 1991.

P. Fritzson, N. Shahmehri, M. Kamkar, T. Gyimothy: Generalized Algorithmic Debugging and Testing. Accepted 1992 to *ACM LOPLAS - Letters of Programming Language and Systems*. A previous version in *Proc. of SIGPLAN'91 Conference on Programming Language Design and Implementation*, Toronto, June 24 - 28, 1991

G. Goldkuhl: Contextual activity modelling of information systems. In *Proc. of 3rd Int. Working Conference on Dynamic Modelling of information systems*, Noordwijkerhout, 1992.

G. Goldkuhl: On the relations between information systems and organizational activities: integration and separation as thought models. In *Proc. of 15th IRIS (Information systems Research In Scandinavia)*, Oslo university, 1992.

G. Goldkuhl, S. Cronholm, K. Krysander: Adaptation of CASE tools to different systems development methods. In *Proc. of 15th IRIS (Information systems Research In Scandinavia),* Oslo university, 1992.

G. Goldkuhl, A Röstlinger: Joint elicitation of problems: An important aspect of change analysis. Accepted to *IFIP 8.2 Conference Information Systems development: Human, Social and Organizational Aspects*, May 93, Noordwijkerhout.

X. Gu, K. Kuchcinski, Z. Peng: An Approach to Testability Analysis and Improvements for VLSI Systems. In *The Euromicro Journal, Microprocessing and Microprogramming,* Nos 1-5, pp. 485 - 492, Aug., 1992.

X. Gu, K Kuchcinski, Z Peng: Testability Measure with Reconvergent Fanout Analysis and its Applications. In *The Euromicro Journal, Microprocessing and Microprogramming,* Nos 1-5, Aug., 1991.

S. Hägglund: Office applications panel: Chairman's summary. In *Proc. of the IFIP WG 9.1 Working Conference on Human Jobs and Computer Interfaces*, Tampere, Finland. North-Holland. 1991.

S. Hägglund: Expert Critiquing as a Paradigm for Giving Consultative Advice. In *Proc. of the ECAI'92 Workshop on Expert Judgement, Human Errors and Intelligent Systems*, Vienna, 1992.

S. Hägglund, I. Rankin: A knowledge-based critiquing approach to explanation generation. In *Proc. of the AAAI Spring Symposium Workshop on Explanations*, Stanford. 1992.

S. Hägglund: Investigating Consultation Paradigms for Knowledge-Based Decision Support. In *Proc. of the IFIP Working Conf. on Support Functionality in the Office Environment*, Verrijn-Stuart, Sol and Hammersley (Eds.), North-Holland, 1991.

F. van Harmelen, R. Lopez de Mantaras, J. Malec, J. Treur: Comparing Formal Specification Languages for Complex Reasoning Systems. Accepted for publication in the book *Formal Specification of Complex Reasoning Systems,* J. Treur, T. Wetter, eds, Ellis Horwood, 1993.

J. Harrius: Text Generation in Expert Critiquing Systems Using Rhetorical Structure Theory and Conceptual Graphs. *Proc. of the 5th Annual Workshop on Conceptual Structures,* Stockholm 1990.

J. Harrius: Text Generation in Expert Critiquing Systems Using Rhetorical Structure Theory and Conceptual Graphs. In T. Nagel, L. Gerholz and P. Eklund (eds). In *Current Directions in Conceptual Structure Research.* Ellis-Horwood, 1992. Earlier version presented at 5th Annual Workshop on Conceptual Structures, 1990.

J. Harrius: AREST, a System for Deep Generation of a Critique Using Aggregate Schemas. In *Proc. of the ECAI'92 Workshop on Expert Judgement, Human Errors and Intelligent Systems*, Vienna. 1992.

R. Hirsch: Powerful stuff - Discourse Processing: Social and Cognitive Aspects of Content Development in Conversation.In *Nordic Journal of Linguistics*, Vol. 15, No. 1, 65-84, 1992.

R. Hirsch: Things as they are - Reflections on the Comprehension of Metaphor in Man and Machine. To appear in *Papers from the Third Nordic Conference on Text Comprehension in Man and Machine*, Linköping, 21-23 April, 1992.

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

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

R. Hjerppe: Database visualization: Asking “What do you have?” instead of answering “What do you want?” In *Teknologi och Kompetens*. *Proc. of the 8th. Nordic Conference for Information and Documentation*, Helsingborg, Sweden, May 19-21, 1992. Adler et. al. Ed. TLS, Stockholm, 1992.

H. Holmgren, T. Timpka: System development in health care as action research. *Proc. of Medical Informatics in Europe ‘90*, Springer Verlag, Berlin, 1990.

H. Holmgren, T. Timpka, G. Goldkuhl, J. Nyce, C. Sjöberg: Argumentative Design: a necessary complement to Action Design. In *Proc. of MedInfo ’92 (World Conference of Medical Informatics)*. Amsterdam: Elsevier, 1992.

A. Jönsson, L. Ahrenberg: Extensions of a Descriptor-Based Tagging System into a Tool for the Generation of Unification-Based Grammars.In *Readings in Humanities Computing, vol. 1*.

A. Jönsson: Application-Dependent Discourse Management for Natural Language Interfaces: An Empirical Investigation. J. Pind and E.Rögnvaldsen (eds.) *Papers from the 7th Scand. Conf. of Computational Linguistics*, pp 297–307.

A. Jönsson: A Dialogue Manager Using Initiative Response Units and Distributed Control. *5th European ACL Conference*, Berlin 10–12 April.

M. Kamkar, N. Shahmehri, P. Fritzson: Bug Localization by Algorithmic Debugging and Program Slicing. In *Proc. of the Int. Conf. on Programming Language Implementation and Logic Programming*, 1990, LNCS 456, Springer-Verlag, 1990.

M. Kamkar, N. Shahmehri, P. Fritzson: Interprocedural Dynamic Slicing. In *Proc of PLILP’92 - Symposium on Programming Language Implementation and Logic Programming*, Leuven, Belgium, Aug., 1992. LNCS 631. Springer Verlag.

I. Klein, C. Bäckström: On the planning problem in sequential control. In *Proc. of the 30th IEEE Conference on Decision and Contro*l, Brighton, UK, Dec., 1991.

P. Kroha, P. Fritzson: A Compiler with Scheduling for a Specialized Synchronous Multiprocessor System. In *Proc. of the 3rd International Workshop on Compiler Compilers,* Schwerin, GDR, Oct. 22–26 1990. Proc in *LNCS* by Springer-Verlag, 1991.

P. Kroha, P Fritzson: Software Features of an Extended Single Instruction Machine. In *Proc. of the EuroMicro 91 conference*, short note session, Vienna, Sept. 1991. Proc. in a special issue of the EuroMicro Journal.

K. Kuchcinski, Z. Peng, B. Fjellborg: Parallelism Exploration for Automatic Design of VLSI Systems. *1st Nordic Workshop on Parallel Algorithms and Architectures in Vision and Image Processing*, Linköping, March 7–9, 1990.

K. Kuchcinski, Z. Peng: Towards Design for Testability in VLSI High-Level Synthesis Systems, *NORSILC-2/EKF seminar on VLSI testing*, Oslo, 19–20 March, 1990.

K. Kuchcinski: System Level Modelling and Analysis of Complex Digital Systems. In *Proc. of the 18th EUROMICRO Conference, Software and Hardware Specification and Design*, (short notes session), Paris, Sept. 14-17, 1992.

K. Kuchcinski: Toward Automatic Test Pattern Generation for VHDL Descriptions. In *Proc. of the EURO-VHDL, Second European Conference on VHDL Methods*, Sept. 8-11, 1991, Stockholm.

K. Kuchcinski, A. Postula: Comparing VHDL High-Level Synthesis Systems: Call for Benchmarks. In *Proc. of the EURO-VHDL, Second European Conference on VHDL Methods,* Sept. 8-11, 1991, Stockholm (poster session presentation).

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

P. Lambrix: Versioning of Strongly Connected Composite Objects. In *Proc. of the Golden West International Conference on Intelligent Systems,* pp 97-103, Reno, Nevada, USA, 1992.

P. Lambrix: Temporal Aspects of Composite Objects. In *Proc. of the Golden West International Conference on Intelligent Systems*, pp 116-125, Reno, Nevada, USA, 1992.

M. Larsson: Mixed Static and Dynamic Reasoning in a Formal System, *Sixteenth Symposium on Microprocessing and Microprogramming, EUROMICRO’90*, Short Notes Session, Amsterdam, Aug. 27–30, 1990.

T. Larsson: The Symbolic Manipulation of Event Traces. To appear in *Proc. of the ACM/IFIP 1991 International Workshop on Formal Methods in VLSI Design*, Miami, 9–11 Jan, 1991.

T. Larsson, A. Vainio-Larsson: Software Producers as Software Users. In *NATO-ARW*: *User-Centered Requirements for Software Engineering Environments*, Chateau de Bonas, Toulouse, France, Sept. 5 - 10, 1991.

G. Lindström, J. Maluszynski, T. Ogi: Our LIPS Are Sealed: Interfacing Functional and Logic Programming Systems. In *Maurice Bruynooghe and Martin Wirsing*, editors, *Proc. PLILP’92*, pages 428-442. Springer-Verlag, 1992. LNCS 631.

W. Litwin, T. Risch: Main Memory Oriented Optimization of OO Queries using Typed Datalog with Foreign Predicates. In special section on *Main Memory Databases in IEEE Transactions on Knowledge and Data Engineering,* vol.4, No. 6, Dec., 1992. Also LiTH-IDA-R-92-24.

P. Loborg, M. Sköld, A. Törne, P. Holmbom: A Model for the Execution of Task Level Specifications for Intelligent and Flexible Manufacturing Systems, LiTH-IDA-R-92-22. In *Proc. of the Fifth International Symposium on Artificial Intelligence, ISAI92,* Cancun, Mexico, Dec., 1992.

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

J. Löwgren: The Ignatius Environment. Supporting the Design and Development of Expert-System User Interfaces. *IEEE Expert,* 7(4):49–57. 1992.

J. Löwgren, T. Nordqvist: Knowledge-Based Evaluation as Design Support for Graphical User Interfaces. In *Human Factors In Computing Systems (CHI'92 Proc.),* pages 181–188, Monterey. ACM Press. 1992.

J. Malec: Dynamic Scene Description Systems – ODYS, In *Proc. of the 11th IFAC World Congress,* Tallin, Estonia, Aug. 13–17, 1990.

J. Malec: Complex Behavior Specification for Autonomous Systems. In *Proc. IEEE ISIC-92,* Glasgow, Scotland, Aug. 11-13, 1992. LITH-IDA-R-92-14.

J. Malec: Applied Knowledge Representation. In *CC-AI: The Journal for the Integrated Study of Artificial Intelligence, Cognitive Science and Applied Epistemology,* Vol. 9, No. 1, pp 9-41, 1992.

J. Malec: Process Transition Networks: What are they for? In *Proc. the IEEE International Conference on Systems, Man and Cybernetics*, Charlottesville, VA, 13-16 Oct.

J. Maluszynski: Attribute Grammars and Logic Programs: A Comparison of Concepts. In *Attribute Grammars, Applications and Systems*, Prague Summer School, H Alblas and B. Melichar (eds), pp 356-373, Lecture Notes in Computer Science 545, Springer-Verlag, 1991.

J. Maluszynski, S Bonnier, J Boye, A Kågedal, F Kluzniak, U Nilsson: Logic Programs with External Procedures. Accepted for publication in the book: *Current trends in logic programming languages, integration with functions, constraints and objects,* K.R. Apt, J. W. de Bakker, J. J. M. M. Rutten (eds.), The MIT Press. To appear 1993.

S. Moen: Drawing Dynamic Trees. In *IEEE Software, vol. 7, No 4*, July 1990.

M. Morin, E. Sandewall: A Software Architecture supporting Integration of Sub-systems. In *Prometheus: Proc. of the fourth Workshop*, 1990.

M. Morin, S. Nadjm-Tehrani, P. Österling, E. Sandewall: Real-time Hierarchical Control. In *IEEE Software special issue on real-time systems development*, Sept., 1992.

S. Nadjm-Tehrani: Debugging Prolog Programs Declaratively. In *M. Bruynooghe, editor, Proc. of the Second Workshop on Meta-Programming in Logic*, pages 137–155, April 1990.

S. Nadjm-Tehrani: Driver interactions with a signal support system. In *Prometheus: Proc of the Third Workshop,* 1990.

S. Nadjm-Tehrani, P Österling: Characterization of Environment Conditions with Metric Temporal Feature Logic. In *Proc. of* the *3rd IEEE International Conference on AI, Simulation and Planning in High Autonomy Systems,* Perth, Australia, July, 1992.

T. Nagle, N. Nagle, L. Gerholz, P. Eklund (editors): *Conceptual Structures: Current Research and Practice*. Ellis-Horwood Limited. 1992.

T. Näslund: Injective Formative Evaluation for Enhancing Computer Support for Cooperative Staff Work. In *Proc. of the Human Interface '92*, Kawasaki. In press. 1992.

B. Nebel, C. Bäckström: On the Computational Complexity of Temporal Projection and Plan Validation. In *Proc. of AAAI-92,* pp 748 - 753, San Jose, Ca, July, 1992.

B. Nebel, C Bäckström: On the Computational Complexity of Temporal Projection, Planning and Plan Validation. Accepted for publication in *the journal Aritifical Intelligence.* To appear.

H. Nilsson, P. Fritzson: Algorithmic Debugging for Lazy Functional Languages. In *Proc* of *PLILP'92 - Symposium on Programming Language Implementation and Logic Programming,* Leuven, Belgium, Aug., 1992. LNCS 631, Springer Verlag.

U. Nilsson: Systematic Semantic Approximations of Logic Programs. In *P. Deransart and J. Maluszynski, editors, Programming Language Implementation and Logic Programming, International Workshop PLILP '90,* pages 293–306. Springer-Verlag LNCS 456, 1990.

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

J. Nyce, T. Timpka: Work, Knowledge and Argument in Specialist Consultations: Incorporating Tacit Knowledge into System Design and Development. *Med Biol Eng Comp,* 1993. In press.

L. Padgham: Defeasible Inheritance: A Lattice Based Approach, *Computers and Mathematics with Applications*, special issue on Semantic Nets.1991.

L. Padgham: Defeasible Inheritance: A Lattice Based Approach, Computers and Mathematics with Applications, special issue on *Semantic Nets*, vol. 23, No. 6-9, pp 527 - 541, 1992. Also in *Semantic Networks in Artificial Intelligence*, pp 527 - 542, F Lehman (ed), Pergamon Press, 1992.

L. Padgham, J Löwgren: User Interface Management for an Object-Oriented Database System. In *Proc. of the ISMM First Int. Conf. on Information and Knowledge Management*, pp 300 - 309, 1992.

L. Padgham, B. Nebel: Combining Classification and Non-Monotonic Inheritance Reasoning: A First Step. In *Proc. of Workshop in Description Logics, AAAI Fall Symposium*, Boston, MA, Oct., 1992.

L Padgham: Negative Reasoning Using Inheritance. In *Inheritance Networks for Artificial Intelligence Intellect*, Raad Al-Asady and Ajct Narayanan (ed.).

M.R.K Patel: A Design Representation for High Level Synthesis, *Proc. European Design Automation Conference (EDAC'90)*, Glasgow, Scotland, 12–15 March 1990.

M.R.K Patel: Random Logic Circuit Implementation of Extended Timed Petri Nets, *Microprocessing and Microprogramming, The Euromicro Journal*, vol. 30, no. 1–5, Aug. 1990, pp. 313–319.

Z. Peng: Design of Clocking Schemes in High-Level Synthesis, *Sixteenth Symposium on Microprocessing and Microprogramming, EUROMICRO'90*, Short Notes Session, Amsterdam, Aug. 27–30, 1990.

Z. Peng, J Fagerström and K Kuchcinski: A Unified Approach to Evaluation and Design of Hardware/Software Systems. In *Proc. Workshop on Software/Hardware CoDesign, 13th International Conference on Software Engineering*, Austin, Texas, May 13, 1991.

Z. Peng: Digital System Simulation with VHDL in a High-Level Synthesis System. In *Proc. 18th Euromicro Conference on Software and Hardware: Specification and Design*, Paris, Sept., 14-17, 1992.

Z. Peng and K Kuchcinski: An Algorithm for Partitioning of Application Specific Systems. In *Proc. of the European Conference on Design Automation, EDAC-93*, Paris, Feb. 22 -25, 1993.

J. Persson, E. Jungert: Generation of multi-resolution maps from run-length-encoded data. Accepted for publication in *International Journal of Geographical Information Systems*.

T. Persson, L. Staflin: A causation theory for a logic of continuous change. In *Proc. European Conf. on AI. (ECAI)*, 1990.

M. Pettersson: Generating Efficient Code from Continuation Semantics. In *Proc. of the 3rd International Workshop on Compiler Compilers (CC'90)*, Schwerin, GDR, Oct. 22–26 1990. In LNCS-477, Springer-Verlag, 1991.

M. Pettersson, P. Fritzson: A General and Practical Approach to Concrete Syntax Objects within ML. In *Proc. of the ACM SIGPLAN Workshop on ML and its Applications*, San Francisco, CA, June, 20-21, 1992.

M. Pettersson, P. Fritzson,: DML - A Meta-language and System for the Generation of Practical and Efficient Compilers from Denotational Specifications. In *Proc. of the 1992 International Conference on Computer Languages*, Oakland, California, April 20 - 23, 1992.

M. Pettersson: A Term Pattern-Match Compiler Inspired by Finite Automata Theory. In *Proc. of the International Workshop on Compiler Construction (CC'92)*, Paderborn, Germany, Oct., 5-7, 1992. LNCS 641, Springer Verlag.

H. Przymusinska, T. C. Przymusinski: Stationary Extensions of Default Theories. In *Proc. of the Fourth Workshop on Non-monotonic Reasoning*, Plymouth, Vermont, 1992, (In print.).

H. Przymusinska, T. C. Przymusinski, H. Seki: Soundness and Completeness of Partial Deductions for Well-founded Semantics. In *A. Voronkov, editor, Logic Programming and Automated Reasoning*, St. Petersburg, Russia, July 1992. (*Lecture Notes in Artificial Intelligence, Vol.624*), pages 1-12. Springer-Verlag, 1992.

T. C. Przymusinski: Autoepistemic Logics of Closed Beliefs and Logic Programming. In *A. Nerode, W. Marek, and V. S. Subrahmanian, editors, Proc. of the First International Workshop on Logic Programming an Non-monotonic Reasoning*, Washington, D.C., July 1991, pages 3-20, Cambridge, Mass., 1991. The MIT Press.

T. C. Przymusinski: Well-founded Completions of Logic Programs. In *Proc. of the Eight International Logic Programming Conference*, Paris, France, pages 726-744, Cambridge, Mass., 1991. Association for Logic Programming, The MIT Press.

T. C. Przymusinski: Two Simple Characterizations of Well-founded Semantics. In I. Havel and V. Koubek, editors, *Proc. of the International Conference on Mathematical Foundations of Computer Science (MFCS'92)*, pages 451-462, Prague, Czechoslovakia, 1992. Springer-Verlag.

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

I. Rankin, J. Harrius: Rethorical Aggregates in Text Planning. In *Proc. of the Third Nordic Conference on Text Comprehension in Man and Machine*, Linköping, Sweden. In press. 1992.

B. Rapp: Management Training in Small Companies: a Case Studie. Paper presented at Berch Symposium, Uppsala, Sweden, 1991.

J. Ringström, P. Fritzson, J. Fagerström: PREDULA - A Multi-Paradigm Parallel Programming and Debugging Environment. In *Proc. of the EuroMicro 91 conference*. Short note session, Vienna, Sept. 1991. Proc. in a special issue of the EuroMicro Journal.

T Risch, M Sköld: Active Rules based on Object-Oriented Queries, LiTH-IDA-R-92-35. In *Special Issue on Active Databases of IEEE Data Engineering*, Jan., 1993.

R. Rönnquist: A Logic for Propagation Based Characterisation of Process Behaviour. In *Proc.* of the *International Symposium on Methodologies for Intelligent Systems (ISMIS*), Oct. 24–27, 1990.

K. Sandahl, H. Eriksson, T. Padron-McCarthy, T. Sokolnicki, B. Österlund: Meeting the Requirements of Knowledge Management. In *Expert Systems with Applications*, 3:259-267, 1991.

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

E. Sandewall: LISP as a Very High Level Implementation Language. In *Proc. First European Conf. on the Practical Applications of LISP (EuroPAL),* 1990.

E. Sandewall: Towards a logic of dynamic frames. In Ken Ford and Patrick Hayes, editors, *The Frame Problem in Artificial Intelligence,* JAI Press, 1990

E. Sandewall: Reasoning about the world as perceived by an agent. In *Proc. European Conf. on A.I.*(ECAI), 1990.

E. Sandewall, Hamm, de Charentenay, Höfflinger, Reister, E. Sandewall, Williams: Improved driver information and driver assistance by PROMETHEUS Systems. In *Proc. FISITA Congress*, Torino, 1990.

E. Sandewall: Knowledge-based systems, Lisp, and very high level implementation languages. In *The Knowledge Engineering Review,* Vol. 7:2, 1992, pp 147-155.

E. Sandewall: Causal Qualification and Structure-Based Ramification. Presented at *the Symposium on Commonsense Reasoning,* Austin, Texas, Jan., 1993.

B. Savén: Factory Simulation - Concepts and one Approach, *Robotics & Computer-Integrated Manufacturing,* Vol. 9, No 4/5, pp 341-345, 1992.

N. Shahmehri, P. Fritzson: Algorithmic Debugging for Imperative Languages with Side-effects. In P*roc. of the 3rd International Workshop on Compiler Compilers*, Schwerin, Germany, Oct. 22–24, 1990. In LNCS, Springer-Verlag.

N. Shahmehri, M. Kamkar, P. Fritzson: Semi-Automatic Bug Localization in Software Maintenance. In *Proc of* the *IEEE Conf. on Software Maintenance,* San Diego, Nov.,1990.

H. Shu: Planning with reactivity. In *Proc. Pacific Rim International Conf. on Artificial Intelligence,* 1990.

H. Shu: The Architecture of a Reactive Car Support System. In *Proc. of the 4th PROMETHEUS workshop*, Compiègue, Oct. 1990.

H. Shu: Hierarchical planning using action taxonomies. In Z. Ras, editor, *Methodologies for Intelligent Systems, V.* 1990.

H. Shu, J. Malec: From Process Transition Networks to Behaviour Automata: Presented at *ISIC-91*, Arlington, VA, Aug. 13 -15, 1991.

H. Shu: A Preferential Logic for Reasoning with Goals and Intentions. Presented at *the Logic and Change Workshop at GWAI'92*, Bonn, Germany, Sept. 3, 1992.

Y. Shu: Integrating Terminological Reasoning into an Intelligent Information System. In *Proc. of PRICAI'92*. Seoul, Korea, Sept., 1992.

Y. Shu: Introducing CLASSIC-like Functions in LINCKS System. In *Terminological Logic Users Workshop Proc*., Berlin, Dec., 1991.

C. Sjöberg, T. Timpka, J.M. Nyce, M. Peolsson, T. af Klercker: From clinical literature to medical hypermedia: procedures and experiences. *Proc. of MedInfo '92 (World Conference of Medical Informatics)*. Amsterdam: Elsevier, 1992 In press.

T. Sokolnicki: Towards Knowledge-Based Tutors: A Survey and Appraisal of Intelligent Tutoring Systems. *Journal Knowledge Engineering Review*, 1991, vol. 6, No 2, pp 59 -95.

L. Strömberg, P. Fritzson, J. Fagerström: Postmortem Debugging of Parallel Real Time Programs. In *Proc. of the 1st National Swedish Symposium on Real-Time Systems*, Uppsala, Aug. 19-20, 1991.

J-E. Strömberg, J. Top, U. Söderman: Variable Causality in Bond Graphs Caused by Discrete Effects. Accepted to *the International Conference on Bond Graph Modeling and Simulation ICBGM'93,* San Diego, USA, Jan., 1993.

J-E. Strömberg, U. Söderman, J. Top.: Bond Graph Supported Estimation of Discretely Changing Parameters. Accepted to *International Conference on Bond Graph Modeling and Simulation ICBGM'93*, San Diego, USA, Jan., 1993.

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

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

T. Timpka, P. Hedblom, H. Holmgren: Using an object-oriented environment for group process development of medical software. *Proc. of Software Engineering in Medical Informatics '90*, Rotterdam, 1990.

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

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

T. Timpka, T. Buur, P. Hedblom: Is expertext support feasible in work life applications? An empirical study using an expert panel for simulating "optimal" expertext support. *Information and Decision Technologies*, 1992;18:81-87.

T. Timpka, J. M. Nyce: Dilemmas at a primary health care centre: a baseline study for Computer Supported Cooperative Work (CSCHW). *Meth Inf Med*, 1992;31:204-9.

T. Timpka, J. M. Nyce: Towards a pragmatics of medical hypertext. In *Proc. of MedInfo '92 (World Conference of Medical Informatics)*. Amsterdam: Elsevier, 1992. In press.

T. Timpka, J. M. Nyce, C. Sjöberg, P. Hedblom, P. Lindblom: Developing a clinical hypermedia corpus: Experiences from a practice-centered method. In *Proc. of SCAMC '92*. Baltimore: IEEE Press, 1992. In press.

A. Törne: The Instruction and Control of Multi-Machine Environments. In *Proc. Applications of Artificial Intelligence in Engineering V*, Springer-Verlag, Boston, 1990.

R. Ubar, K. Kuchcinski, Z. Peng: Test Generation for Digital Systems at Functional Level. In *Proc. European Test Conference*, Munich, April 10–12, 1991.

R. Ubar, K Kuchcinski: Functional Level Controllability Analysis for Digital Circuits. In *Proc Design Automation Conference*, June 1-4, 1992, Kaunas, Lituania.

A. Vainio-Larsson, R. Orring: Evaluating the Usability of User Interfaces: Research in Practice, Nov. 1989, 18p. Presented at INTERACT 90, The 3rd IFIP Conference on Human-Computer Interaction, Cambridge, England, Aug. 1990.

A. Vainio-Larsson: As We May Learn: Training with Multimedia. In *Proc. of the IFIP Conference on Human Jobs and Computer Interfaces*, Tampere, Finland, June 24- 28, 1991.

A. Vainio-Larsson: Training within an Interactive Multimedia Environment. In *People and Computers, HCI 91, Usability Now,* Edinburgh, Scotland, Aug. 20 - 23, 1991, pp.

L. Viklund, J. Herber, P. Fritzson: The Implementation of ObjectMath - A High-Level Environment for Scientific Computing. In *Proc. of CC'92 - International Workshop on Compiler Construction*, Paderborn, Germany, Oct., 1992. LNCS 641, Springer Verlag.

L. Viklund, P. Fritzson: An Object-Oriented Language for Symbolic Computation - Applied to Machine Elements Analysis. In *Proc of ISSAC'92 - International Symposium on Symbolic and Algebraic Computation*, Berkeley, July, 1992.

J. E. Villegas, M Sjölin: Industrial training with Simulation Games. In *Proceedings of the European Simulation Symposium ESS92*, pp. 555.Dresden, Nov., 5-8, 1992.

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

Y. Wærn, S. Hägglund, J Löwgren, I. Rankin, T. Sokolnicki, A. Steinemann: Communication Knowledge for Knowledge Communication. In *Int. Journal of Man-Machine Studies*, vol. 37, No 2, pp 215 - 239, 1992.

M. Wedlin: Interface between UNIX and Hypercard. Accepted for the EUUG Spring Conference 90, Munich, FRG, April 23–27, 1990.

M. Wirén: Incremental Parsing and Reason Maintenance. In *Papers presented to the 13th International Conference on Computational Linguistics,* Helsinki 20–25 Aug., 1990, Vol. 3, pp. 287–293.

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

T. Zhang, L Padgham: Diagnosis Using a Default Inheritance Hierarchy. In *Proc. of the IJCAI Workshop on Knowledge Representation in Medicine*, Sydney, 1991. Also accepted for publication in *AI in medicine*. Sydney, 1991.

E.6 Departmental Reports 1991–92

LiTH-IDA-R-91-01 **Staffan Bonnier, Ulf Nilsson, Torbjörn Näslund:** A Simple Fixed Point Characterization of Three-valued Stable Model Semantics. Also accepted for publication *in Processing Letters.*

LiTH-IDA-R-91-02 **Ulf Nilsson:** Abstract Interpretation: A Kind of Magic. Also accepted to the *3rd International Symposium on Programming Language Implementation and logic Programming (PLILP),* Passau, Germany, Aug. 26-28, 1991.

LiTH-IDA-R-91-03 **Arne Jönsson:** A Dialogue Manager Using Initiative-Response Units and Distributed Control. Also presented at the *5th Conference of the European Chapter of the ACL,* Berlin, Germany, April 9-11, 1991.

LiTH-IDA-R-91-04 **Peter Fritzson, Dag Fritzson:** The Need for High-Level Programming Support in Scientific Computing Applied to Mechanical Analysis.

LiTH-IDA-R-91-05 **Dag Fritzson, Peter Fritzson:** Equational Modeling of Machine Elements – Applied to Rolling Bearings.

LiTH-IDA-R-91-06 **Peter Fritzson:** ObjectMath - Object Oriented Mathematical Modeling in Scientific Computing, Applied to Machine Elements Analysis

LiTH-IDA-R-91-07 **Simin Nadjm-Tehrani:** Analysis of the Overtaking Scenario: Specification of an Autonomous Car and a Driver Support System.

LiTH-IDA-R-91-08 **Jacek Malec:** How To Pass an Intersection, or Automata Theory is Still Useful To be presented at *SCAI'91*, Roskilde, Denmark, May 21-24, 1991.

LiTH-IDA-R-91-09 **Peter Loborg, Anders Törne:** A Hybrid Language for the Control of Multimachine Environments. Accepted to the *4th Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems,* Hawaii, June 2-5, 1991.

LiTH-IDA-R-91-10 **Arne Jönsson:** A Natural Language Shell and Tools for Customizing the Dialogue in Natural Language Interfaces.

LiTH-IDA-R-91-11 **Christer Bäckström, Inger Klein:** Parallel Non-Binary Planning in Polynomial Time: The SAS-PUS Class. A short version is accepted to the *12th International Joint Conference on Artificial Intelligence (IJCAI'91).* Sydney, Australia, Aug. 24-30, 1991.

LiTH-IDA-R-91-12 **Andreas Kågedal, Feliks Kluzniak:** Enriching Prolog With S-unification. Also in *"Phoenix" Seminar on Declarative Programming*, Hohritt, Germany, Nov. 1991, Springer-Verlag, LNCS.

LiTH-IDA-R-91-13 **Patrick Doherty,Withold Lukaszewicz:** NML3 — An Non-monotonic Logic with Explicit Defaults. To appear in the *Journal of Applied Non-Classical Logics, 1991.*

LiTH-IDA-R-91-14 **Patrick Lambrix:** Temporal Aspects of Composite Objects. Also accepted to the *Golden West International Conference on Intelligent Systems,* Reno Nevada 1992

LiTH-IDA-R-91-15 **Ulf Söderman, Jan-Erik Strömberg:** Combining Qualitative and Quantitative Knowledge to Generate Models of Physical Systems. A Short version is accepted to the *12th International Joint Conference on Artificial Intelligence (IJCAI-91)* Sydney, Australia, Aug. 24-30, 1991.

LiTH-IDA-R-91-16 **Zebo Peng, Johan Fagerström, Krzysztof Kuchcinski:** A Unified Approach to Evaluation and Design of Hardware/Software Systems. Also accepted to the *ICSE (13th International Conference on Software Engineering) Workshop on Software/hardware Codesign,* Austin, Texas, May 13, 1991.

LiTH-IDA-R-91-17 **Xinli Gu, Krzysztof Kuchcinski, Zebo Peng:** Testability Measure with Reconvergent Fanout Analysis and Its Applications. This paper will be published in *Microprocessing and Microprogramming the EUROMICRO Journal, Vol. 33, 1991.*

LiTH-IDA-R-91-18 **Johan Boye:** S-SLD-resolution — An Operational Semantics for Logic Programs with External Procedures. Also accepted to the *3rd International Symposium on Programming Language Implementation and Logic Programming,* Passau, Germany, Aug. 26-28, 1991.

LiTH-IDA-R-91-19 **Mariam Kamkar:** An Overview and Comparative Classification of Static and Dynamic Program Slicing.

LiTH-IDA-R-91-20 **Mariam Kamkar:** Interprocedural Dynamic Slicing Applied to Algorithmic Debugging.

LiTH-IDA-R-91-21 **Lars Ahrenberg, Arne Jönsson, Nils Dahlbäck:** Discourse Representation and Discourse Management for a Natural Language Dialogue System**.** Also in Proc. of *The Second Nordic Conference on Text Comprehension in Man and Machine (NOTEX'90)*, Täby-Stockholm, Sweden, 1990.

LiTH-IDA-R-91-22 **Teodor C. Przymusinski:** Stable Semantics for Disjunctive Programs. This paper will appear in the special issue of the *Journal of New Generation Computing.*

LiTH-IDA-R-91-23 **Wlodzimierz Drabent:** Constructive Negation by Fail Answers. Also published in the *Journal Acta Informatica*, 1992.

LiTH-IDA-R-91-24 **Per-Olof Fjällström:** Polyhedral Approximation of Bivariate Functions. Also in Proc. of *Third Canadian Conference on Computational Geometry*, Vancouver, B.C., Canada, Aug. 5-10, 1991.

LiTH-IDA-R-91-25 **Per-Olof Fjällström, Jyrki Katajainen, Jan Petersson**: Algorithms for the All-Nearest-Neighbors Problem. Also in *Proc. of Conf. on System Modelling and Optimization*, International Federation for Information Processing (IFIP), Zurich, Switzerland, Sept. 2-6, 1991.

LiTH-IDA-R-91-26 **Feliks Kluźniak:** Towards Practical Executable Specifications in Logic.

LiTH-IDA-R-91-27 **Feliks Kluźniak:** PageBreak: An Exercise in Specification and Derivation.

LiTH-IDA-R-91-28 **Feliks Kluźniak:** SPILL: A Specification Language Based on Logic Programming.

LiTH-IDA-R-91-29 **Erik Sandewall:** Features and Fluents. An agenda and logic-based framework for the Representation of Knowledge about Dynamical Systems. Review version of Chapters 1 and 3-7. *This is a preliminary version of parts of a forthcoming book.*

LiTH-IDA-R-91-31 **Nils Dahlbäck:** Empirical Analysis of a Discourse Model for Natural Language Interfaces. Also in *Proc. of the 13th Annual Conference of the Cognitive Science Society (COG SCI -91)*, Aug. 7-10, 1991, Chicago, Illinois, USA.

LiTH-IDA-R-91-32 **Teodor C Przymusinski:** Semantics of Disjunctive Logic Programs and Deductive Database.Accepted to *The Second International Conference on Deductive and Object-Oriented Databases*, Munich, Germany, Dec., 1991.

LiTH-IDA-R-91-33 **Laurent Fribourg:** A Decision Procedure for a Subtheory of Linear Arithmetic with Lists.

LiTH-IDA-R-91-34 **Bernhard Nebel, Christer Bäckström:** On the Computational Complexity of Temporal Projection and some Related Problems. Also published as *DFKI Research Report RR-91-34, German Research Center for Artificial Intelligence (DFKI),* Saarbrucken, Germany. The results in this report has been divided into conference papers that will appear in proceedings of the *AAAI-92, conference,* San Jose, Ca USA, Jul. 12-17, 1992 and in proceedings of the *ECAI-92 conference,* Vienna, Austria, Aug. 3-7, 1992 respectively and has been accepted for publication in Artificial Intelligence (ie.AI Journal)

LiTH-IDA-R-91-35 **Lena Strömbäck:** Unifying Disjunctive Feature Structures.

LiTH-IDA-R-91-36 **Krzysztof Kuchcinski:** Toward Automatic Tets Pattern Generation for VHDL Description. Also in Proceedings of the *EURO-VHDL 91*, Sweden, Stockholm, Sept. 8-11, 1991.

LiTH-IDA-R-91-37 **Christer Bäckström, Inger Klein:** On the Planning Problem in Sequential Control. This report also appears in the proceedings of the *30th IEEE Conference on Decision and Control*, 11-13 Dec. 1991 Brighton, England

LiTH-IDA-R-91-38 **Peter Fritzson, Dag Fritzson, Lars Viklund, Johan Herber:** Transformation of Equation-Based Real-World Models to Efficient Code, Applied to Machine Elements Geometry. Also in Proceedings of the *1st National Swedish Symposium on Real-Time Systems*, Uppsala, Aug. 19-20, 1991.

LiTH-IDA-R-92-01 **Torbjörn Näslund:** Evaluation as a Means for Improving Development of Decision of Support Systems. Presented at *Doctoral Consortium, 11th International Conference on Information Systems,* Copenhagen, Denmark, Dec., 17-19, 1990

LiTH-IDA-R-92-02 **Patrick Doherty:** A Constraint-Based Approach to Proof Procedures for Multi-Valued Logics.Also published in *1st World Conference on the Fundamentals of Artificial Intelligence*, Paris, France, 1-5 July 1991

LiTH-IDA-R-92-03 **Raimund Ubar.** Functional Level Testability Analysis for Digital Circuits.

LiTH-IDA-R-92-04 **Petru Eles, Krzysztof Kuchcinski, Zebo Peng, Marius Minea:** Compiling VHDL into a High-Level Synthesis Design Representation. Also has been accepted for presentation at the *EURO-DAC (EURO- Design Automation Conference),* Hamburg, Germany, Sept. 7-10, 1992. Was selected as the Best Paper of the *EURO-DAC/VHDL* conference in Hamburg, Germany

LiTH-IDA-R-92-05 **Christer Bäckström:** Planning with Partical States in $O(n^2)$ Time: The SAS$^+$-PUS Planning Problem.

LiTH-IDA-R-92-06 **Niclas Andersson, Peter Fritzson:** Comparative Evaluation and Industrial Application of Code Generator Generators

LiTH-IDA-R-92-07 **Jacek Malec:** Process Transition Networks: The Final Report.

LiTH-IDA-R-92-08 **Tony I Larsson, Arja A Vainio-Larsson:** Software Producers as Software Users. Presented at *NATO Workshop on Software Requirement for Software Environments,* Toulouse, France, 5-11 Sept. 1991.

LiTH-IDA-R-92-09 **Mikael Pettersson:** A Term Pattern-Match Compiler Inspired by Finite Automata Theory. Also accepted for presentation at the *International Workshop on Compiler Construction (CC'92),* Paderborn, Germany, Oct. 5-7, 1992.

LiTH-IDA-R-92-11 **Patrick Lambrix:** Management of Historical Information of Composite Objects, 1992. (**Master Thesis No 1**)

LiTH-IDA-R-92-13 **Douglas Busch:** Sequent Formalizations of Three-valued Logic. Accepted: revised version of *"An Intuitionistic Three-valued Logic"* delivered at the Logic Programming in AI First Compulog-Net Workshop, 23-24 March 1992, Imperial College London. Is going to be presented at the *Nordic Workshop and Summer School on Partial Semantics and Non-Monotonic Reasoning for Knowledge Representation,* May 25-29 at Linköping.

LiTH-IDA-R-92-14 **Jacek Malec:** Complex Behavior Specification for Autonomous Systems, 1992. Accepted to *IEEE International Symposium on Intelligent Control'92,* Glasgow, Scotland, Aug. 11-13, 1992.

LiTH-IDA-R-92-15 **Johan Fagerström**, **Peter Fritzson, Johan Ringström, Mikael Pettersson:** A Data-Parallel Language and its Compilation to a Formally Defined Intermediate Language. In Proceedings of *4th International Conference on Computing and Information, ICCI'92,* Toronto may 28-30, 1992.

LiTH-IDA-R-92-16 **Mikhail Auguston, Peter Fritzson:** Parforman - an Assertion Language for Specifying Behaviour when Debugging Parallel Applications. Has been accepted to the *EuroMicro workshop on Parallel and Distributed Processing,* Jan., 1993.

LiTH-IDA-R-92-17 **Henrik Nilsson, Peter Fritzson:** Algorithmic Debugging for Lazy Functional Languages. In Proceedings of *PLILP'92, Symposium on Programming Language Implementation and Logic Programming, LNCS,* Springer Verlag, Leuven, Belgium, Aug. 26-28, 1992.

LiTH-IDA-R-92-18 **Patrick Doherty, Dimiter Driankov, Alexis Tsoukias:** Partiality, Para-consistency and Preference Modeling: Preliminary Version.

LiTH-IDA-R-92-19 **Nils Dahlbäck, Arne Jönsson, Lars Ahrenberg:** Wizard of Oz-studies - why and how. Has been accepted to the *International Workshop on Intelligent User Interfaces,* Orlando, Florida, Jan., 4-7, 1993.

LiTH-IDA-R-92-20 **Patrick Doherty, Witold Lukaszewicz**: FONML3 - A First-Order Non-monotonic Logic with Explicit Defaults This is an extended version of a paper published in *the Proceedings of the 10th European Conference on Artificial Intelligence (ECAI 92),* Vienna, Austria, Aug. 3-7, 1992.

LiTH-IDA-R-92-21 **Patrick Doherty. Dimiter Driankov, Hans Hellendoorn:** Fuzzy If-Then-Unless Rules and their Implementation. Extended Version of a Paper accepted at *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'92* Palma de Mallorca, July 6-10, 1992.

LiTH-IDA-R-92-22 **Peter Loborg, Per Holmbom, Martin Sköld, Anders Törne:** A Model for the Execution of Task Level Specifications for Intelligent and Flexible Manufacturing Systems. Accepted at the *V International Symposium on Artificial Intelligence, ISAI92,* Cancun, Mexico, Dec. 7-11, 1992.

LiTH-IDA-R-92-23 **Gustav Fahl:** Integration of Heterogeneous Databases in a Mediator Architecture. Also in the Proceedings of *the Third Annual IDA Conference on Computer and Information Science* Linköping University, Sweden, March 1992.

LiTH-IDA-R-92-24 **Witold Litwin, Tore Risch:** Main Memory Oriented Optimization of OO Queries using Typed Datalog with Foreign Predicates. Accepted for publication in *IEEE Transactions on Knowledge and Data Engineering* and accepted to *IEEE Transactions on Knowledge and Data Engineering* in a special section on main memory databases.

LiTH-IDA-R-92-25 **C Sjöberg, T Timpka, JM Nyce, M Peolsson, T af Klercker:** From Clinical Literature to Medical Hypermedia: Procedures and Experiences. Also Accepted to *MEDINFO'92*, Geneva, Switzerland, Sept. 1992.

LiTH-IDA-R-92-26 **Toomas Timpka, James M Nyce:** Towards a Pragmatics for Medical Hypermedia Systems. Accepted to the *MEDINFO'92* Geneva, Switzerland, Sept. 1992.

LiTH-IDA-R-92-27 **T Timpka, JM Nyce, C Sjöberg, P Hedblom, P Lindblom:** Developing a Clinical Hypermedia Corpus: Experiences from the use of a Practice-centered Method. Accepted to *SCAMC'92*, Baltimore, Maryland, Nov. 1992.

LiTH-IDA-R-92-28 **H Holmgren, T Timpka, G Goldkuhl, JM Nyce, C Sjöberg:** Argumentative Design of Medical Software: an Essential Part of Action Design. Accepted to the *MEDINFO'92*, Geneva, Switzerland, Sept. 1992.

LiTH-IDA-R-92-29 **Richard Hirsch:** About Belief: Representation of Content, Change and Development.

LiTH-IDA-R-92-30 **Erik Sandewall:** Features and Fluents. A Systematic Approach to the Representation of Knowledge about Dynamical Systems.

LiTH-IDA-R-92-31 **Magnus Merkel:** Recurrent Patterns in Technical Documentation.

LiTH-IDA-R-92-32 **Nils Dahlbäck, Arne Jönsson:** An Empirically Based Computationally Tractable Dialogue Model. Also in the proceedings of the *14th Annual Conference of the Cognitive Science Society (COG SCI-92)*, Bloomington, Indiana, July 29th to Aug. 1st.

LiTH-IDA-R-92-33 **Jonas Persson:** Generation of Multi-Resolution Maps. Awarded the prize *"Lilla Polhemspriset*" 1992.

LiTH-IDA-R-92-34 **Hua Shu:** A Preferential Logic for Reasoning with Goals and Intentions. Accepted to *LOGIC & CHANGE workshop at GWAI'92*, Bonn Sept. 3th 1992.

LiTH-IDA-R-92-35 **Tore Risch, Martin Sköld:** Active Rules based on Object-Oriented Queries. To be published in special issue on *Active Databases in IEEE* Data Engineering.

LiTH-IDA-R-92-36 **Jan-Erik Strömberg, Jan Top, Ulf Söderman:** Variable Causality in Bond Graphs Caused by Discrete Effects. Accepted to *1993 International Conference on Bond Graph Modeling and Simulation ICBGM 93*, San Diego, USA, Jan., 1993.

LiTH-IDA-R-92-37 **Jan-Erik Strömberg**, **Ulf Söderman, Jan Top:** Bond Graph Supported Estimation of Discretely Changing Parameters. Accepted to *1992 International Conference on Bond Graph Modeling and Simulation ICBGM 93*, San Diego, USA, Jan., 1993.

LiTH-IDA-R-92-38 **Jan-Erik Strömberg, Jan Top, Ulf Söderman:** Modelling Mode Switching in Dynamical Systems.

LiTH-IDA-R-92-39 **Göran Goldkuhl, Annie Röstlinger:** Att bygga in verksamhetskvalitet i informationssystem. Accepterad till *Konferens "Sundsvall 42"*, Dataföreningen, Sundsvall, Oct. 15-17, 1991.

LiTH-IDA-R-92-40 **Göran Goldkuhl:** Information Systems Design as Argumentation - An Investigation into Design Rationale as a Conceptualization of Design. Accepted to the *14th IRIS (Information Systems Research Seminar in Scandinavia)*, Umeå, Aug. 11-14, 1991.

LiTH-IDA-R-92-41 **Göran Goldkuhl:** Stöd och Struktur i Systemutvecklingsprocess. Accepterad till *"Systemutveckling i praktisk belysning"* Dataföreningen, Norrköping, 19 Nov. 1991.

LiTH-IDA-R-92-42 **Erik Sandewall:** Causal Qualification and Structure-Based Ramification. Accepted to *The Second Symposium on Logical Formalizations of Commonsense Reasoning*, Austin, TX, USA, Jan. 11-13, 1993.

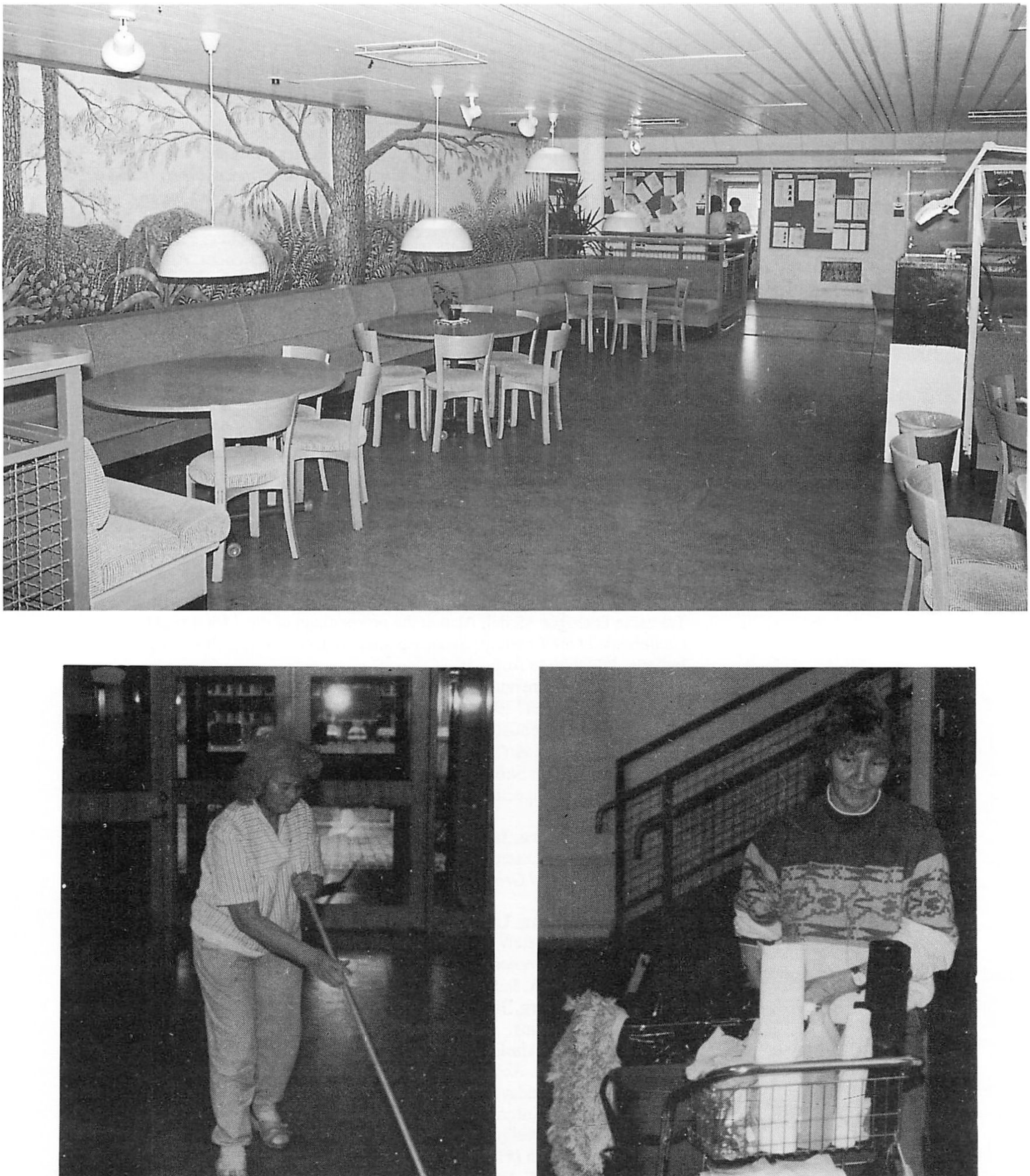


Figure E-1. *The E building of the department, and Kerstin Andersson and Maud Pettersson make sure that IDA's rooms are the cleanest rooms!*

Appendix F
Mailing Addresses, Phone and Electronic Mail

The departmental mailing address is:

Dept. of Computer and Information Science
Linköping University
S-581 83 Linköping, Sweden

The phone number to the university switchboard is +46 13 281000. Direct phone number to each staff member is given below. Telefax may be sent to the department:

Telefax +46 13 142231

Electronic mail may be sent using one of the following net addresses:

Internet *username*`@ida.liu.se`
UUCP `enea!liuida!`*username*
BITNET *username*`@seliuida`

Each person's *username* is listed in the column *E-mail* in the staff list below.

IDA staff

Below follows an alphabetically ordered list of the IDA staff members, as by December 1992, with phone numbers (including area code), room numbers and user names on the local computer network. Room numbers are composed by a building name (B, E or E++), floor, and room number (a leading B, F or G in a room number means corridor B, F or G).

Name	*Phone*	*Room no.*	*E-mail*
Ahlenbäck, Britt-Marie	013-28 1318	E, 1tr F 458	`bma`
Ahrenberg, Lars	013-28 2422	E, 1st fl., F 488	`lah`
Aleryd, Anders	013-28 1707	E, gd. fl., G 434	`aoa`
(mobile phone)	010-22 31 704		
Andersson, David	013-28 2698	E, 1st fl., 467	`dan`
Andersson, Jörgen	013-28 2537	E++, gd. fl., 106	`joran`
Andersson, Mats S	013-28 1425	E, gd. fl., G 436	`msa`
Andersson, Niclas	013-28 2601	E++, 1st fl.,	`nican`
Arendell, Pia	013-28 1426	E, 1st fl., G 489	`par`
Bennet, Thomas	013-28 2363	B, 1st fl., B 484	`thobe`
Berggrund, Tor		B, 1st fl., B 484	
Berglund, Eva-Britt	013-28 1458	E, 1st fl., G 484	`evabe`

Name	*Phone*	*Room no.*	*E-mail*
Bergman, Per	013-28 2494	E++, 1st fl., 227	perbe
Bilos, Rober	013-28 2164	E, gd. fl., F 482	rob
Björklind, Andreas	013-28 1969	E, gd. fl., F 448	abj
Björklund, Lisbeth	013-28 2420	E, gd. fl., F 452	lbj
Blom, Stefan	013-28 2603	B, 1st fl., B 482	stebl
Boye, Johan	013-28 2694	E++, 1st fl., 201	johbo
Brolin, Anna	013-28 2669	E, 1st fl., G 485	annbr
Busch, Douglas	013-28 2403	E, gd. fl., F 466	dbu
Bäckström, Anders	013-28 2859	B, 1st fl., B 490	andba
Bäckström, Christer	013-28 2429	E, gd. fl., F 488	cba
Carlshamre, Pär	013-28 2696	E, 1st fl., G 453	parca
Carlsson, Peter	013-28 2537	E++, gd. fl., 106	petca
Cheikes, Brant	013-28 1430	E, 1st fl., G 436	brant
Cronholm, Stefan	013-28 1638	E, 1st fl., F 453	scr
Dahlbäck, Nils	013-28 1664	E, 1st fl., F 472	nda
Davidsson, Maria	013-28 2859	B, 1st fl., B 490	marol
Degerstedt, Lars	013-28 2168	E, gd. fl. G 464	larde
Doherty, Patrick	013-28 2426	E, gd. fl., F 460 A	pdy
Drabent, Wlodzimierz	(on leave)		
Driankov, Dimiter	013-28 1716	E, gd. fl., F 470	ddr
Ekman, Barbara	013-28 1410	E, 1st fl., F 464	bek
Elfinger, Eva	013-28 1524	E++, gd. fl., 103	evael
Emanuelson, Inger	013-28 2180	E, 1st fl., 1:151	ime
Eriksson, Henrik			her
Eskilsson, Anne	(on leave)		
Fagerström, Johan	013-28 1364	E++, 1st fl., 214	jfa
Fahl, Gustav	013-28 2670	E++, 1st fl., 223	gusfa
Finmo, Leif	013-28 1443	E, gd. fl., G 488	lbf
Fjällström, Per-Olof	013-28 2412	E++, gd. fl., 119	pof
Forslund, Göran	013-28 2696	E, 1st fl., G 453	gorfo
Franzén, Birgitta	013-28 2692	E, 1st fl., G 446	bfr
Fritzson, Peter	013-28 1484	E++, 1st fl., 216	paf
Fäldt, Arne	013-28 1962	E, gd. fl., G 452	arf
Goldkuhl, Göran	013-28 1452	E, 1st fl., G 488	ggo
Granlund, Rego	013-28 2465	E, 1st fl., G 435	reg
Gu, Xinli	013-28 1970	E++, 1st fl., 23	xgu
Gudinge, Elisabeth	013-28 1832	B, 1st fl., B 478	eligu
Hall, Thomas	013-28 1465	E++, gd. fl., 127	thoha
Haraldsson, Anders	013-28 1403	E, 1st fl., 29:149	aha
Harrius, Jonni	013-28 2493	E, 1st fl., G 454	jha

Name	*Phone*	*Room no.*	*E-mail*
Hedblom, Per	013-28 2413	E, 1st fl., F 434	`phe`
Helander, Björn	013-28 1547	E++, gd. fl., 105	`bjohe`
Herber, Johan	013-28 2869	E++, 1st fl., 209	`johhe`
Hirsch, Richard	013-28 1937	E, 1st fl., F 482	`richi`
Hjerppe, Roland	013-28 1965	E, gd. fl., F 454	`rhj`
Holmgren, Hans	013-28 1966	E, 1st fl., G 470	`heh`
Häger, Madeleine	(on leave)		
Hägglund, Sture	013-28 1431	E, 1st fl., G 442	`sth`
Jacobson, Anne-Marie	013-28 1496	E, gd. fl., G 458	`amj`
Jansson, Michael	013-28 1470	E++, gd. fl., 124	`mij`
Johannesson, Peter	013-20 7440	Berzelius school	`pjo`
Johansson, Eva	013-28 1652	E, entr. 27, gd. fl.,	`evajo`
Johansson, Mikael	013-28 1479	E, 1st fl., G 472	`mjo`
Johansson, Olof	013-28 2390	E++, 1st fl., 224	`ojo`
Jonsson, Torbjörn	013-28 2467	E, 1st fl., F 460	`tjo`
Jönsson, Arne	013-28 1717	E, 1st fl., F 484	`arj`
Kamkar, Mariam	013-28 1949	E++, 1st fl., 211	`mak`
Kim, Ezra	013-28 1891	E, gd. fl., G 478	`heeki`
Krysander, Christian	013-28 1411	E, 1st fl., F 448	`ckr`
Kuchcinski, Krzysztof ("Kris")	013-28 1883	E++, 1st fl., 228	`kku`
Kågedal, Andreas	013-28 2419	E++, 1st fl., 206	`aka`
Lagerström, Bo	013-28 2859	B, 1st fl., B 490	
Larsen, Kristina	013-28 2607	E++, gd. fl., 110	`krila`
Larsson, Mats	013-28 1819	E++, 1st fl., 230	`mla`
Lilja, Carita	013-28 1463	E, gd. fl., G 440	`cli`
Lind, Jonas	013-28 1526	E++, gd. fl., 114	`jonli`
Lind, Leili	013-28 1973	E, 1st fl., F 441 B	`lei`
Lindblom, Patrik	013-28 2594	E, 1st fl., F 440	`patli`
Lindvall, Mikael	013-28 1950	E, 1st fl., G 464	`mikli`
Linge, Lisbeth	013-28 1472	E, 1st fl., F 490	`lli`
Lingenhult, Gunilla	013-28 2297	E++, 1st fl., 212	`glt`
Loborg, Peter	013-28 2494	E++, 1st fl., 227	`plo`
Löwgren, Jonas	013-28 1482	E, 1st fl., G 452	`jlo`
Malec, Jacek	013-28 2362	E, gd. fl., F 490	`jam`
Maleki, Jalal	013-28 1963	E, 1st fl., F 470	`jma`
Malmen, Joakim	013-28 1948	E++, gd. fl., 117	`joama`
Maluszynski, Jan	013-28 1483	E++, 1st fl., 210	`jmz`
Mattsson Kihlström, Bodil	013-28 2169	E++, gd. fl., 118	`bmk`
Mattson, Lars	013-28 1088	B, 1st fl., B 480	`larma`
Merkel, Magnus	013-28 1964	E, 1st fl., F 476	`mme`

Name	*Phone*	*Room no.*	*E-mail*
Moberg, Anna	013-28 2542	E++, gd. fl., 107	annmo
Morin, Magnus	013-28 2425	E, gd. fl., G 490	mmo
Nadjm-Tehrani, Simin	013-28 2411	E, gd. fl., G 482	snt
Nilsson, Anders G	013-28 1525	E++, gd. fl., 101	andni
Nilsson, Bernt	013-28 1975	E, gd. fl., G 454	bkn
Nilsson, Björn	013-28 1476	E, gd. fl., G 446	ban
Nilsson, Fredrik	013-28 2859	B, 1st fl., B 490	freni
Nilsson, Henrik			henni
Nilsson, Ivan	013-28 2423	E, 1st fl., G 466	ivani
Nilsson, Nils	013-28 2418	E, gd. fl., G 460	nni
Nilsson, Peter J	013-28 1354	E, gd. fl., G 448	pjn
Nilsson, Rolf	013-28 2405	E, 1st fl., G 478	ron
Nilsson, Ulf	013-28 1935	E++, 1st fl., 208	urn
Noghabai, Mehran	013-28 1486	E++, gd. fl., 109	mehno
Nordling, Patrick	013-28 2594	E, 1st fl., F 446	–
Näslund, Torbjörn	013-28 1957	E, 1st fl., G 448	tor
Ohlsén, Lennart	013-28 1555	E++, gd. fl., 108	lenoh
Ollinen, Jan	013-28 2544	E++, gd. fl., 112	janol
Olsen, Hans			hanol
Olsson, Tommy	013-28 1954	E, 1st fl., F 454	tao
Orsborn, Kjell			kjeor
Padgham, Lin	013-28 2066	E++, gd. fl., 122	lin
Padron-McCarthy, Thomas	013-28 2677	E, 1st fl., F 453	tpm
Partain, David	013-28 2608	E++, gd. fl., 125	dlp
Peng, Zebo	013-28 2067	E++, 1st fl., 235	zpe
Persson, Jonas	013-28 1477	E, gd. fl., 29:151	jonpe
Persson, Mats	013-28 1948	E, 1st fl., G 467	map
Persson, Tommy	013-28 2428	E, gd. fl., G 474	tpe
Petersson, Jan	013-28 1936	E++, gd. fl., 116	jpe
Pettersson, Karin	013-28 2679	E, 1st fl., G 485	karpe
Pettersson, Mikael	013-28 2683	E++, 1st fl., 213	mpe
Poignant, Lars	013-28 1832	B, 1st fl., B 478	larpo
Ragnemalm, Eva L.	013-28 1947	E, 1st fl., F 453	elu
Rankin, Ivan	013-28 2424	E, 1st fl., G 458	ira
Rapp, Birger	013-28 1525	E++, gd. fl., 101	birra
Rauch, Ewa	013-28 1938	E, 1st fl., F 440	era
Raunio, Lise-Lotte	013-28 2430	E, 1st fl., G 482	llr
Reshagen, Lars	(on leave)		
Ringström, Johan	013-28 2406	E++, 1st fl., 217	jri
Risch, Tore	013-28 2541	E++, 1st fl., 217	torri

Name	*Phone*	*Room no.*	*E-mail*
Rundfelt, Rolf	013-28 2538	E++, gd. fl., 102	–
Rönnquist, Ralph	013-28 1979	E++, gd. fl., 121	rrq
Röstlinger, Annie	013-28 2407	E, 1st fl., G 490	aro
Sandahl, Kristian	013-28 1467	E, 1st fl., G 440	krs
Sandewall, Erik	013-28 1408	E, gd. fl., G 470	ejs
Savén, Bengt	013-28 2549		bensa
Sedvall, Göran	013-28 1977	E, gd. fl., G 442	gse
Shahmehri, Nahid	013-28 1464	E++, 1st fl., 218	nsh
Sjöberg, Cecilia	013-28 2596	E, 1st fl., F 446	cecsj
Sjölin, Martin	013-28 2410	E++, gd. fl., 129	marsj
Sjöström, Camilla	013-28 2859	B, 1st fl., B 490	camsj
Sköld, Martin	013-28 2698	E++, 1st fl., 222	marsk
Stolt, Anette	013-28 2859	B, 1st fl., B 490	an0st
Stoy, Erik	013-28 2691	E++, 1st fl., 233	est
Strömbäck, Lena	(on leave)		
Strömberg, Lars	013-28 1976	E++, 1st fl., 205	lst
Sunnerud, Katarina	013-28 1481	E, 1st fl., F 466	ksd
Svenberg, Stefan	013-28 2690	E, 1st fl., F 489	ssv
Svensson, Birgitta	013-28 2595	E, 1st fl., F 446	birsv
Svensson, Eva-Chris	013-28 1457	E, 1st fl., G 476	ecs
Svensson, Lise-Lott	013-28 2360	E, gd. fl., G 466	lissv
Söderlund, Siv	013-28 1426	E, 1st fl., G 489	sso
Söderman, Ulf	013-28 2681	E, gd. fl., F 484	uso
Thurée, Åke	013-28 2492	E, 1st fl., F 477 B	ath
Timpka, Toomas	013-28 1471	E, 1st fl., F 436	tti
Törne, Anders	013-28 2365	E++, 1st fl., 225	ath
Vainio-Larsson, Arja	013-28 1477	E, gd. fl., 29: 151	ava
Viklund, Lars	013-28 2689	E++, 1st fl., 207	lvi
Villegas, Jaime	013-28 1539	E++, gd. fl., 111	jaivi
Wahllöf, Niclas	013-28 2609	E++, gd. fl., 123	nicwa
Wallgren, Jonas	013-28 2682	E++, 1st fl., 204	jwc
Wallgren, Lillemor	013-28 1480	E, 1st fl., G 460	lew
Werner, Magnus			magwe
Westman, Richard			ricwe
Wigh, Lena	(on leave)		
Willén, Olle	013-28 1413	E, 1st fl., F 452	owl
Wretland, Åsa	013-28 2668	E, 1st fl., G 467	asw
Yi, Choong-Ho	013-28 2693	E, gd. fl., G 478	choyi
Österling, Per	013-28 2415	E, gd. fl., G 484	pos

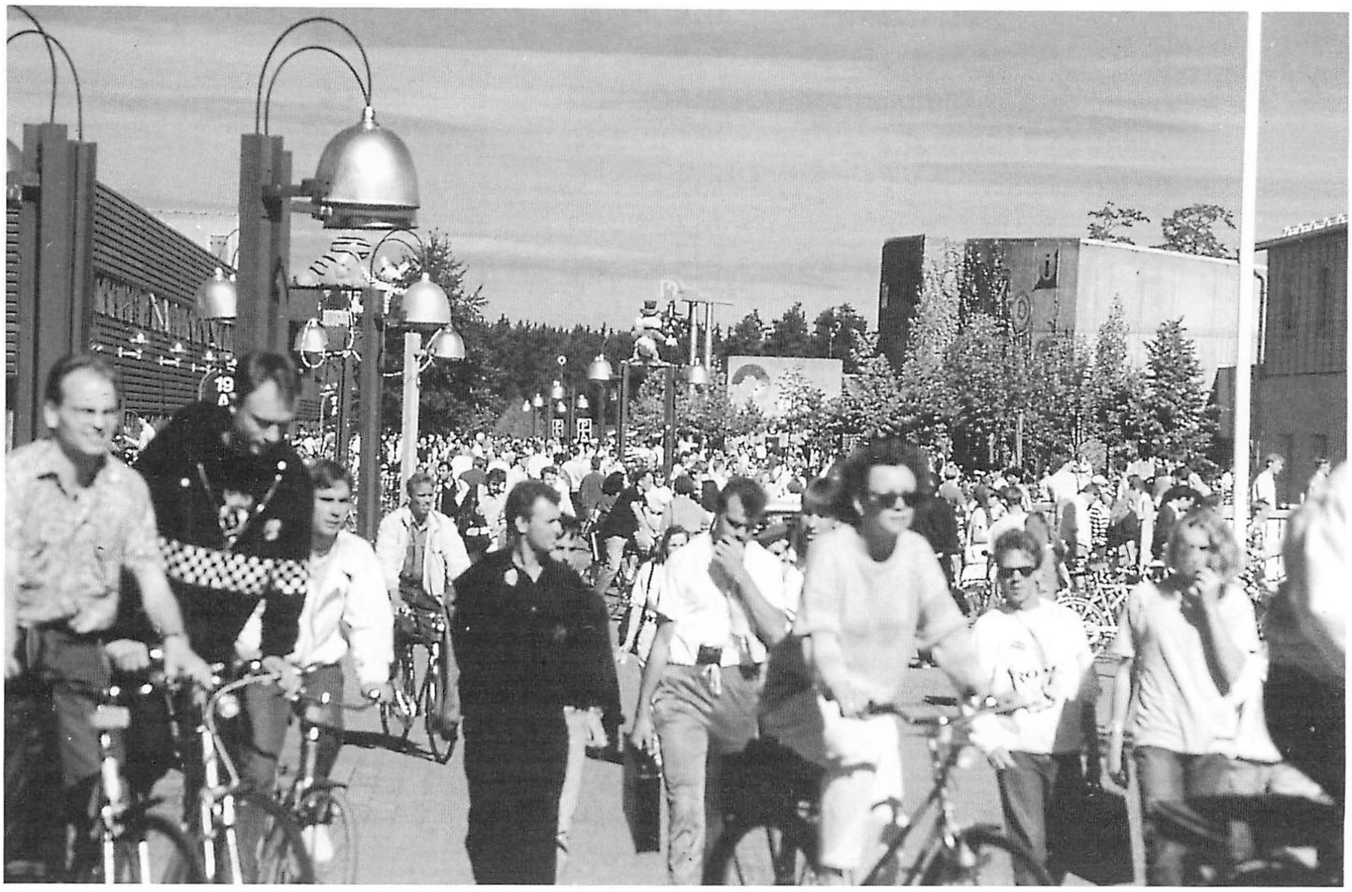
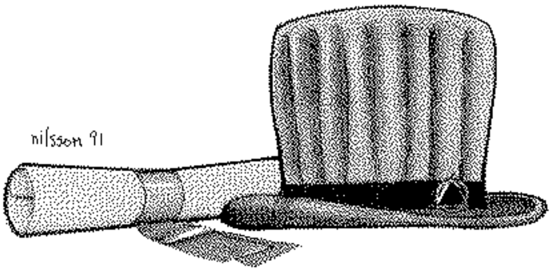
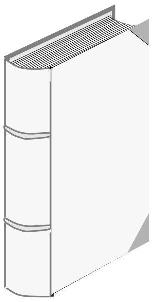


Figure F-1. *A view of the university campus.*

IDA-Events 1991–92

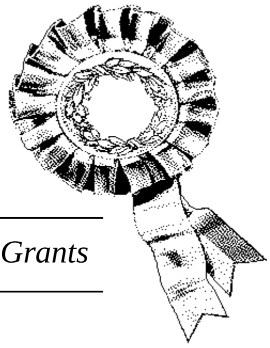


Ph.D. Dissertations and Lic. Theses



IDA-authors

Assignments



Awards and Travel Grants

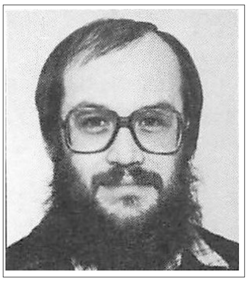
Etc...

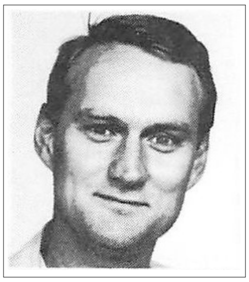
Department of Computer and Information Science

Assignments

Undergraduate Teaching

On July 1st 1991 **Olle Willén** replaced **Bengt Lennartsson** as Chairman of the Undergraduate Teaching Committee (IDUN).

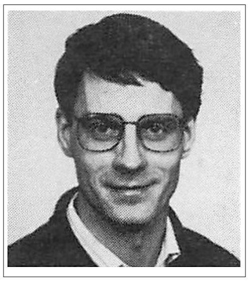




Arne Jönnson has replaced **Lennart Ohlsén** as vice chairman of the Undergraduate Teaching Committee.

Graduate Studies

On July 1st 1991 **Anders Törne** replaced **Douglas Busch** as Director of Graduate Studies.



On January 1st 1993 **Per-Olof Fjällström** replaced **Anders Törne**.

New position

Carita Lilja is IDA's new payroll officer.

IDAit new CENIIT manager

On July 1st 1992 the manager position of CENIIT, the Center of Industrial Information Technology at LiTH, was taken over by **Anders Törne**, also director of CAELAB.

The activities of CENIIT are spread over most of the departments of LiTH and the manager functions as a coordinator of the different projects. The advisory board for CENIIT has lately strongly advocated that the interdiciplinary work between projects and the industrial liaison within the projects must increase. Presently new goals and strategies for funding are under preparation.

Erik Sandewall member of The Royal Swedish Academy of Sciences

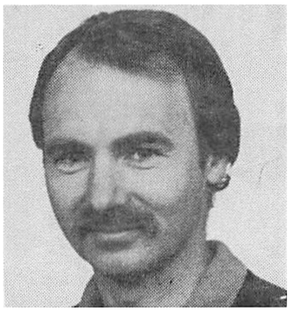
Erik Sandewall has been elected member of The Royal Swedish Academy of Sciences, which was founded in 1738.

There are ten divisions within the Academy. Erik Sandewall is in the category "Engineering Sciences".

New Research Group

A new research group, ***VITS – Information Systems and Work Contexts***, was founded in 1992.

The person in charge of the group is **Göran Goldkuhl**.



New Docents

Lars Ahrenberg, docent in Computational Linguistics**.**

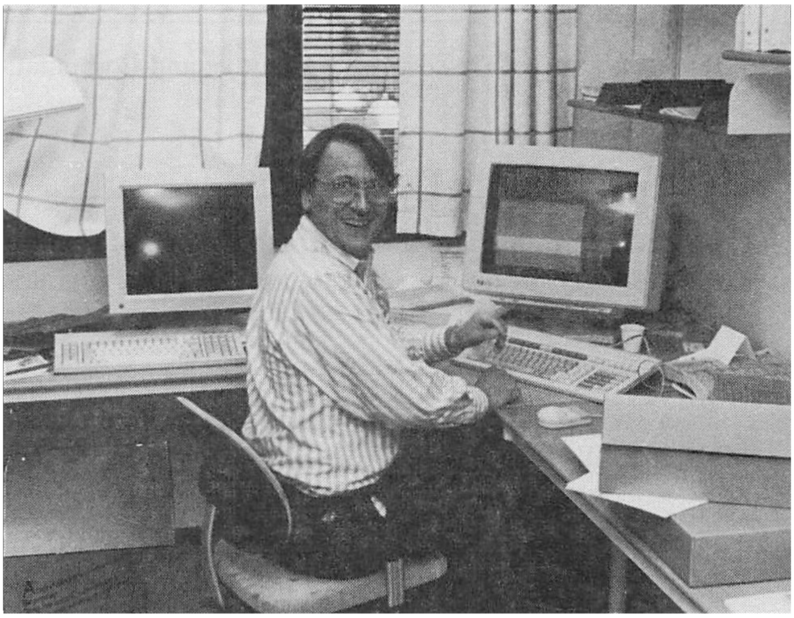
Richard Hirsch, docent in the Theme Group "Communication".

Peter Fritzson, docent in Computer Science**.**

Toomas Timpka, docent in Social Medicine in the Faculty of Medicine, and in Medical Informatics in the Faculty of Engineering.

Assignments

From Hewlett-Packard, Tore Risch received a donation worth about one-half million SEK, including three large workstations and a file server. The equipment will be used in database research, which has connections both to Ericsson and to Hewlett-Packard.

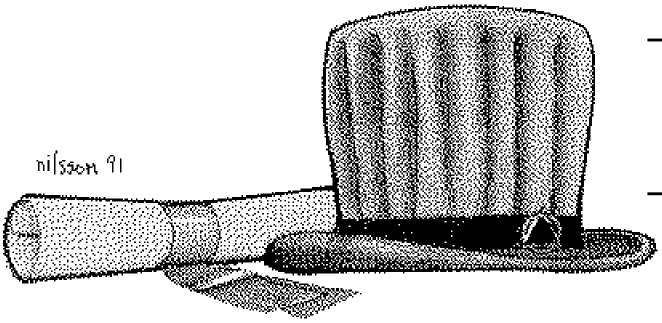


New Professor at IDA

Tore Risch, professor in Engineering Data Bases and supported by CENIIT at LiTH arrived in March to CAELAB, IDA.

Tore Risch immediately initiated a rush of activities within the research area - presently three graduate students are occupied with extensions to WS-IRIS, an object-oriented database developed by Tore Risch at Hewlett-Packard Laboratories, Palo Alto, CA. He also directly in July received funding from both Nutek and TFR for this research.

Tore Risch has worked in Silicon Valley in California since the early 80's. He had a position at Hewlett-Packard Laboratories, where he was also attached to a data-base group at Stanford University as a guest researcher.



Ph. D. Dissertations

from September 1991 to September 1992

Peter Eklund

An Epistemic Approach to Interactive Design in Multiple Inheritance Hierarchies. (No 252, 1991.)

Patrick Doherty

NML3 – A Non-Monotonic Formalism with Explicit Defaults. (No 258, 1991.)

Nahid Shahmehri

Generalized Algoritmic Debugging. (No 260, 1991.)

Nils Dahlbäck

Representation of Discourse – Cognitive and Computational Aspects. (No 264, 1992.)

Ulf Nilsson

Abstract Interpretations and Abstract Machines: Contributions to a Methodology for the Implementation of Logic Programs. (No 265, 1992.)

Ralph Rönnquist

Theory and Practice of Tense-bound Object References. (No 270, 1992.)

Björn Fjellborg

Pipeline Extraction for VLSI Data Path Synthesis. (No 273, 1992.)

Staffan Bonnier

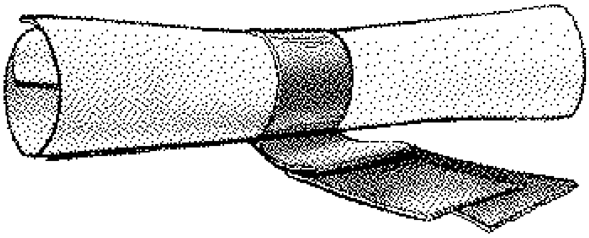
A Formal Basis for Horn Clause Logic with External Polymorphic Functions. (No 276, 1992.)

Kristian Sandahl

Developing Knowledge Management Systems with an Active Expert Methodology. (No 277, 1992.)

Christer Bäckström

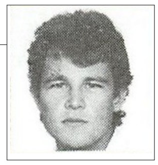
Computational Complexity of Reasoning about Plans. (No 281, 1992.)



...and Lic. Theses

from June 1991 to October 1992

Olof Johansson



Improving Implementation of Graphical User Interface for Object-Oriented Knowledge-Bases. (No 283, 1991.)

Rolf G. Larsson

Aktivitetsbaserad kalkylering i ett nytt ekonomisystem. (No 298, 1991.)

Lena Strömbäck

Studies in Extended Unification-Based Formalism for Linguistic Description: An Algorithm for Feature Structures with Disjunction and a Proposal for Flexible Systems. (No 318, 1992.)

Mikael Pettersson

DML – A Language and System for the Generation of Efficient Compilers from Denotational Specifications. (No 319, 1992.)

Andreas Kågedal

Logic Programming with External Procedures: An Implementation. (No 326, 1992.)

Patrick Lambrix

Aspects of Version Management of Composite Objects. (No 328, 1992.)

Xinli Gu

Testability Analysis and Improvement in High-Level Synthesis Systems. (No 333, 1992.)

Torbjörn Näslund

On the Role of Evaluation in Iterative Development of Managerial Support Systems. (No 335, 1992.)


ODELLER'92

The number of doctors awarded in Computer Science[1]

Academic year	*86/87*	*87/88*	*88/89*	*89/90*	*90/91*	*91/92*	*Totally*
Luleå	-	-	-	-	-	-	-
Umeå	-	-	-	-	-	-	-
Uppsala	-	-	-	-	2	-	2
Stockholm	2	1	1	1	3	1	9
Linköping	**2**	**-**	**3**	**2**	**2**	**8**	**17**
Göteborg	1	1	1	1	1	3	8
Lund	1	-	-	2	1	1	5

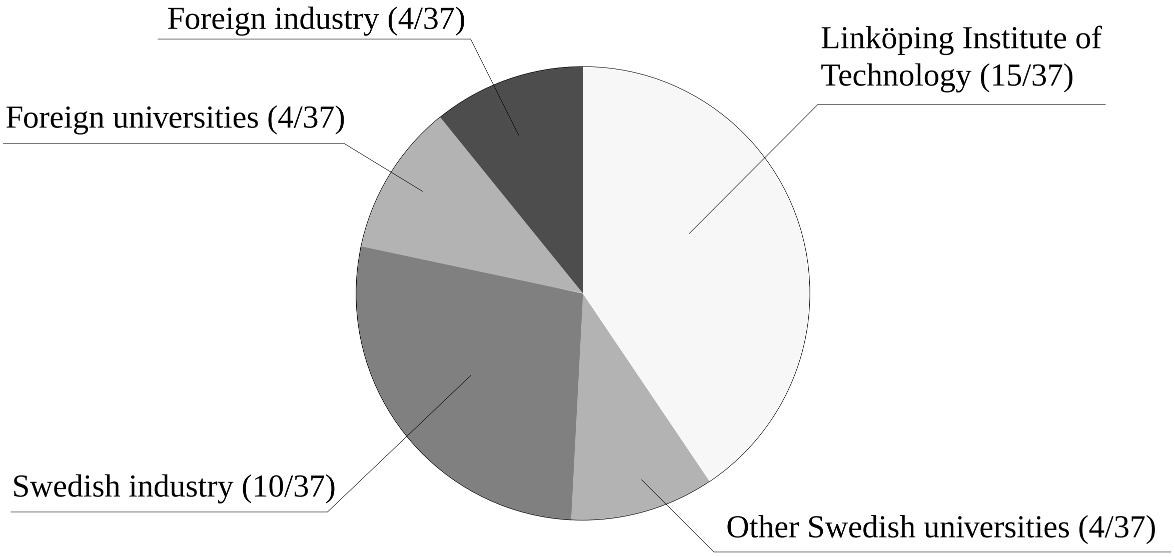
1. "Datalogi".

Doctors and Licentiates Awarded in Computer Science in Sweden 1986 – 1992

The number of Licentiate degrees awarded in Computer Science

Academic year	*86/87*	*87/88*	*88/89*	*89/90*	*90/91*	*91/92*	*Totally*
Luleå	-	-	-	-	-	-	-
Umeå	-	-	-	-	1	2	3
Uppsala	1	1	-	1	-	2	5
Stockholm	-	1	1	1	1	1	5
Linköping	**8**	**4**	**5**	**9**	**5**	**3**	**34**
Göteborg	2	2	5	1	3	1	14
Lund	-	3	-	1	1	1	6

The investigation was carried out at Gothenburg University at the request of the National Mathematics Committee.



Where did all the IDA-Ph.D.'s go?

Ph. D. Theses
Department of Computer and Information Science, Linköping Institute of Technology

No 14 Anders Haraldsson: A Program Manipulation System Based on Partial Evaluation, 1977, ISBN 91-7372-144-1.

No 17 Bengt Magnhagen: Probability Based Verification of Time Margins in Digital Designs, 1977, ISBN 91-7372-157-3.

No 18 Mats Cedwall: Semantisk analys av processbeskrivningar i naturligt språk, 1977, ISBN 91-7372-168-9.

No 22 Jaak Urmi: A Machine Independent LISP Compiler and its Implications for Ideal Hardware, 1978, ISBN 91-7372-188-3.

No 33 Tore Risch: Compilation of Multiple File Queries in a Meta-Database System 1978, ISBN 91-7372-232-4.

No 51 Erland Jungert: Synthesizing Database Structures from a User Oriented Data Model, 1980, ISBN 91-7372-387-8.

No 54 Sture Hägglund: Contributions to the Development of Methods and Tools for Interactive Design of Applications Software, 1980, ISBN 91-7372-404-1.

No 55 Pär Emanuelson: Performance Enhancement in a Well-Structured Pattern Matcher though Partial Evaluation, 1980, ISBN 91-7372-403-3.

No 58 Bengt Johnsson, Bertil Andersson: The Human-Computer Interface in Commercial Systems, 1981, ISBN 91-7372-414-9.

No 69 H. Jan Komorowski: A Specification of an Abstract Prolog Machine and its Application to Partial Evaluation, 1981, ISBN 91-7372-479-3.

No 71 René Reboh: Knowledge Engineering Techniques and Tools for Expert Systems, 1981, ISBN 91-7372-489-0.

No 77 Östen Oskarsson: Mechanisms of Modifiability in large Software Systems, 1982, ISBN 91-7372-527-7.

No 94 Hans Lunell: Code Generator Writing Systems, 1983, ISBN 91-7372-652-4.

No 97 Andrzej Lingas: Advances in Minimum Weight Triangulation, 1983, ISBN 91-7372-660-5.

No 109 Peter Fritzson: Towards a Distributed Programming Environment based on Incremental Compilation, 1984, ISBN 91-7372-801-2.

No 111 Erik Tengvald: The Design of Expert Planning Systems. An Experimental Operations Planning System for Turning, 1984, ISBN 91-7372-805-5.

No 155 Christos Levcopoulos: Heuristics for Minimum Decompositions of Polygons, 1987,ISBN 91-7870-133-3.

No 165 James W. Goodwin: A Theory and System for Non-Monotonic Reasoning, 1987, ISBN 91-7870-183-X.

No 170 Zebo Peng: A Formal Methodology for Automated Synthesis of VLSI Systems, 1987, ISBN 91-7870-225-9.

No 174 Johan Fagerström: A Paradigm and System for Design of Distributed Systems, 1988, ISBN 91-7870-301-8.

No 192 Dimiter Driankov: Towards a Many-Valued Logic of Quantified Belief, 1988, ISBN 91-7870-374-3.

No 213 Lin Padgham: Non-Monotonic Inheritance for an Object Oriented Knowledge Base, 1989, ISBN 91-7870-485-5.

No 214 Tony Larsson: A Formal Hardware Description and Verification Method, 1989, ISBN 91-7870-517-7.

No 221 Michael Reinfrank: Fundamentals and Logical Foundations of Truth Maintenance, 1989, ISBN 91-7870-546-0.

No 239 Jonas Löwgren: Knowledge-Based Design Support and Discourse Management in User Interface Management Systems, 1991, ISBN 91-7870-720-X.

No 244 Henrik Eriksson: Meta-Tool Support for Knowledge Acquisition, 1991, ISBN 91-7870-746-3.

No 252 Peter Eklund: An Epistemic Approach to Interactive Design in Multiple Inheritance Hierarchies,1991, ISBN 91-7870-784-6.

No 258 Patrick Doherty: NML3 - A Non-Monotonic Formalism with Explicit Defaults, 1991, ISBN 91-7870-816-8.

No 260 Nahid Shahmehri: Generalized Algorithmic Debugging, 1991, ISBN 91-7870-828-1.

No 264 Nils Dahlbäck: Representation of Discourse-Cognitive and Computational Aspects, 1992, ISBN 91-7870-850-8.

No 265 Ulf Nilsson: Abstract Interpretations and Abstract Machines: Contributions to a Methodology for the Implementation of Logic Programs, 1992, ISBN 91-7870-858-3.

No 270 Ralph Rönnquist: Theory and Practice of Tense-bound Object References, 1992, ISBN 91-7870-873-7.

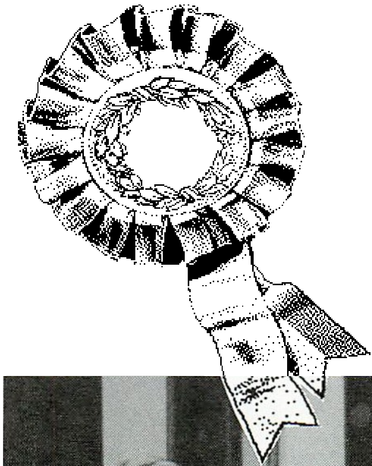
No 273 Björn Fjellborg: Pipeline Extraction for VLSI Data Path Synthesis, 1992, ISBN 91-7870-880-X.

No 276 Staffan Bonnier: A Formal Basis for Horn Clause Logic with External Polymorphic Functions, 1992, ISBN 91-7870-896-6.

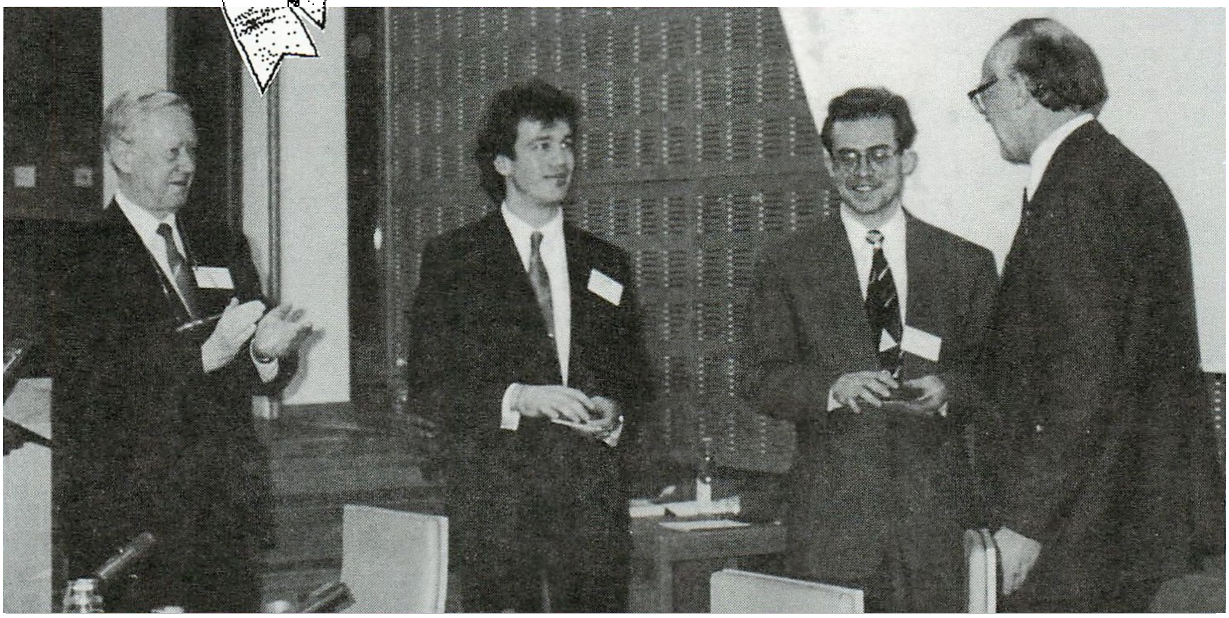
No 277 Kristian Sandahl: Developing Knowledge Management Systems with an Active Expert Methodology, 1992, ISBN 91-7870-897-4.

No 281 Christer Bäckström: Computational Complexity of Reasoning about Plans, 1992, ISBN 91-7870-979-2.

No 292 Mats Wirén: Studies in Incremental Natural Language Analysis, 1992, ISBN 91-7871-027-8.



Awards, Nominations, and Travel Grants



Gösta Lindberg, Staffan Bonnier, Ulf Nilsson and Stig Hagström at the prize ceremony at the Royal Swedish Academy of Engineering Sciences on November 25, 1992.

The Chester Carlson Prize

The Chester Carlson Prize in information technology is awarded annually in commemoration of the Swedish-American Chester Carlson who invented the xerographic copying method. The prize has been created thanks to a donation by Xerox Corporation to the Royal Swedish Academy of Engineering Sciences. A joint committee with representatives from the Academy and Rank Xerox AB decides each year on the award winner(s). This was the second time that the Chester Carlson Prize went to IDA members – Erik Sandewall received the prize in 1985.

The Chester Carlsson Prize for research in Information Science has been awarded for 1992 to **Dr. Ulf Nilsson** (tekn dr) and **Dr. Staffan Bonnier** (fil dr), both of the Department of Computer and Information Science, Linköping University, for their respective contributions to the development of new theoretical approaches in the area of Logic Programming. Each of them has separately, in his own way, dealt with problems of significance for the practical application of Logic Programming, both by improving the theoretical conceptual machinery, and by producing results which lead to an enhanced interaction between Logic Programming and other programming paradigms.

Ulf Nilsson has developed methods of abstract interpretation and demonstrated their relevance to query-evaluation in deductive data-bases.

Staffan Bonnier has developed a form of unification, S-unification, which allows conventional programs to be embedded in logic programs while preserving the efficiency of the former and the declarative properties of the latter.

The common factor in both Nilsson's and Bonnier's work is that they both use theoretical methods to facilitate the practical applicability of Logic Programming.

Financial support and Grant from Ecole Normale Supérieure, the Swedish National Board for Industrial and Technical Development, the Swedish Institute

In November 1991 **Staffan Bonnier** went to Paris to spend a postdoctorial year at Laboratoire d'Informatique, Ecole Normale Supérieure.

Financial support was supplied by three sources; *Ecole Normale Supérieure* (20.000 FF), *The Swedish National Board for Industrial and Technical Development* (180.000 SEK), and finally a grant from the *Swedish Institute* (14.000 SEK), making a two-month extension of the stay possible.

Financial support from the Swedish Institute, the Fulbright Commission, and Stanford University

Henrik Eriksson is visiting the Section on Medical Informatics, Knowledge Systems Laboratory at Stanford University, Stanford, California as a postdoctoral scholar from September 1991 to August 1993. Currently, he is working on software reuse, and on generation of domain-specific knowledge-acquisition tools in the PROTÉGÉ-II project. The PROTÉGÉ-II project aims at providing tools for developers of knowledge-based systems. In this approach, the developer assembles problem solvers for knowledge-based systems from reusable components.

Financial support was supplied by the *Swedish Institute*, the *Fulbright Commission*, and *Stanford University.*

Financial Support from SAAB Missiles, ABB Automation, and Australian Research Council

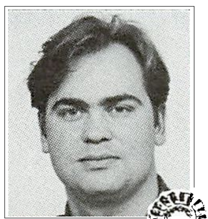
Peter Eklund received financial support from *SAAB Missiles* and *ABB Automation AB* for international studies in Japan at Hosei University in Tokyo, where he investigated fuzzy control as part of a postdoctoral stay.

Since January, 1992 he has been appointed as Senior Lecturer at The University of Adelaide, Australia.

Peter Eklund has recently secured funding from the Australian Research Council for a project in Remote Sensing, Expert Systems and Geographical Information Systems for Dryland Farming.

Awards & Travel Grants

The Sweden-America Foundation Award



Lars Degerstedt was awarded a SEK 40.000 grant by the *Sweden-America Foundation*.

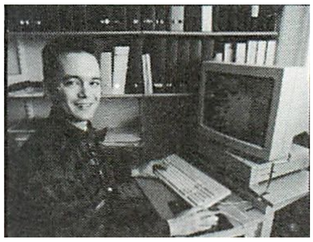
The research was focused on relations between non-monotonic reasoning in AI and logic programming. In particular, Lars Degerstedt looked at how to simplify and generalized existing computational methods in logic programming to make them more attractive for non-monotonic reasoning.

EURO-DAC ’92
(EURO VHDL ’92)

This conference is considered as the most prestigious European conference in the area of design automation for VLSI, and this year had 1750 participants.

Krzysztof Kuchcinski, **Zebo Peng**, in cooperation with the Rumanian guest researchers **Petru Eles** and **Marius Minea** got the Best Paper Award at the European Design Automation Conference, Hamburg, September 1992, for their paper “Compiling VHDL into a High-Level Synthesis Design Representation”.

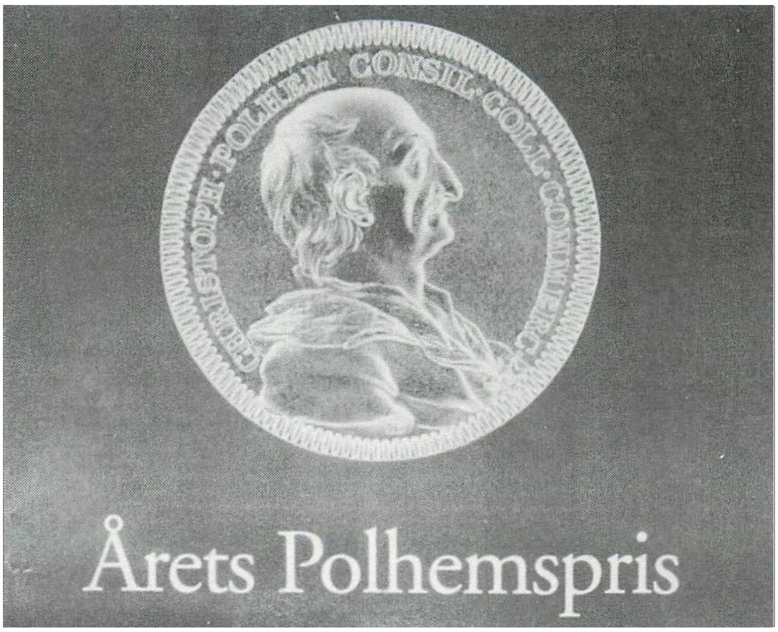
Best Paper Awards



“Lilla Polhemspriset” & Best Paper Award

The Swedish Association of Graduate Engineers annually makes an award SEK 20.000 of the best “examensarbete” (Graduation Project) carried out in Swedish schools of Engineering.

The 1992 award was given to **Jonas Persson**, Linköping Institute of Technology, for his work on digital maps for which he has developed a method to compute the shortest path between two geographical points.

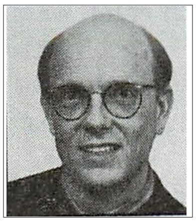


Best Paper Award

A preliminary version of the work attracted a lot of attention at an international conference in Brussels in 1991. It was given the “Best Paper” award of the conference and was published in the “International Journal of Geographical Information systems”.

Jonas Person is continuing his work in the area of digital maps as a graduate student at IDA.

The Micro-Electronics Centre Award

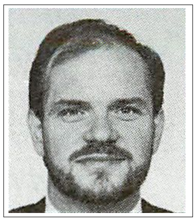


Each year the Micro-Electronics Centre bestows an award on a teacher of distinction within the area of its activities, mainly computing and electronics.

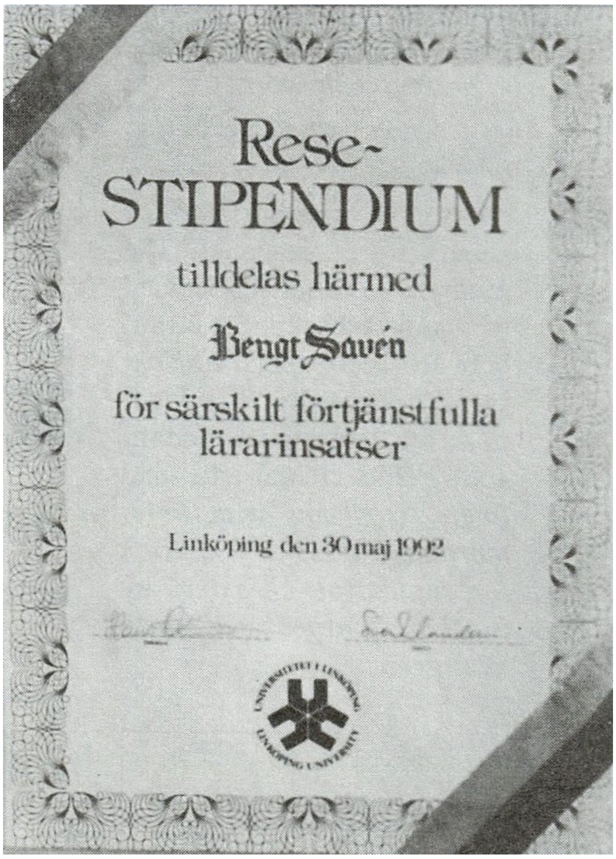
The prize for 1992 of 10 .000 SEK was awarded to **Nils Dahlbäck**, for his greatly appreciated contribution to the teaching of Cognitive Science, and his deep engagement in the subject. He has developed this subject area within the Computer Science course program, laying the foundation for a “human” communication between people and computers, and he has been a dedicated teacher in Cognitive Psychology and Models of Communication.

Awarded IDA-Teachers

The Linköping Institute of Technology Award



Bengt Savén, EIS, was bestowed the Institute of Technology award of SEK 15.000 as the best teacher of the year 1992.



Rese-
STIPENDIUM
tilldelas härmed
Bengt Savén
för särskilt förtjänstfulla lärarinsatser

Linköping den 30 maj 1992

The Golden Carrot Award

Annual nomination by the student body for the best teacher of the year. SEK 10.000 SEK for travelling.

This year two teachers from IDA were nominated, **Arne Jönsson**, by the Computer Engineering students, and **Johan Boye**, by the Computer Science students.

The 1992 award bestowed on **Johan Boye**, for “his pedagogical, enthusiastic, and inspiring way of teaching”, and for “his ability to be both very knowledgeable and able to teach on the right level”.

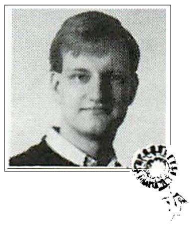
The Third Annual IDA Conference on Computer and Information Science, and the Lawson Grant

The Annual IDA Conference on Computer and Information Science is a departmental conference where graduate students are invited to present their work.

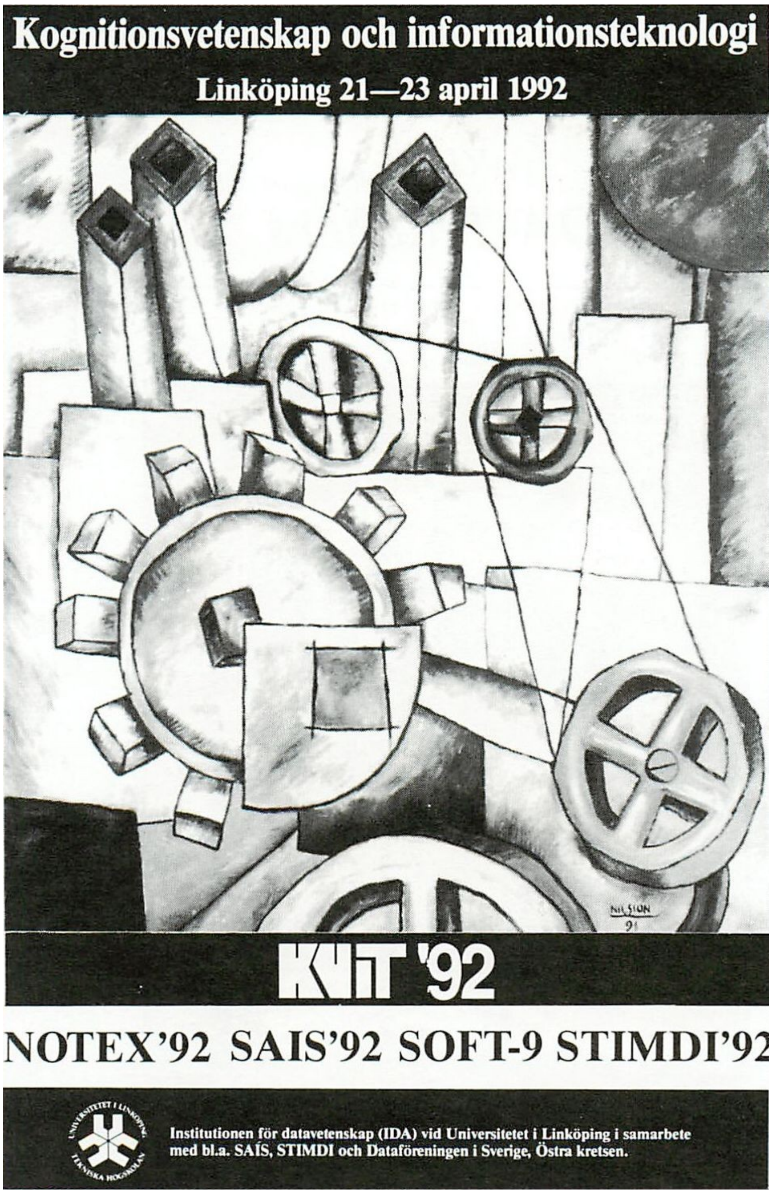
The 1992 conference was held March 17-18. It consisted of five sessions and 22 papers were presented. The program committee decided to share the best paper award between two papers, *Enriching Prolog with S-Unification*, by **Andreas Kågedahl** and **Feliks Kluzniak**, and *Logic Programs with Polymorphic Types: A condition for static type checking*, by **Staffan Bonnier** and **Jonas Wallgren**.

In connection with the conference the Lawson Grant was handed over. The grant was founded in 1989 and is awarded to people at the department, who have attained well-deserved international recognition.

The 1991/92 grant was bestowed on two persons, **Christer Bäckström**, for his research in the area of Complexity Analysis for Planning and Temporal Reasoning Algorithms, which has received well-deserved international recognition, and **Toomas Timpka**, for international activity, which includes the establishment of several European co-operation projects in medical informatics and an impressive number of research articles in several international journals.



Examples of IDA Conferences



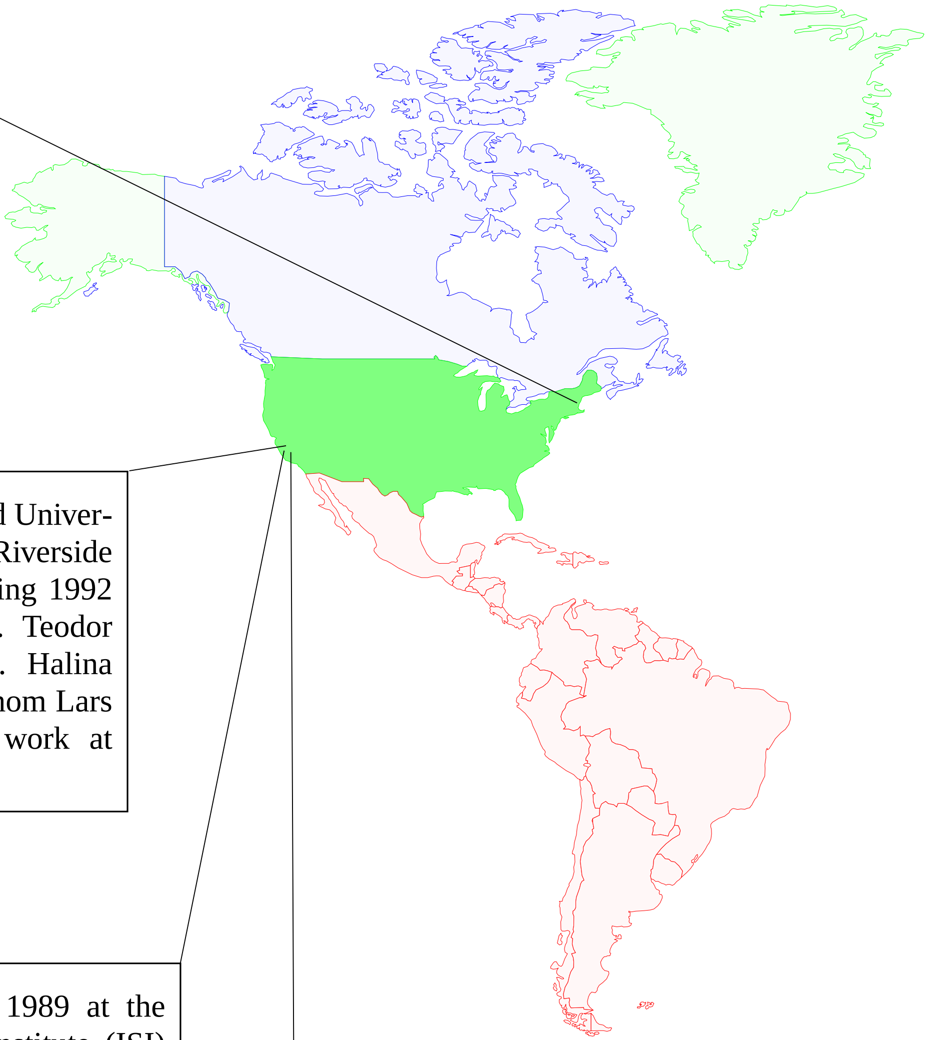
International Environment
A Source of Strength

Some IDA researchers of foreign nationality

IDAites scattered abroad 1992

Ulf Nilsson visited the State University of New York (SUNY) at Stony Brook, from September 1991 to September 1992 During his stay as a visiting scholar Ulf Nilsson completed his Ph.D. thesis and studied aspects of top-down and bottom-up evaluation of logic programs in collaboration with professor David S. Warren.

The visit was supported by the Sweden-America Foundation.



Lars Degerstedt visited University of California, at Riverside (UCR), during the spring 1992 to cooperate with Dr. Teodor Przymusinski and Dr. Halina Przymusinska, with whom Lars Degerstedt begun to work at IDA in 1991.

Peter Åberg is since 1989 at the Information Science Institute (ISI) of the University of Southern California, as a member of the IN-USE (Intelligent USer Environment) group under Dr. Robert Neches.

He is working on his PhD on user interface design and data accessibility issues, and present the projects he has been involved in at ISI and IDA.

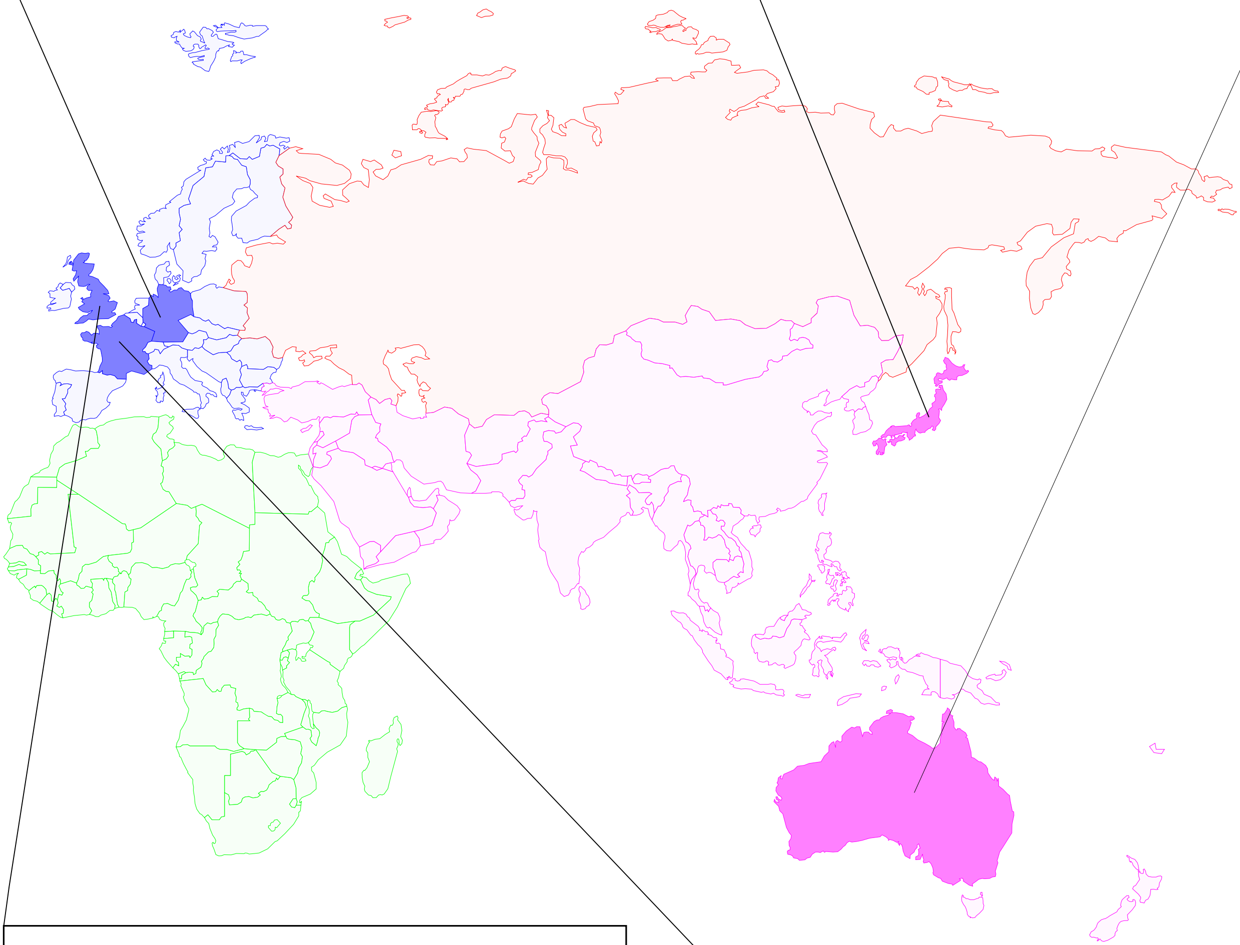
Henrik Eriksson is visiting the Section on Medical Informatics, Knowledge Systems Laboratory at Stanford University, Stanford, California as a postdoctoral scholar since Sept. 1991. Currently, he is working on software reuse, and on generation of domain-specific knowledge-acquisition tools in the PROTÉGÉ-II project.

In Jan. 1991 Siemens Corporate Research and Development started a major industrial research programme in the field of Fuzzy Control. The fuzzy-theoretic core of the Laboratory had to be reinforced with specialists in the field of fuzzy logic and AI.

These were the main reasons why ***Dimiter Driankov*** participated in the work of the Laboratory as a visiting-researcher, May-Sept., 1991 and June-Oct., 1992. Another four-months visit is planned for June-Sept., 1993.

Peter Eklund spent 4 months in Japan at Hosei University in Tokyo. There he investigated fuzzy control as part of a postdoctoral stay.

Since Jan., 1992, Peter Eklund has been appointed as Senior Lecturer at The University of Adelaide, Australia. He has been teaching AI and continue his research in Epistemic Approaches to Human Computer Interaction, machine learning and fuzzy control.



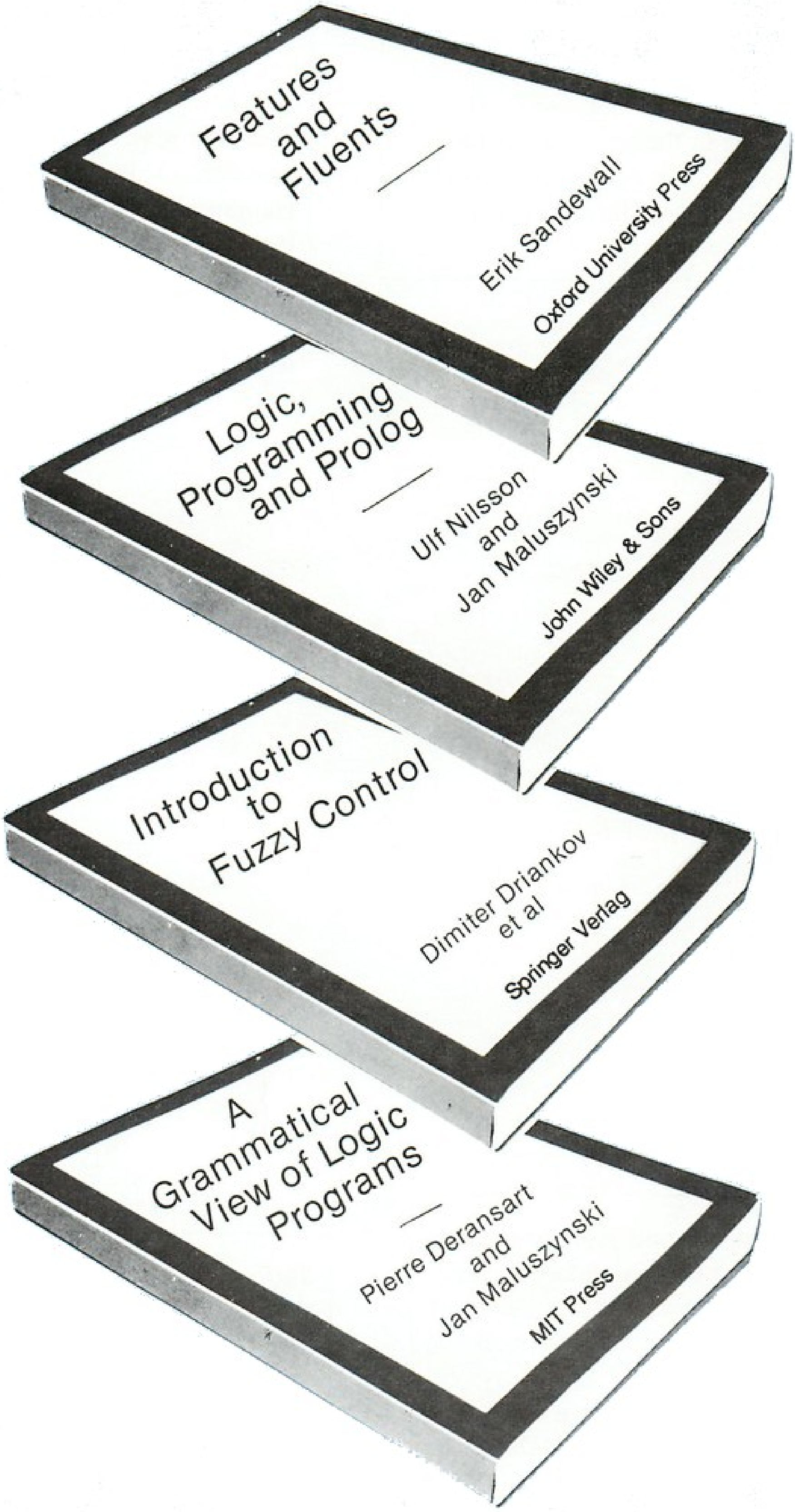
Mats Larsson has spent one year, from Sept. 1991 to Sept. 1992, at the University of Cambridge, England. He worked in the hardware verification group, led by Mike Gordon, doing work on formal methods in digital system design. This work will result in a Licentiate thesis and lay the foundation for his Doctorate project.

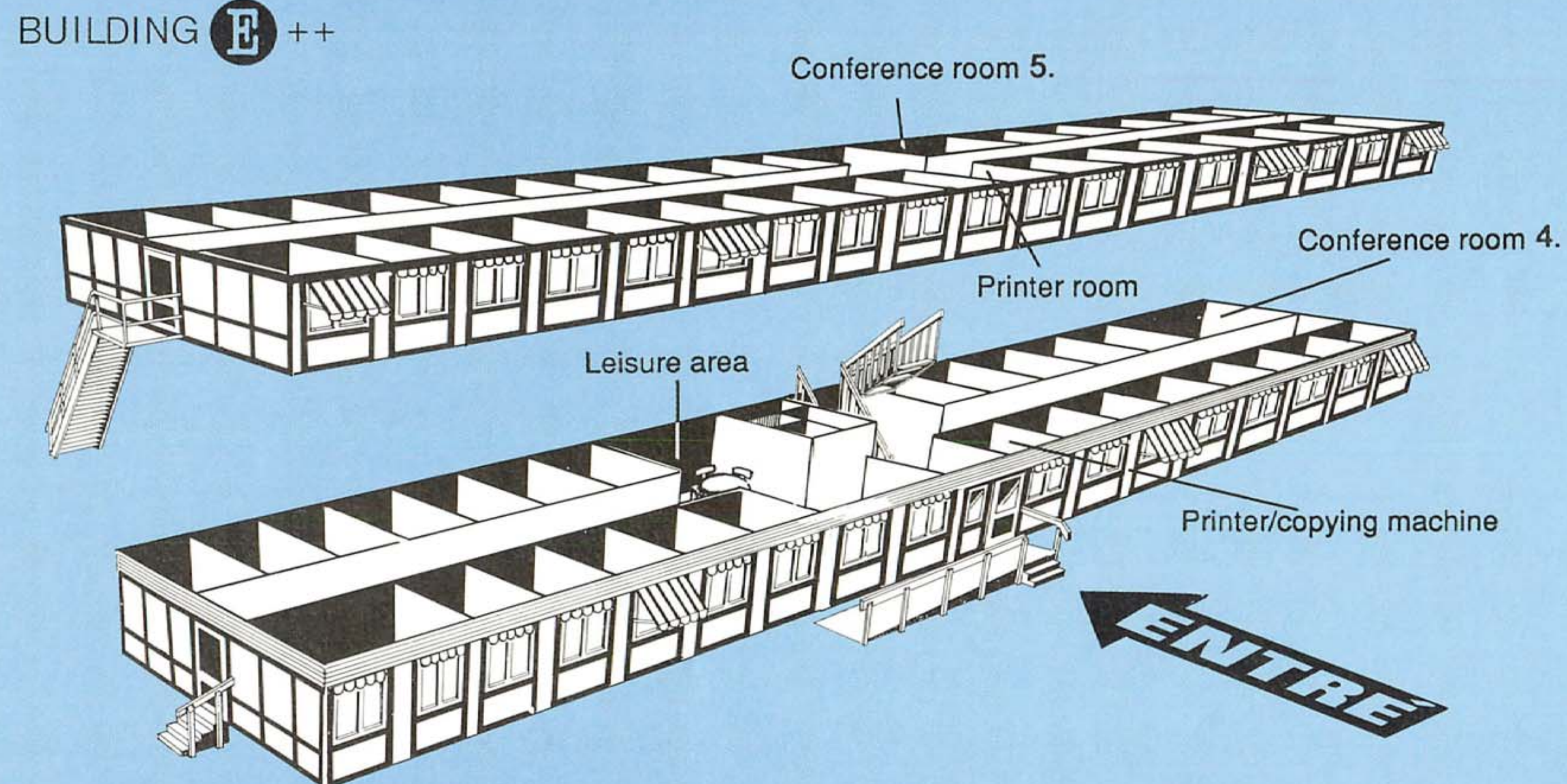
Mats Larsson's visit was made possible by the Prince Bertil Scholarship, and a supplementary scholarship from Telefonaktiebolaget L M Ericsson.

Staffan Bonnier spent a postdoctorial year at Laboratoire d'Informatique, Ecole Normale Supérieure, Paris, during the period Nov. 1991 to Dec. 1992

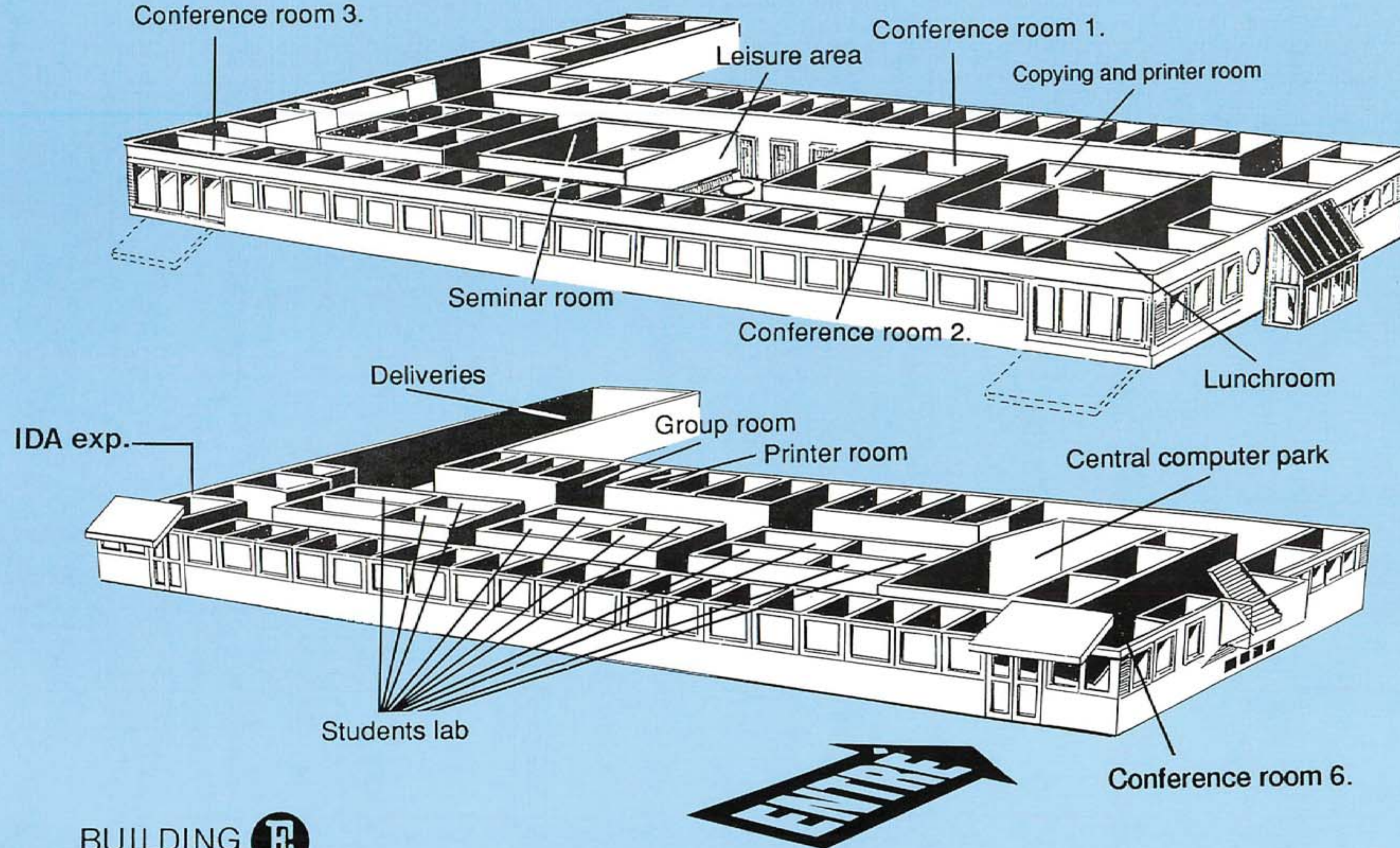
The aim of the visit was to develop methods for program verification – in particular for Horn clause programs – in collaboration with Professor Laurent Fribourg. Their work resulted in a verification method, which extends an earlier method developed by professor Fribourg.

Books recently written by IDA-authors

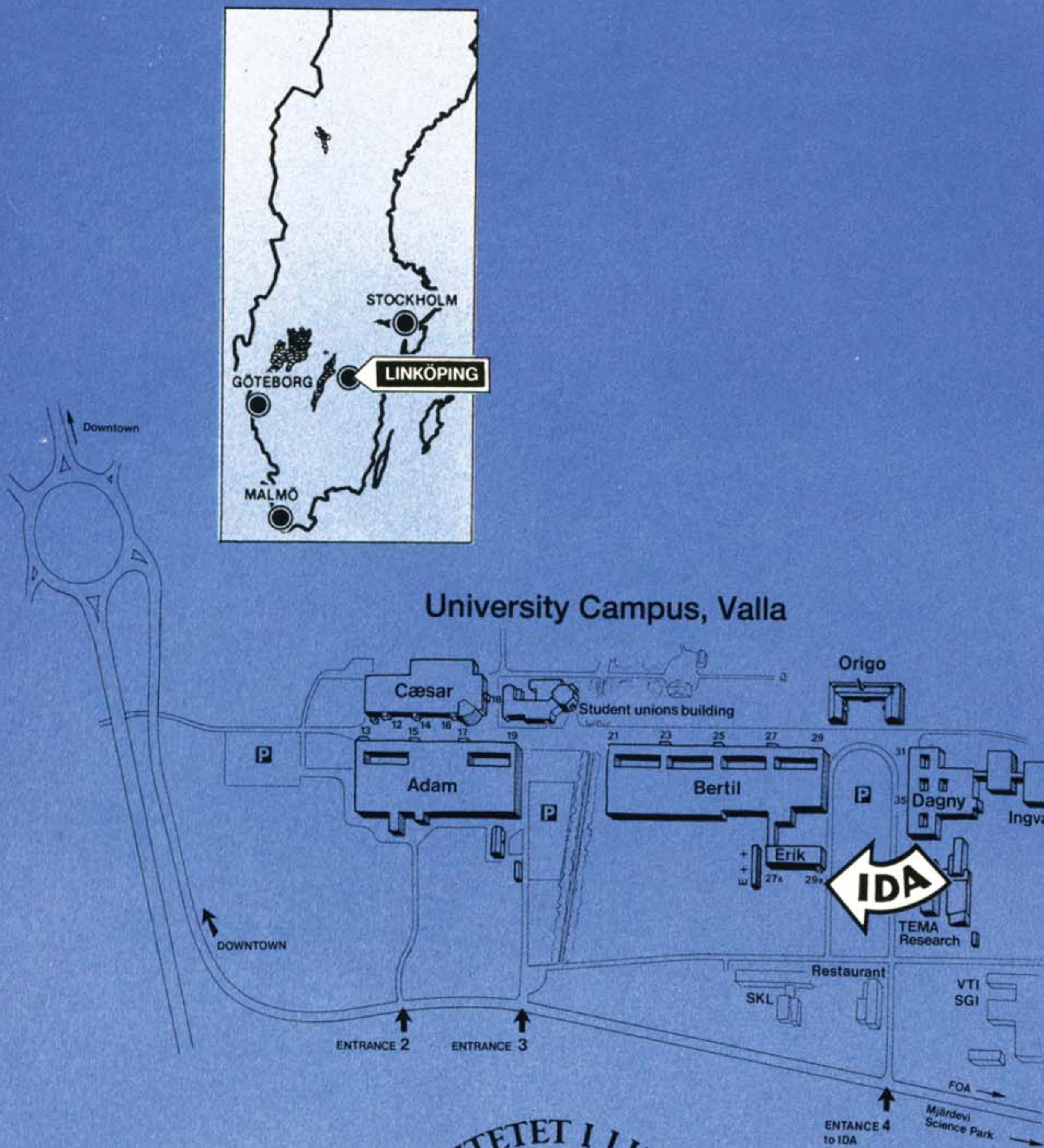




Department of COMPUTER and INFORMATION SCIENCE


Conference room 3.
Leisure area
Conference room 1.
Copying and printer room
Seminar room
Conference room 2.
Lunchroom
Deliveries
IDA exp.
Group room
Printer room
Central computer park
Students lab
ENTRÉ
Conference room 6.

BUILDING E



UNIVERSITETET I LINKÖPING

LINKÖPING UNIVERSITY