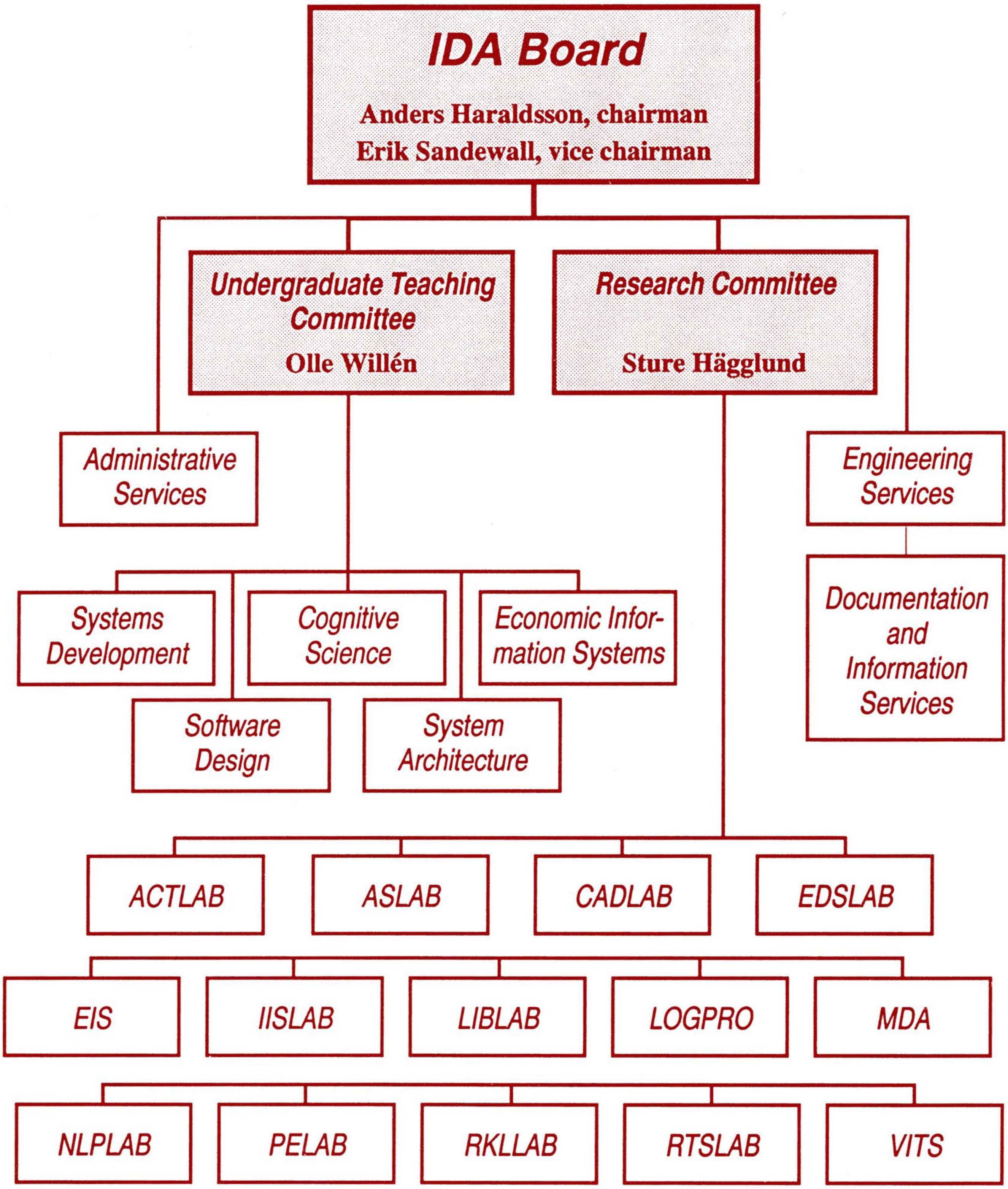
LINKÖPINGS UNIVERSITET

Department of Computer and Information Science

Activity Report

1993–94

The Department of Computer and Information Science



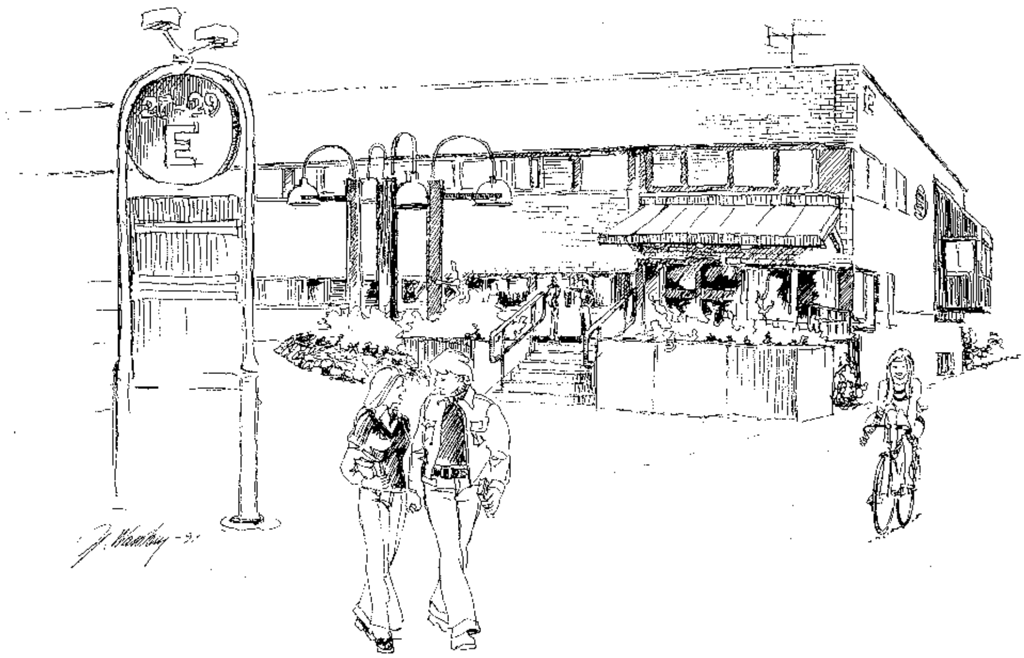
Department of Computer and Information Science

Activity Report 1993-94

ISBN 91-7871-529-6
Rapportnummer: LiTH-IDA-94-55

Department of Computer and Information Science

Mailing address:	Dept. of Computer and Information Science Linköping University S-581 83 Linköping Sweden
Phone number:	+46 13 281000 (university switchboard)
Telefax:	+46 13 142231
Internet address:	ida.liu.se
Anonymous FTP:	ftp.ida.liu.se
WWW home page:	http://www.ida.liu.se/



Data on Linköping University

Departments	39
Students*	13.000
Postgraduates	1.000
Academic and administrative staff	2.400

* Of whom 10 % are enrolled at Norrköping

The editorial board.

Lisbeth Linge (chief editor), Anne Eskilsson, Lillemor Wallgren,
Tommy Olsson.
Missing: Bengt Lennartsson (below)

Contents

Appendices

1
The Department – A Quick Guide

Formed in 1983 - Now Widely Recognized

The Department of Computer and Information Science, IDA, at Linköping University was formed in 1983 as an independent department. Since then it has grown to be the largest in its area in Sweden with several internationally well-known and recognized research groups.

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 180 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. There are eight full professors at the department and 90 full-time graduate students.

IDA has an annual budget of 92 MSEK (12.5 million US$). About 30% of the money and efforts are for undergraduate education, 5% for continuing education and technology transfer activities, and 60% for research and graduate programmes.

The computer facilities for IDA's employees and students are a distributed network including more than 300 Sun workstations, some HP workstations, an IBM AS/400, about 60 Macintoshes and 100 PCs, one Parsytec GC/Powerplus, and one SPARCcenter 2000.

Broad Impact on Industry and Society via Educational Programmes

Industry representatives often point out that teaching the next generation of computer scientists and software engineers is the most important knowledge transfer activity for a university.

IDA delivers over 130 undergraduate courses with about 7700 participants corresponding to a volume of about 600 full-time students.

IDA has also graduate programmes in:
computer science (datalogi),
computer systems (datorsystem),
information systems and media (informatik),
computational linguistics (datorlingvistik),
engineering information systems (tekniska informationssystem),
economic information systems (ekonomiska informationssystem), and
information systems development (informationssystemutveckling).

International Cooperation

In computer science, like in most other disciplines, the most important international cooperation is informal. Mutual exchange of results, ideas, and experiences takes place at international conferences and through personal contacts and visits. Such visits, ranging from a few days up to a year in some cases, are frequently organized.

In addition IDA, or specific labs within IDA, are participating in a number of organized international projects, for example the Programming Environments Laboratory is participating in seven ESPRIT projects related to the use of parallel computer architectures.

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

Graduate School in Computer and Information Science

IDA has established a Graduate School in Computer and Information Science. 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. Special agreements have been signed with some of the university colleges in Sweden, giving their staff an opportunity to participate in our graduate programme.

Special attention has been given to graduate studies for software engineers in industry. We think it is important for high-tech. companies to give their specialists such an opportunity, and we also expect such persons to convey very useful industrial experience to our research and educational programmes.

The IDA Graduate School may in the future appear as components in the proposed *Centre of Excellence in Computer Science and Systems Engineering*, and effort over the department boundaries within Linköping University.

Research School in Accounting and Auditing In operation since 1991

In cooperation with six different accounting firms 3 - 6 researchers are, since 1991, offered a special “research-school” each year. The companies are KPMG-Bohlins, Deloitte & Touche, Ernst & Young, Lindebergs, SET and Öhrlings-Reveko. The researchers have mostly graduated from Business Administration at the University. The courses are offered in compact form during some research weeks. During the rest of their time the researchers are working with their accounting firm. This special education lasts for three years and is unique. In Sweden it is only offered in Linköping as a result of an initiative taken by professor Birger Rapp.

The projects and the “research-school” are supported to some extent by KPMG–Bohlins, Deloitte & Touche, Ernst & Young, SET, Lindebergs, and Öhrlings-Reveko.

Flexible and Dynamic Internal Organization

The organization is described in more detail in Appendix A. Anders Haraldsson has been Department Chairman (prefekt) since July 1, 1990, and Erik Sandewall is Vice Chairman (proprefekt).

Separation of management responsibilities

We have chosen to separate the executive responsibility for the research and graduate studies from that for undergraduate teaching. With respect to the undergraduate education, IDA has been divided into five disciplinary areas, each managed by a director of studies (studierektor), and there is a special director for the European student exchange programme, Erasmus.

The research projects and graduate education are organized in research laboratories, each conducted by a lab leader. The lab structure is intended to be changed when necessary. Appointment of lab leaders and directors of undergraduate studies is handled locally within IDA.

This organisation has turned out to be more flexible and dynamic than the usual Swedish model, where a strict hierarchical structure, shared for research and for undergraduate teaching, is more or less permanently defined by the static set of professorships at the department.

Integrated execution

The division into research labs has consciously been made different from the definition of the areas for undergraduate teaching. However, what is separated is the administrative or executive responsibility; the directors of studies vs the lab leaders. The links between the actual research and the undergraduate teaching are however very strong. Almost everyone belonging to a research lab, lead by its lab leader, is also responsible for undergraduate teaching in one of the five disciplinary areas. In the annual planning and budget process the proportions of teaching, research, and other obligations for the coming year for each teacher/researcher, are decided in an agreement between the teacher/researcher, the lab leader, and the director(s) of studies. For most academic persons at IDA, the teaching share each year is in the interval 20-80%.

Convertible resources

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 courses from the department, for a price that is set in kronor. The "studierektor" uses the money partly for paying people to teach some of the courses. The lab leaders see a number of 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 it is the task of the directors of studies and the laboratory leaders to maximize the performance with respect to its research programme and education responsibilities, within the constraints of the rules.

Ambition and Ability in Linköping to Develop New Educational Programmes

IDA has been heavily involved in the development of several new undergraduate programmes. First the Computer Science programme in the early eighties and most recently the Cognitive Science programme, that started 1994. The Information Technology programme, based on problem-based-learning, will start in 1995.

At the undergraduate level IDA gives about 130 courses in particular within the programmes at the master level:

applied physics and electrical engineering (teknisk fysik och elektroteknik)
*cognitive science*s (kognitionsvetenskap)
computer science (datavetenskap)
computer science and engineering (datateknik)
industrial engineering and management (industriell ekonomi)
mechanical engineering (maskinteknik)
systems analysis (systemvetenskap)

The "Directors of Undergraduate Education"

The undergraduate courses have been grouped into five areas, each managed by a *director of undergraduate education*. There is a also special coordinator for the student exchange programme Erasmus, within the European Union.

Area presentations in Chapters 5 - 10.

System Development
Lise-Lotte Raunio
The area involves methodologies for activity-oriented develo
tion systems accordin
vian method for syste

Cognitive Science
Jalal Maleki (Arne Jönsson)
This area includes courses in artificial intelligence, linguistic
psychology. The cours
cording to the objectiv

System Architecture
Johan Fagerström
The courses are concentrated around computer systems, compilers and interpreters, computer architecture, distributed systems, formal languages and pro-

Program Design
Olle Willén
The over 30 courses given are mainly related to the fields c
languages and parad
tures, and program c

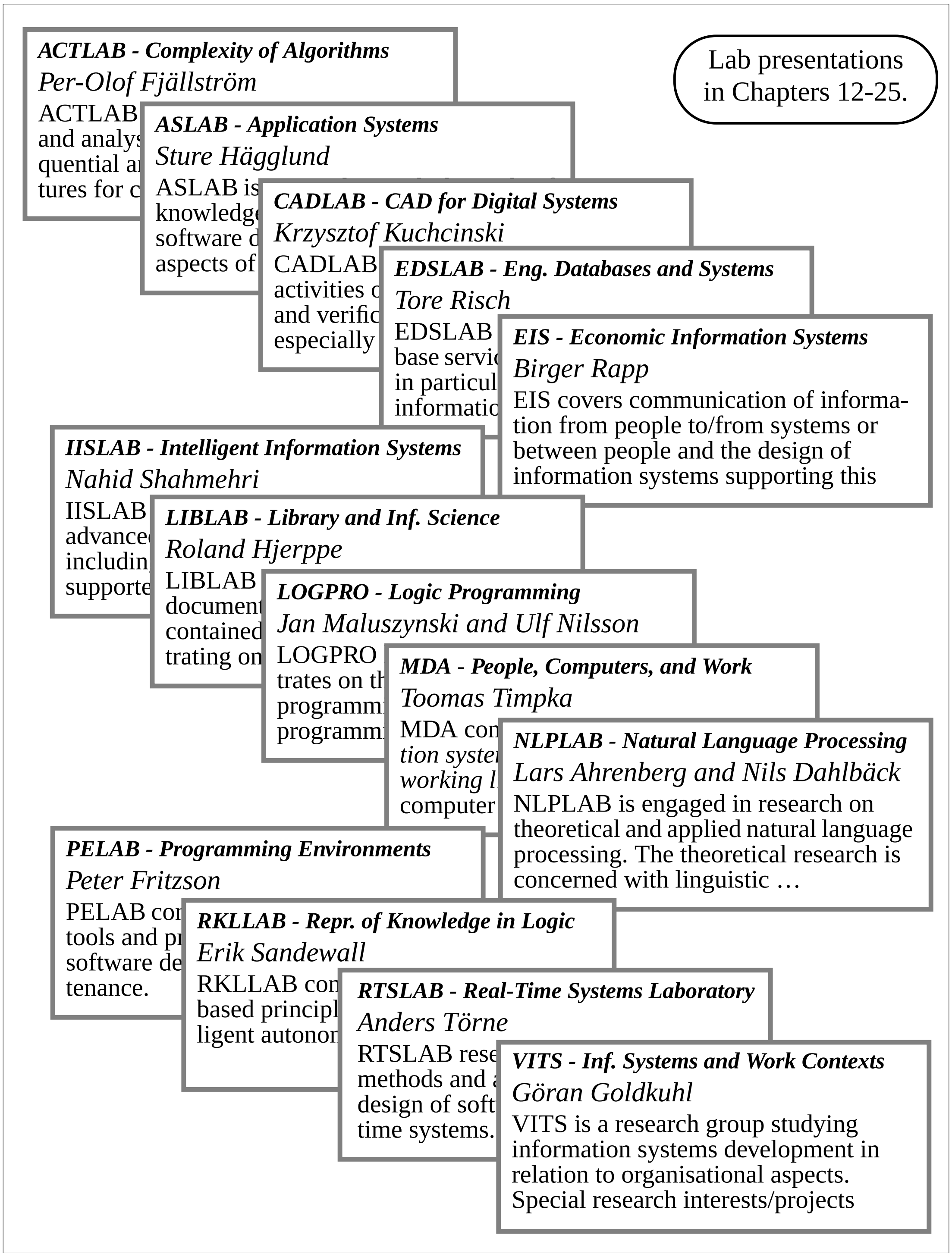
Economic Information Systems
Jörgen Andersson
EIS provides courses m
programme Industrial I
Management, but also
ministration and System

ERASMUS Coordination
Nahid Shahmehri
The Department is coordinating the European-based programme in the fields of Informatics and Computer Science. This is a European university network

Most of the actual teaching in all the different courses is done by persons sharing their time between a research lab and one (or sometimes more) of the five education areas. The role of the director of studies is to take care of the assignment of lecturers and teachers to all the courses, and to coordinate the activities.

Navigation Among the Research Labs

There are presently fourteen research groups and laboratories in the department. Each of the research labs is presented in a chapter of its own. Some of the labs are based on specific research methods, some on particular application domains, and some on an established sub-area within computer and information science.



Each laboratory consists of a few senior researchers, five to ten (typically) graduate students, and some lab-specific support staff.

ADM

2 Administrative Services

Inger Emanuelson
administrative manager

Some of the the Administrative Office personnel.
Back row: Inga-Lill Andersson, Lena Larsson, Carita Lilja, Madeleine Häger.
Along the stairs: Gunilla Blom-Lingenhult, Britt-Marie Ahlenbäck, Lise-Lotte Svensson.
Sitting back row: Eva-Britt Berglund, Inger Emanuelson
Lena Wigh, Bodil Mattsson-Kihlström
Birgitta Franzén, Camilla Edgar, Lisbeth Linge, Anne Eskilsson.

2.1 Administrative office personnel

The administrative office, headed by Inger Emanuelson is responsible for administrative services including economic and staff services. The administrators are members of the administrative group as well as members of one or more educational or research groups. In that way the administrators are more prepared to step in for each other but everyone still has her/his own tasks.

Below is a list of the administrative office personnel and their responsibilities.

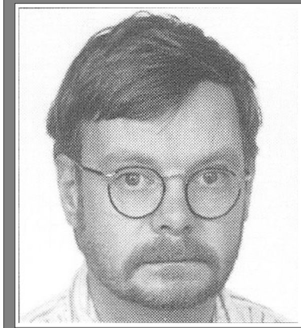
Inger Emanuelson	administrative manager.
Britt-Marie Ahlenbäck	general educational secretary.
Ingalill Andersson	substitute administrator of the group for People, Computers and Work.
Eva-Britt Berglund	administrator of the program Systems Analysis, and the program Cognitive Sciences, and the area System Development, and the subject area Systems Development, and the subject area Cognitive Sciences.
Camilla Edgar	economic administrator.
Barbara Ekman	administrator of the subject area Program Design and the Laboratory for Natural Language Processing.
Eva Elfinger	administrator of the research group for Economic Information Systems.
Anne Eskilsson	administrator of the Laboratory for Engineering Databases and Systems.
Birgitta Franzén	administrator of the Laboratory for Applications Systems.
Madeleine Häger	office assistant and administrator of the Laboratory for Complexity of Algorithms.
Anne-Marie Jacobson	administrator of the Technical Support Group.
Lena Larsson	substitute administrator and extra resource.
Carita Lilja	staff administrator, and the department registrar.
Lisbeth Linge	project leader of the group for Documentation and Information.
Gunilla Blom-Lingenhult	administrator of the subject area Computer Architecture, and the laboratories of Computer Assistance in Engineering, Logic Programming, Programming Environments.
Bodil Mattsson Kihlström	economic administrator for the department and administrator of the Laboratory for Library and Information Science.
Nils Nilsson	department porter.

Lise-Lott Svensson	administrator of the laboratory for Representation of Knowledge in Logic.
Lillemor Wallgren	general research secretary.
Lena Wigh	administrator of the subject area Economic Information Systems, and of the Laboratory for Intelligent Information Systems, and the Laboratory for Development of Informations Systems and Work Context.

Figure 2-1. *Kansligruppen.*
Anders Haraldsson, Nils Nilsson,
Bodil Mattson, Inger Emanuelsson, Carita Lilja, Camilla Edgar.

TUS

3 Technical Services



Anders Aleryd
managing engineer

Some of the members of the TUS group.
Björn Nilsson, Peter J. Nilsson, Leif Finmo, Göran Sedvall, Anders Aleryd, Bernt Nilsson, Annemarie Jacobson.

3.1 Technical services personnel

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.
Annemarie 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.
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.

4 Undergraduate Educational Programmes

Undergraduate education in Sweden is organized in *study programmes*. Each is made up of a set of courses from different departments and subject areas. Figure 4-1 below shows the higher levels of degrees in the Swedish university system.

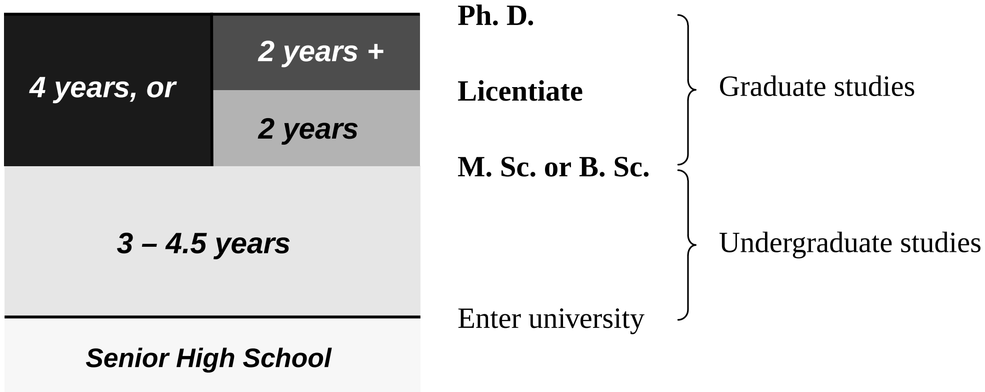


Figure 4-1. *Levels of degrees in Sweden.*

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. Such a study programme is normally composed of a proper mix of courses of various subjects, the sum of which will lead to the desirable competence. Besides, students can chose to study single subject courses and compose a mix of their own. To obtain the bachelor or master degree, however, the student is required to take courses from one subject up to a specified level of advancement, and it is also required to accomplish a final thesis. 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. In addition, Swedish universities offer study programmes of shorter extent; they are often just 2 years long, and since they are more oriented toward practical work than theoretical studies they do not aim at a bachelors degree and do not give admission to graduate studies.

The role of the department in all of these study programmes is to sell, develop, 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, and they have the responsibility for the overall contents and structure of a study programme. The courses given by our department for 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.

4.1 Undergraduate education at IDA

Undergraduate education accounts for roughly 35% of the total budget of the department, including a small rate of continuing education and technology transfer activities. (Approx. 30 MSEK together for 94/95.) The efforts required to carry out our undergraduate education commission is approximately equivalent to 40 full time teacher positions. The influx of students as well as the total efforts have increased slightly during the last three years. We have however realized that the finances have not increased at the same rate.

4.1.1 Organization of undergraduate education

The department is responsible for the formal subject areas, Computer Science (datalogi), Computer Systems (datorsystem), Economic Information Systems (ekonomiska informationssystem), Administrative Data Processing (administrativ databehandling, ADB), and Informatics (informatik). We deliver a set of courses, and in principle any teacher could be assigned any course, no matter which study programme committee has ordered it.

Under the department board, there is the Undergraduate Teaching Committee (IDUN, IDA's undervisningsnämnd), since July 1991 headed by Olle Willén. The committee is formally responsible for the contents of courses given by the department and the assignment of teachers to courses. Britt-Marie Ahlenbäck is the secretary of the committee, and she is in charge of all the common activities concerning undergraduate education. There are also representatives from the student unions in this committee as well as teachers, researchers, and technicians.

The practical and every day management of all our courses are delegated to five executive directors of studies (studierektorer). They each head one 'subject area', and such an area has been composed from a course content point of view. The Economic Information Systems courses form a coherent area, and the same applies to the Cognitive Science area. Out of the big choice 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 the three other subject areas. It is desirable that many study programmes find their courses in different subject areas, and we have the intention of letting teaching staff with various interests meet in one subject area. The responsibilities of the executive directors are among other things to staff the courses, hire students to assist in training, lead the upgrading of courses, negotiate with the programme committees, and, of course, to act 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 4-2, 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. The Erasmus student exchange programme is presented in a separate chapter.

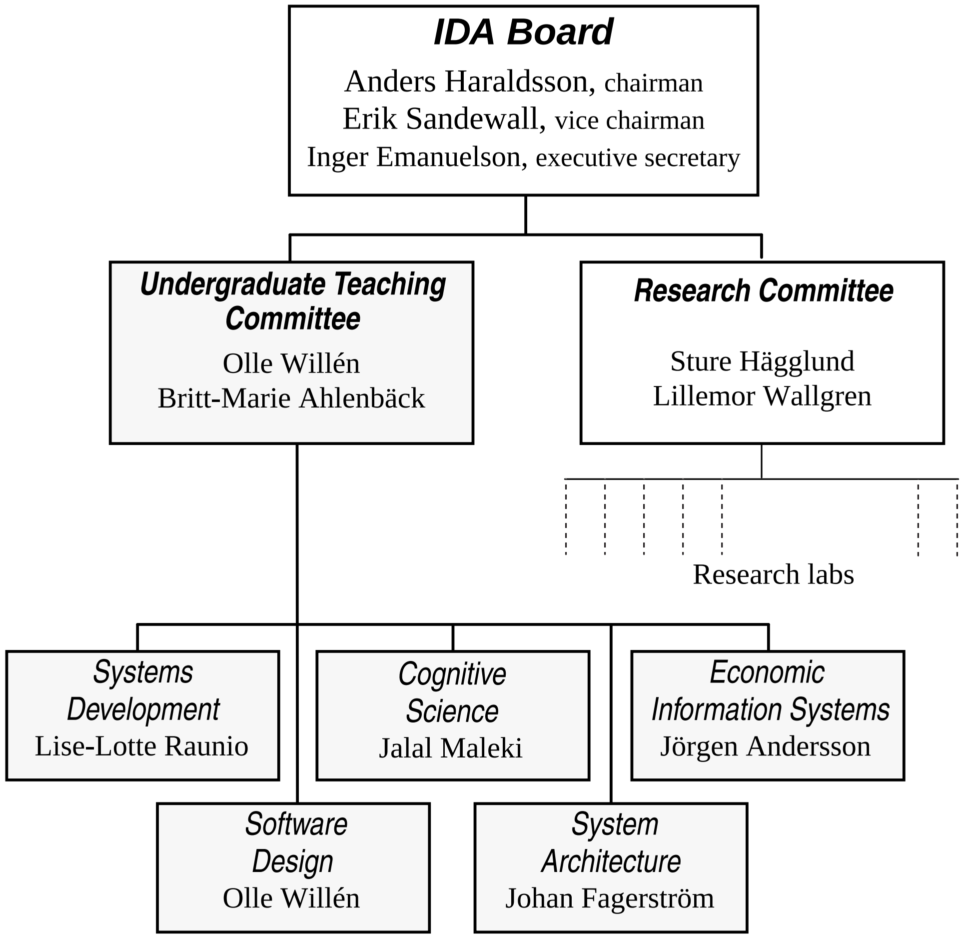


Figure 4-2. *The strategic and executive responsibilities for undergraduate education at IDA 1991/92*

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 programme 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.

4.1.2 Undergraduate curricula

Linköping is the only Swedish university offering all of the following set of Computer Science, Systems Analysis, or IT related curricula:

Master programmes (magister- och civilingenjörsprogram):

- Computer Science (datavetenskapliga programmet), the C programme
- Computer Science and Engineering (programmet för datateknik), the D programme

- Systems Analysis (systemvetenskapliga programmet), the SVP programme
- Administrative Data Processing (administrativ databehandling), the ADB single subject curriculum
- Cognitive Science (kognitionsvetenskapliga programmet), the KGP programme

Bachelor programmes (kandidatprogram):

- most of the programmes above might optionally lead to a bachelors degree
- Computer Engineering (dataingenjörsprogrammet), the three years version of DI programme
- User-oriented Systems Development (programmet för användarinriktad systemutveckling), the ASP programme managed by the department of Technical Education (ITU) in Norrköping

University Diploma in Engineering (ingenjörsprogram):

- Computer Engineering (dataingenjörsprogrammet), an optional exit after 2 years from the DI programme

Single subject curricula:

- A set of courses in the area of Economic Information Systems (EIS).
- Courses covering up to 40 study points in the area of Informatics.

The intended goal for a Swedish full-time student is to take 40 study points (‘credits’) every year, each point corresponding to one week of full time studies. This means that the masters degree could be reached after 4 years (4,5 years in the case of engineering studies) and the bachelor after 3 years. Courses are also arranged in increasing levels of advancement. Level A is the basic level, and D is the most advanced. To obtain the degrees it is required that the student takes a minimum amount of points altogether, that a specified portion of these points lay within one single subject area, that the level of advancement is high enough, and that the student writes a final thesis of a specified size at this level and within the main subject. To get a masters degree 80 points in one subject are required, and an additional 80 points in more or less selective subjects. 20 points has to be taken from level D of the main subject, and the thesis must correspond to 20 points, the most of which are related to the D level. For the bachelor degree a total of 120 points are required, where 60 come from the main subject, up to an including level C. The thesis at the C level must correspond to 10 points. Specialized rules might apply in some cases, but these are the general requirements.

As a response to the demand for a non theoretic, more practically oriented engineering education a number of shorter engineering study programmes was been introduced into the Swedish universities during the recent years. Initially they were two years of length and resulted in the non graduate degree of University Diploma in Engineering, but nearly every such programme has now been extended to three years. This gives students the opportunity of finishing with the degree of Bachelor of Science in Engineering. Linköping University offers six such programmes, three of them are located in Norrköping, and the remaining three have at least one basic programming course included, for which IDA is responsible. The goal of the Computer Engineering programme (DI, ‘dataingenjörslinjen’), with 30 annual attendants, 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 gives about 65% of all the courses in this programme. The two other Linköping programmes in this category are those for Mechanical Engineering (MI) and Electrical Engineering (EI).

All the programmes above are organized within the Institute of Technology, though not all of them are in engineering. The same situation also applies to the mathematics, physics, chemistry, and biology programmes. But only the mathematicians and (from 1995) the chemists really have to take IDA courses. Several mathematics students are however often found in our courses, and some of them take Computer Science as a second subject in their degree.

A new contribution to the offering of engineering programmes has been decided on by the Institute of Technology Board. This is the IT programme which will start in the fall of 95, with 30 students, and a quite different educational approach. The philosophy behind the studies is ‘problem based learning’, with ‘semester themes’ and the students taking over the a great deal of the responsibility for their own learning. This is a form of educational method that engages many IDA teachers. We are anxious to report on the introduction of this programme in the next edition.

In 1993 the curriculum SVP, Systems Analysis (Systemvetenskapliga programmet) started to replace SVL (systemvetenskapliga linjen). To design the SVP programme the contents and structure of SVL were brought up to date, slightly modified and increased to range over four years of studies. So now the Systems Analysis programme (SVP) is a programme in the faculty of Arts and Sciences leading to a Master of science degree, and it has an annual intake of 30 students. This programme aims at professional activities of design, evaluation, and implementation of computer-based information systems.

The first two years of the Systems Analysis programme constitute a common core of basic studies for all students. There are courses in Systems Development and Systems Theory, general courses in programming and in Computer Science as well as compulsory courses from other subjects.

The first half of the third year the students make free choices of courses and for the second part they take courses in prototyping, high level tools, systems development, and methods in social sciences. The third year ends with a term paper reporting a subject in the area of systems analysis.

In the final year the SVP students also make a free choice of courses, and this year ends with a term paper reporting a subject in the area of systems analysis. The master thesis in this programme is thus divided into two parts, both corresponding to ten study points, and at the C and D levels respectively.

The new master programme in Cognitive Science, KGP (kognitionsvetenskap), started in the fall semester 1994 with an intake of 30 students. The overall objective of this programme is to create the knowledge of how to make people and technology cooperate. In order to obtain this knowledge, and to be able to adapt the technology to human requirements, you need a combination of psychology, linguistics, philosophy, computer science, and other subjects. So this programme has been designed as a cooperative effort of participation from many departments; interdisciplinary studies is an essential characteristics of the programme. IDA is one of participating departments, and the original initiative to develop and establish the programme was also taken by enthusiasts from IDA.

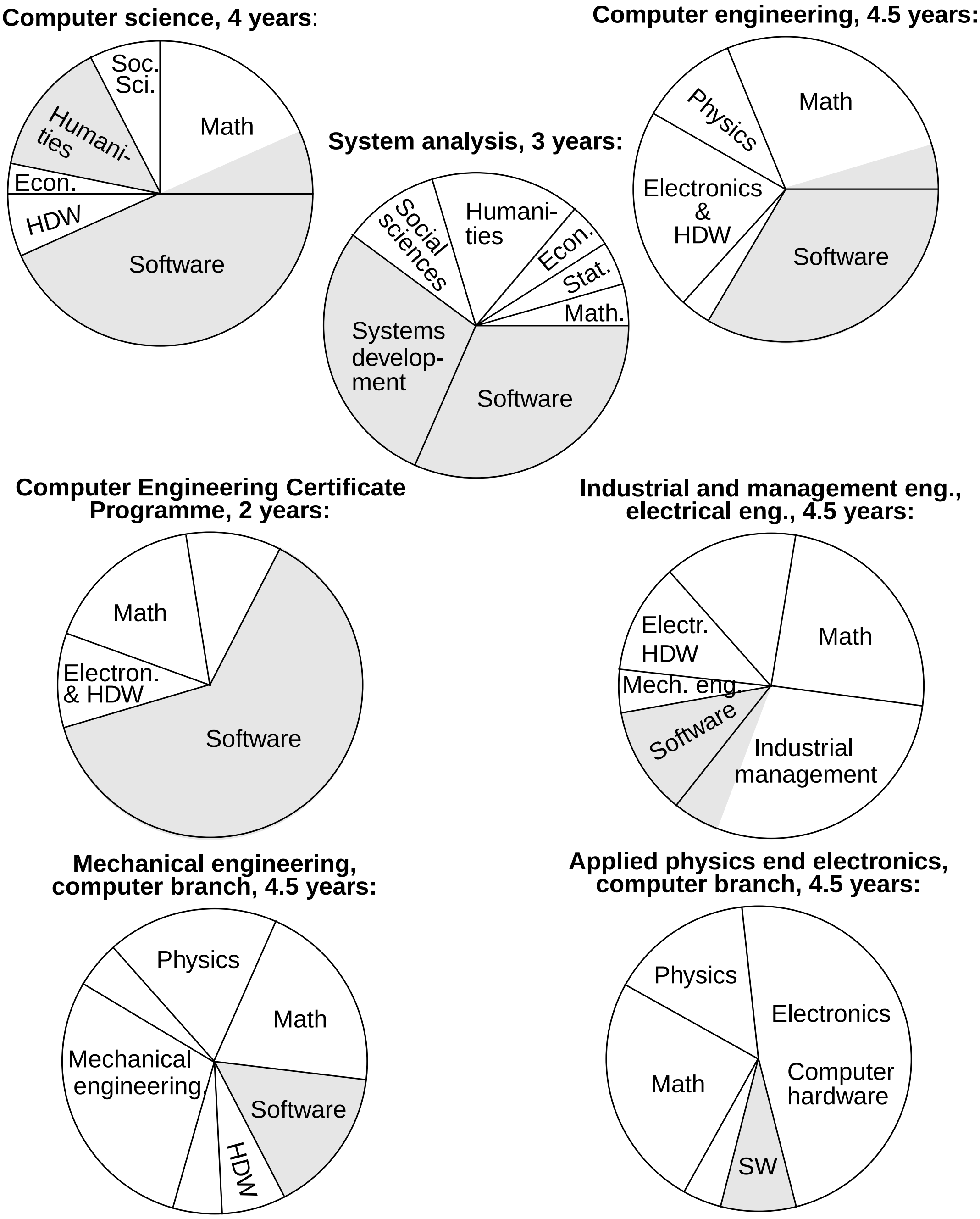


Figure 4-3. *The department share (shaded) of the selected study programmes.*

Informatics (‘informatik’) has been steadily increasing at IDA for a period of 5 years. This area consists of the study of methods and tools for retrieving, distributing, and presenting information with the support of information technology. Now we have the capability of giving courses in this subject up to 40 study points, and we have a well defined sequence of courses that are attracting some 20-30 students every year. Since the phenomena of ‘IT’, ‘Multimedia’, and ‘Internet’ just have become known to the broad public we realize that our activities in this area will develop rather rapidly.

In 1994 the university decided to offer academic courses in the summertime. Such courses should be already existing courses of 5 or 10 study points, but they had to be restructured to fit an appropriate summer schedule (July was normally excluded). Summer courses have never been given in a large extent before, but this experiment indicated that the interest really exists. Approximately 1000 students followed the courses given, so we expect that this kind of activity very soon will become one of our regular tasks. One of the courses given in 1994, a basic programming course, was given by IDA.

4.1.3 Continuing education for Swedish Industry

For the last two years the demand for continuing education programmes in computer science for industry has been very low. We believe that during the ‘bad times’ the industry in general has been looking not for academic continuing education, but rather for practical education, and in this area we can not compete. However, there have 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).

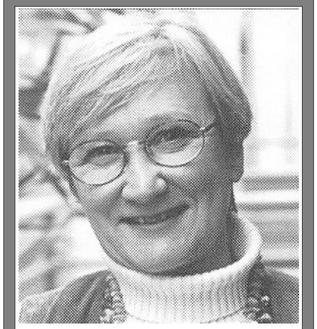
We also gave a 3 days course for 10 people at Saab Missiles in Linköping called *VHDL Hardware Description Language*. The course introduced basic VHDL concepts for hardware description. The structure modelling, timing modelling as well as behavioural modelling techniques were discussed. Finally, the course had one day laboratories to give a practical inside to the field.

For CelsiusTech Systems in Järfälla and for Ericsson Radio Systems in Linköping we have during the last two years been giving a set of courses in computer science and software engineering.

Figure 4-4. *Undergraduate Teaching Organization.*
From the back: Peter J. Nilsson, Fredrik Lönn, Lars Strömbäck
Jan Petersson, Marcus Bjäreland, Dan Laweson, Ulf Nilsson, Olle Willén, Stefan Cronholm
Jalal Maleki, Lise-Lotte Raunio, Britt-Marie Ahlenbäck, Anna Moberg,
Peter Loborg, Tea Andersson.

Systems Development

5 Subject area: Systems Development



Lise-Lotte Raunio
director

Some of the members of the System Development Area.
Back row: Cecilia Sjöberg, Karin Axelsson, Eva-Britt Berglund, Torbjörn Näslund, Mikael Lindvall, Stefan Cronholm, Ivan Nilsson.
Front row: Christian Krysander, Eva-Chris Svensson, Lise-Lotte Raunio, Mikael Johansson.
(All teachers engaged in the undergraduate programme in Appendix C).

5.1 Overview

The subject area Systems Development involves mainly 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, e.g. on the staff, working routines, duties, organizational structure. When it has been decided what to change there are methods 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.

After series of lectures, most of the courses are based on cases where the students in groups of various sizes use the actual method, supervised by a tutor.

The area consists of about 30 courses of 1300 students.

A third of them are courses in the SVP programme, in the ADP-systems analysis section. This programme covers most of the subject area.

The subject area gives several self-contained courses (courses are chosen for one term at a time) in ADP-systems analysis. These subjects are similar to the SVP courses given at the Department of Computer and Information Science. Students may take these kinds of courses up to 80 points and attain a M. A. degree majoring in ADP.

The Laboratory for Library and Information Science (LIBLAB) gives other self-contained courses, at present up to 40 points. The courses are Electronic Media I and II (about techniques, use and consequences), Traditional Information Resources and Digital Information Resources, at 10 points each. The first step in plans to increase the scope of this subject has been successfully made, and we look forward to creating a majoring subject for B. A. and M. A. exam.

The remaining courses are mainly given to students majoring in Computer Science and Computer Engineering. They deal with Systems Development, CASE, Object-Oriented System Development and a practical Software Engineering course focusing on the management of a programming project.

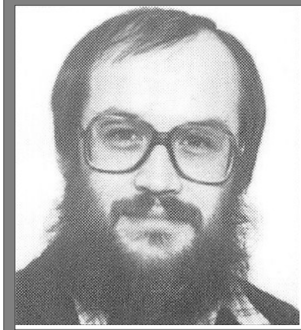
More annual statistics about the area:

- About 5500 student points are taken
- 8 full-time teachers are involved in the courses
- The students spend 25 000 scheduled hours at the terminal (PC and workstation)

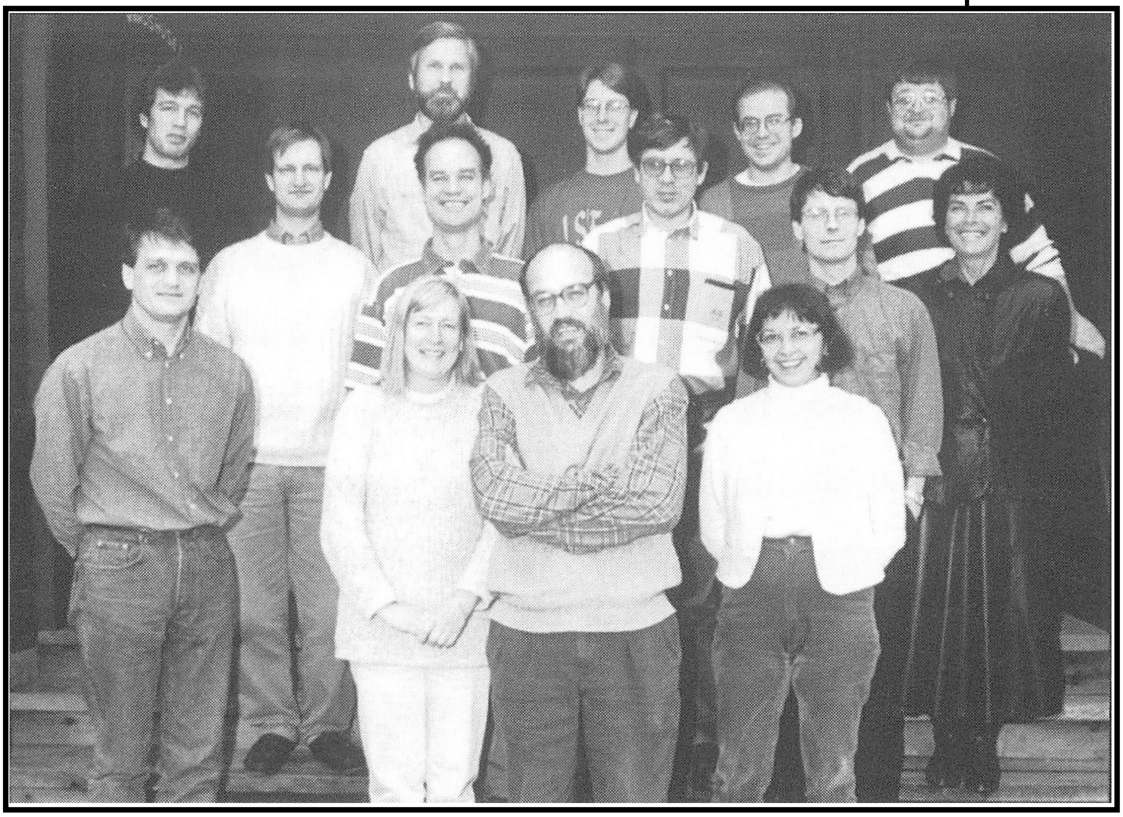
Figure 5-1. *Lecture in session.*

Software Design

6 Subject area: Software Design



Olle Willén
director



Some of the members in the Software Design Area.
Back row: Staffan Bonnier, Hans Holmgren, Torbjörn Jonsson, Ulf Nilsson, Jonas Wallgren Christer Bäckström, Åke Thurée, Anders Haraldsson, Per-Olof Fjällström, Barbara Ekman.
Front row: Tommy Olsson, Katarina Löfstrand, Olle Willén, Simin Nadjm-Tehrani.
(All teachers engaged in the undergraduate programme in Appendix C).

6.1 Overview

All students undertaking any of the master programmes at the Institute of Technology have to take at least one *basic programming course* within their first or second year of studies. This also applies to the students attending the shorter bachelor or diploma engineering programmes. In addition, the Systems Analysis (SVP), the ADP, the Statistics, the Cognition Science, the Mathematics, and now also the Chemistry programmes include such a basic programming course. These courses are of varying length and are based upon different programming languages, but all of them are given by the subject area of Software Design. The number of individual students attending these courses is approximately 1100 each year, but since they often appear in several courses the total number of course attendants is still higher. This figure is supposed to demonstrate one of the characteristics of this subject area, i e education of great volume at primary levels.

Another characteristic is the fact that nearly all the *programming languages* that are taught or used in education given by IDA, with different objectives in the various study programmes, are included in courses within this area. The first language that students learn is in most cases still Pascal, but LISP in various shapes (the C and D programmes since many years, but today also the Y and KGP programmes) is also used for this purpose. But the DI and EI programmes, which are more pragmatic, use C with a shade of C++ instead. Of course we also offer courses where Ada is the main objective, but more often Ada is taught in a context where imperative languages as a paradigm is the main goal for the studies. 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++, and even to give a conceptual overview of Fortran and Algol 68. COBOL is however no longer taught since the Systems Analysis and ADP programmes have been restructured and have introduced programming tools of higher levels.

Those students that go on from this introductory level usually have to take a course in *Data Structures and Algorithms* as their next one. This is definitely a prerequisite for all programmes that lead into further depth of Computer Science (C and D). A set of such courses are also included in the area of Software Design, but they are of different nature, more or less oriented towards mathematics and logic, depending on the level and the purpose of the education in question. We give courses with these kind of contents based on Pascal, Ada, C and C++ as well as LISP. For those students who are deeply engaged in this subject there is a possibility to take an advanced course in Design and Analysis of Algorithms at the very end of their 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. This is also a convenient opportunity of exercising the skills of documenting. Many of the basic courses within the Software Design area have 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 these characteristics, the area of Software Design give more than 30 courses a year, and these are mainly related to the fields of programming languages and paradigms, data structures, algorithms, and program development. To give these courses we require a teaching staff corresponding to 15 full time annual positions and 2000 hours of training supervised by assisting students. While taking our courses the students also consume approximately 60.000 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 one programme or another, while new ones are designed and introduced. A couple of years ago (1992) Ada replaced Pascal as the first imperative language given in the C programme, and it is now running. The introductory Pascal course, and the associated project course which were given to both the C and D programmes for a long period have come to an end. This means that Pascal in no longer taught at full extent in any of those programmes. In the SVP and ADP programmes a new and more extensive course called Program Design has substituted the Data Structure course; the purpose and contents are however principally the same. Within the same programmes the course on Program Development was given for the last time the fall of 94. A basic course on computers and programming for the Chemistry programme is under construction and will be launched in the spring semester to follow. Our first course in programming for the I programme, which has been successfully given and step-wise modified for a long run of years, has now also been adopted by the new study programme, the Engineering Biology programme.

During the last three years much effort has been made to establish and polish the four courses given to the DI programme. This has now been successfully achieved, and as a consequence our two first courses have also been included in the EI programme. Many DI students are taking a third year to obtain the bachelors degree, and this means that we find them in courses given to other programmes as well.

The most radical change is however the substitution of the single compulsory IDA course in the Y programme for two other, quite new courses. The old course was a 5 credit (study point) one, based on Pascal, it included a small project and was given in the second year. An initiative from the programme committee was taken to enhance the competence of the students by first giving them a principle based course in the second year, and then a more practical course in the third year. Both courses should give 5 credits, which doubles the IDA involvement in this programme. The intention was to use LISP (Scheme) in the first course as a vehicle to model the world and introduce the concept of abstraction. This is exactly the same idea that characterizes the start of the C and D programmes. A new course, adapted for the Y students, was designed and given in the fall of 93 for the first time (“Programming - Abstractions and Modelling”, J. Maleki and J. Wallgren). Then the follow up course should have the purpose of introducing practical applicable tools for program development. Hence this second course was decided to be based upon the C++ language, data structures, algorithms, and object oriented methodology. Since it was given in the fall of 94 for the first time (“Programming - Applications and Data Structures”, T. Olsson) it is too early to make any conclusions about the outcome or to evaluate the compliance of the objectives. It is however said to be laborious!

The new profiles, or branches, of the Mechanical Engineering programme (Aircraft Design and Systems Engineering), and the already established branch Computerized Automation have also adopted a variation of the Programming - Abstractions and Modelling course as a compulsory or selective constituent. At the time of writing it has however not yet been given.

When the Cognitive Science programme was designed, an early introduction to the use of computers was regarded as an essential feature. As with other programmes within the faculty of Arts and Sciences the programming skills is not a primary goal, nor is the detailed knowledge of a specific language. The understanding of basic algorithmic principles and the possibilities of modelling should rather act as the goal. In the sprit of these considerations the programming course for these students were developed. The efforts resulted in a course using LISP (Allegro) with approximately the same structure and ideas as in the basic courses for the C, D, and Y programmes, but with a less pretentious goal, and the course made its debut in the fall of 94 (A. Haraldsson). In order to lay the terminological and practical basis for this programming course a small outline of the IT technology was included as a smaller part of the students introductory course (O. Willén), which otherwise was given by the Cognitive Science subject area.

Many of the teachers associated with the subject area are involved in courses given within the framework of continuing education. In the programme which since 1993 is given at Ericsson Radio Systems AB we find single courses dealing with such matters as Programming in C++, Algorithmics, Logic Programming, and Programming Paradigms all of which naturally reside within the Program Design area.

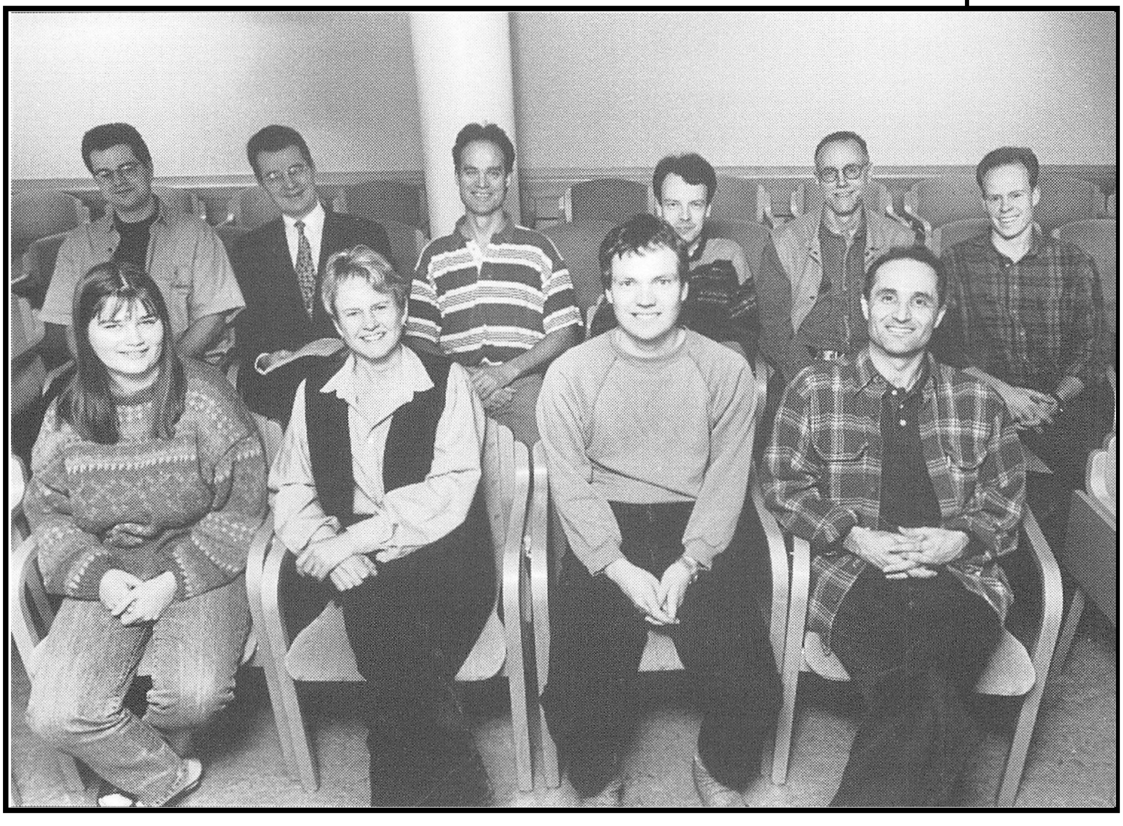
In 1994 the University organized academic courses during the summertime for the first time. IDA's contribution to this activity was the Programming - Abstractions and Modelling course, which received a great interest from the public. Approximately 20 students out of the accepted 30 passed the course.

As one element in the collaboration between the School of Engineering at Linköping University and the Technical University of Riga, Latvia, a course on Software Engineering was given in Riga in the fall semester of 93. The course served as a kind of survey of the area and was given for a group of 30 Computer Science master students during a two weeks period. We have also established a contact with the corresponding department at the Technical University of Kaunas, Lithuania, when two representatives from the Program Design area visited that department in November 94.

Cognitive Science

7 Subject area: Cognitive Science

Jalal Maleki
director



Some of the members in the Cognitive Sciences Area.
Back row: Thomas Padron-McCarthy, Erik Sandewall, Åke Thurée, Henrik Eriksson, Richard Hirsch, Jonas Löwgren.
Front row: Lena Strömbäck, Lisbeth Linge, Stefan Svenvik, Jalal Maleki.
(*All teachers engaged in the undergraduate programme in Appendix C*)

7.1 Overview

This section of IDA's annual report describes the undergraduate courses in the subject area of cognitive science. First we give an overview of the courses and in the next section we briefly describe the new study program in cognitive science in which IDA is extensively involved. For a more detailed list of the courses see Appendix C.

A large number of the courses offered by our department are of interdisciplinary nature. These courses are gathered under the topic of cognitive science. Most of these undergraduate courses are closely related to the ongoing research at our department and involve studying practical and theoretical issues in design and implementation of intelligent, flexible, and user-friendly software systems.

The teaching faculty members are mainly from NLPLAB, ASLAB, RKLLAB and EDSLAB whose research interests coincide with the subjects they teach.

Around thirty courses are offered in the following areas:

- **Discrete mathematics and logic**. Discrete mathematics and logic are central to both computer science and artificial intelligence. Currently, three introductory courses are given. These courses are obligatory and are given for computer science (C-line) computer engineering (D-line), and cognitive science (KOGVET) students.
- **Artificial Intelligence (AI)**. IDA offers four introductory AI-courses. The courses differ in prerequisites and coverage. Three of the courses have logic and Lisp-programming as prerequisites. The courses include laboratory work that involve using or implementing programs for problem solving and search, natural-language processing, planning, machine learning and neural networks.

  An advanced course covering theoretical issues in reasoning about action and change is also given for the computer science students in their final years.
- **Natural Language Processing.** Courses in this area include an introduction to linguistics followed by a more advanced course on computational linguistics. Advanced students also have the opportunity of taking a more practical course that covers implementation issues in natural language parsing and generation. This final course even covers topics such as translation to logical form, compositional semantics and semantic interpretation.
- **Foundations of cognitive science**. The core cognitive science courses include an introductory course in cognitive science covering the whole field, an introductory course in cognitive psychology, an advanced course on cognitive models of communication, and a course on theories in cognitive science.
- **Human-computer interaction (HCI)**. Design, implementation and evaluation of user interfaces are topics that are covered in a number of HCI courses offered by our department. These courses cover all aspects of the interaction design problem including psychological issues, semiotics, heuristics and use of user interface management systems (UIMS).
- **Expert Systems and AI-programming.** The main constituent of the two expert system courses we offer are use of expert system shells and knowledge acquisi-

tion methods. In these courses, the students usually use existing shells. We also provide a course in AI-programming which involves building expert system shells, deductive reasoning and truth-maintenance systems.

- **Databases**. Four courses are offered. The database course for computer science and engineering students (C- and D-line) covers theory and practice of modern database system. As well as the traditional relational databases topics such as active databases, deductive databases and object-oriented databases are also included in the course. Other courses mainly concentrate on more traditional topics, in particular design of relational databases.

7.2 Cognitive Science: A New Study Program At Linköping University

In the fall of 1994 the Faculty of Philosophy at Linköping University started a new study program in Cognitive Science. The program is a joint effort involving four different departments on three faculties: Department of Pedagogy and Psychology (IPP) and Department of Language and Literature (SoL) both from the faculty of humanities and social sciences, Department of Computer and Information Science (IDA) from the engineering faculty and the Department of Communication Studies (Tema-K) from the interdisciplinary faculty of Tema.

The goal is to train students with broad knowledge in cognition and communication between humans and computers, but also with a deep enough knowledge to allow for research studies.

The initiative for the program was taken by Nils Dahlbäck (assistant professor in cognitive science), Arne Jönsson (assistant professor in AI and former director of undergraduate studies in cognitive sciences), Professor Erik Sandewall at IDA and Roger Säljö at Tema-K. Nils and Arne invited Björn Lyxell and Stefan Samuelsson from IPP, Richard Hirsch from IDA, Hans Rossipal from SoL and Yvonne Waern from Tema-K and together they designed the study program of Cognitive Science.

The program leads to a masters degree after four years of study. It is also possible to get a bachelors degree after only three years of study. The study program starts with two years of basic courses providing a broad coverage of the methods and problem areas included in the various cognitive science disciplines.

The compulsory courses provided during the first year begin with an introductory course in cognitive science which also includes an introduction to computer science. This is a five point course (i.e. it involves work corresponding to five weeks of study). Then various subjects of Cognitive Science are provided: Cognitive Psychology (10 p), Linguistics (10 p), Lisp Programming (5 p), Logic and Discrete Mathematics (5 p), and Empirical Methodology (5 p). The second year begins with Artificial Intelligence (10 p), Humans in Group and Society (5 p), Computational Linguistics (5 p), Psycholinguistics and Discourse Theory (5 p). The two base years ends with one course in Theories in Cognitive Science (5 p) and an applied Cognitive Science project (10 p).

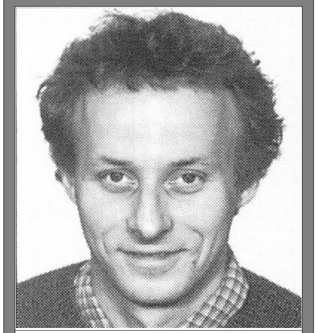
After completing the basic courses the students are given the opportunity to specialize and to achieve the necessary requirements for research studies in at least one of the research areas: 1) Computer Science, 2) Cognitive Psychology and 3) Linguistics. These subjects are strong research areas at Linköping University which will have positive effects on the quality of the education. However, the study program will be kept together during the last years of study by compulsory interdisciplinary courses.

The future labour-market for the students is mainly in areas where the cooperation between users, technical systems and organizations is important.

It is worth noting that more than half of the students in this program, despite its strong emphasis on modern information technology, are females. In the recent years the number of females applying to computer science and engineering programmes has been subject to a drastic fall. Therefore we hope that the interest in the cognitive science program persists.

System Architecture

8
Subject area: System Architecture



Johan Fagerström
director

Some of the members in the Computer Architecture Area.
Back row: Johan Boye, Niclas Andersson, Lars Strömberg, Nahid Shahmehri, Bengt Lennartsson, Zebo Peng, Mats Larsson.
Front row: Krzysztof Kuchcinski, Gunilla Blom-Lingenhult, Mariam Kamkar, Johan Fagerström.
(All teachers engaged in the undergraduate programme in Appendix C).

8.1 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 faculties and to Swedish industry.

The total number of students is approximately 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 practical work, in particular computer network and operating system courses.

Currently we give more than twenty courses divided 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 interpreters

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 discrete simulation techniques, computers in measurements and C-programming.

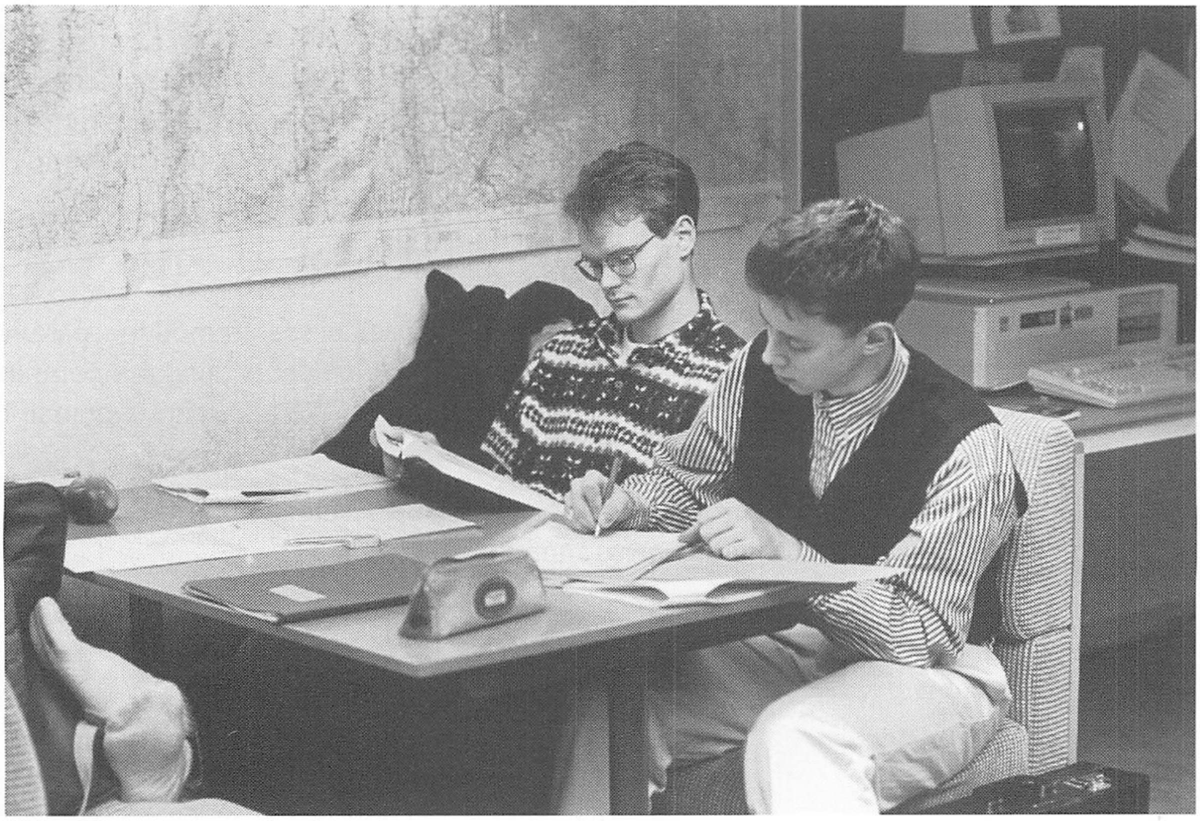
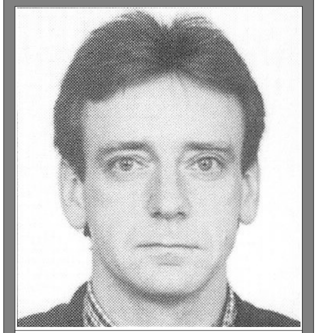


Figure 8-1. *Students at work. .*

Economic Information Systems

9 Subject area: Economic Information Systems



Jörgen Andersson
director

Some of the members in the Economic Information Systems Area.
Anders G. Nilsson, Henrik Eriksson, Jan Ollinen, Kristina Larsen, Birger Rapp, Lena Wigh, Jaime Villegas, Lars Poignant, Anna Moberg, Jörgen Andersson, Bengt Savén, Lennart Olsén.
(All teachers engaged in the undergraduate programme in Appendix C).

Figure 9-1. *A view of the University Campus.*

9.1 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 *Industrial engineering* 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.

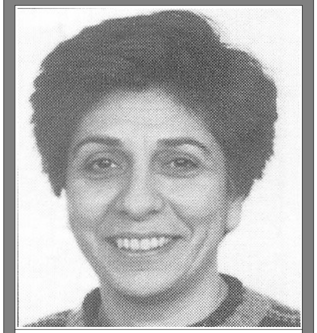
The profile can also be combined with a course in Information technology, communication, and organizational design which discusses the organizational communication and the role of advanced information technology in new organizational networks.

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.

Erasmus

10
Erasmus

(European Community Action Scheme for the Mobility of University Students)



Nahid Shahmehri
Coordinator

Pierpaolo Larocchia, Sabrina Scattchiatani, Sergio Brandani, Silvia Coradeschi.

Figure 10-1. *Other guests at the department.*
The nations represented are: Germany, The United States of America, Russia, The Peoples Republic of China, Italy, Poland, Iran, Belgium, Lithuania.

The Department is coordinating the European-based programme in the fields of Informatics and Computer Science. This is a European university network for constituting an Inter-University Cooperation Programme (ICP-94-S-3017/11). The programme currently involves the student mobility activity.

The ICP started its work in 1993 with eleven universities. The students which are nominated for this programme spend between three months and one year in one of the universities in this ICP. Students are normally assigned Erasmus grants through the Erasmus office at their home university. This grant normally covers part of their travel costs. The study program is recognized by all the universities in the ICP. The goal of the programme is to facilitate the *cultural*, *lingual*, and *scientific* exchange between the participating students and the students in the host university. The universities within this ICP arrange special language courses for Erasmus students. They also help students in finding accommodation.

For the next academic year the ICP will include the universities as below:

- ***Sweden***, Linköping University
  Institutionen för datavetenskap **(*Coordinating department)***
- ***Finland***, Oulu University
  Department of Information Processing Science
- ***France***, Université de Nice - Sophia Antipolis
  École Supérieure en Sciences Informatiques (ESSI)
- ***Germany***, Technische Universität Carolo-Wilhelmina zu Braunschweig
  Fachbereich für Mathematik Informatik und Wirtschaftswissenschaften
- ***Germany***, Universität des Saarlandes, Saarbrücken
  Fachbereich für Mathematik Informatik und Deutsches Forschungszentrum
- ***Irland***, University of Limerick
  Department of Computer Science and Information Systems
- ***Italy***, Università degli studi di Pisa
  Dipartimento di Informatica
- ***Netherlands***, Vrije Universiteit Amsterdam
  Faculteit der Wiskunde en Informatica, Vakgroep Informatica
- ***Netherlands***, Rijksuniversiteit Leiden
  Vakgroep Informatica
- ***Sweden***, Umeå University
  Institutionen för datavetenskap
- ***United Kingdom***, University of Edinburgh
  Department of Computer Science
- ***United Kingdom***, University of Manchester
  Department of Computer Science

Each university in the ICP is represented by a site-coordinator and Linköping is coordinating the whole programme. The site-coordinators have an annual meeting for exchanging experiences, discussing common questions, and planning for the future.

Linköping hosted the first and second gatherings of these ICP members. Interactions for information exchange, defining study programs for students, and other matters take place using E-mail.

Erasmus Activities in Linköping

The Erasmus office in Linköping in cooperation with the International Secretariat make a number of arrangements for all visiting Erasmus students. These include guaranteed accommodation, introductory Swedish course free of charge, and volunteered host families and mentor students.

Linköping university offers to the Erasmus students over twenty courses lectured in English, five of which given by our department. More information about the study programmes in Linköping and the facilities for visiting students can be found in the catalogue *"Courses in English"*. The visiting students can also select courses from our postgraduate course catalogue, most of which are given in English.

During the two years since this ICP has been founded Linköping hosted thirteen students from Braunschweig, Pisa, Edinburgh and Limerick.

This ICP has also been successful in sending students outwards. This program is primarily organized for students of C- and D-line (Computer Science and Computer Engineering). Students from other study programmes are nominated in case of a vacant position. There is a grant for those students who are nominated in a student mobility program.

During 93/94 and 94/95 we have sent out twelve students to Pisa, Edinburgh, Manchester, Nice, Braunschweig, Saarbrücken, and Limerick.

It is worth mentioning that on their return, our students are very happy with the choice they have made to join the Erasmus programme. They are asked to submit a brief report on their experiences and impression of their visit. One of the questions in the questionnaire they fill in is as below:

What did you appreciate most about your Erasmus-supported study period abroad?

We cite some of the responses here:

- Getting in touch with other students
- All the other students I met there
- The chance of living in another country and getting to know the students there
- The possibility to learn to speak another language
- All new friends
- See how another institution works, and the knowledge exchange
- Personal experiences, improved knowledge of another language, and international experiences

We intend to increase the amount of student mobility in both directions (inwards and outwards).

11 Graduate Programmes and Research Organization

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.

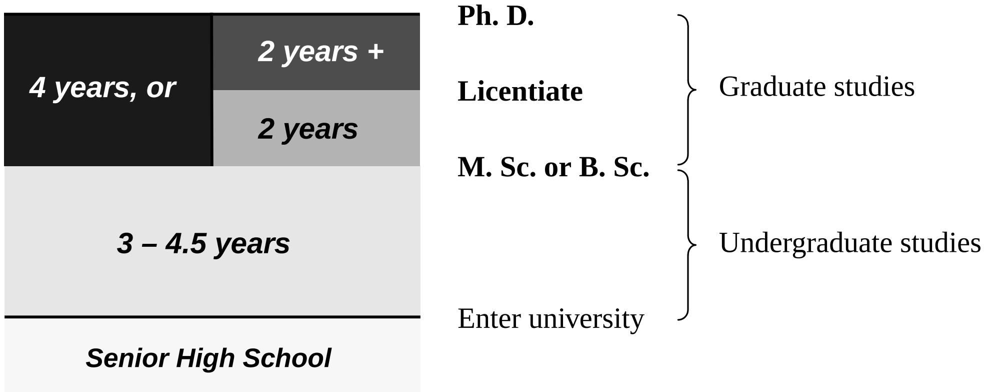


Figure 11-1. *Levels of degrees in Sweden.*

11.1 Graduate school of Computer and Information Science

Graduate studies at the department consists of courses and project participation. The course programme is organized at the department level as *regular courses*, each of which is given approximately every two or three years (if possible), and *occasional courses* which depend on the profile and interests of current faculty and visiting scientists. A thesis project is always done within or in association with one of the research laboratories.

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 40 points (one point equivalent to one week full time studies) of completed courses and 40 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 special *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.

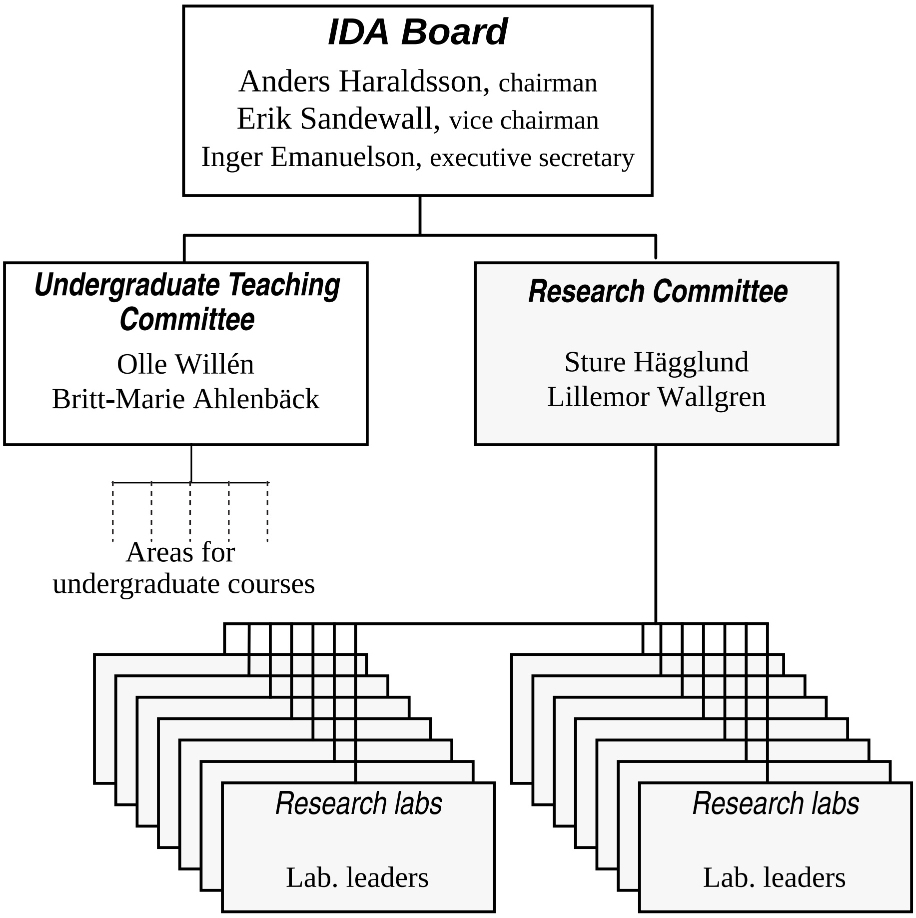


Figure 11-2. *The strategic and executive responsibilities for research projects and graduate education at IDA 1992/94.*

About 100 students participate in the graduate programme, and they may choose among about 20 courses given each year. The Research Committee, headed by Sture Hägglund, is responsible for the organization and implementation of the graduate programme. The members of the research committee are mainly senior researchers, but there are also representatives for the graduate students, and for the technical and administrative staff. As an executive, there is one director of graduate studies (forskningsstudierektor). However, most of the administration and organization rests upon the secretary of research (Lillemor Wallgren). Most graduate students are employed by the department, full time. Their responsibilities comprise of, 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 is spent on courses and thesis project.

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.

11.2 List of research laboratories

The research program has been designed to cover areas of strategic importance, both for our responsibilities in undergraduate education and for the needs in the society as envisaged by our funders. Currently, there are 14 research laboratories. A lab is charcterized by its long-term commitment to development and maintenance of knowledge within a defined area, and by its long-tem responsiblilty for graduate students.

Ashort review of research areas covered in IDA is given below.

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-intensive approaches to software development, including aspects of human-computer interaction. There are currently three subgroups dealing with cognition technology (Hägglund), knowledge engineering and software engineering (Sandahl) and usability matters (Löwgren). Special areas of interest are industrial software quality and reliability, knowledge management, usability engineering and user interface design support, expert critiquing systems, knowledge acquisition, diagrammatic reasoning and intelligent tutoring systems.

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. Currently research projects in the areas of high-level synthesis, hardware/software codesign, and design for testability are being carried out in the laboratory.

EDSLAB – Laboratory for Engineering Databases and Systems

Tore Risch

EDSLAB does research on new database services and advanced applications in particular for supporting engineering information systems. The research is centered around the umbrella project AMOS - Active Mediators Object System, a next generation object-relational database mediator between applications and data sources. The laboratory also conducts database-oriented application projects for engineering support in cooperation with Swedish industry.

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

Nahid Shahmehri

IISLAB studies theory and methods for advanced information systems, including object-orientation, computer supported cooperative work, process modelling, workflow management, information retrieval and hypertext, and description logics. 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 and Ulf Nilsson

LOGPRO long term research concentrates on the foundations of logic programming and relations to other programming paradigms and methodology. Presently focus is on query-optimization of deductive databases, proving dynamic properties of logic programs, and verification and synthesis of logic programs.

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 is engaged in research on theoretical and applied natural language processing. The theoretical research is concerned with linguistic knowledge representation from the syntactic level to the discourse level and algoritms for parsing, generation and unification. The applied research covers natural language dialogue systems and tools for machine-aided translation and document generation.

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.

RTSLAB - Real-Time Systems Laboratory

Anders Törne

RTSLAB research covers tools, methods and architectures for the design of software intensive real-time systems. In particular this includes timing analysis and design synthesis tools based on discrete modelling methods, database modelling of activities for control and simulation, and stratified real-time software architectures. The laboratory also conducts applicative research and case studies in cooperation with industry, for example in the automation and control area.

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).

11.2.1 Fields of graduate studies and degree subjects

The programme is divided into two main areas:

The *computer science field* including the degree subjects: computer science and computer systems, comprising areas of study with 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:

Computational Linguistics (*Prof. Lars Ahrenberg)*

Computer Science *(Prof. Erik Sandewall, Prof. Peter Fritzson, Prof. Sture Hägglund, Prof. Jan Maluszynski, Prof. Tore Risch)*

Computer Systems (*Acting Prof. Krzysztof Kuchcinski)*

Economic Information Systems (*Prof. Birger Rapp)*

Engineering Information Systems *(Prof. Tore Risch)*

Library and Information Science (*Prof. Sture Hägglund, Prof. Erik Sandewall)*

11.2.2 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.

Lists of courses are given in Appendix C.

11.2.3 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 are strengthened. Therefore graduate students are expected to attend invited seminars and licentiate and doctorate dissertations, disregarding research specialities.

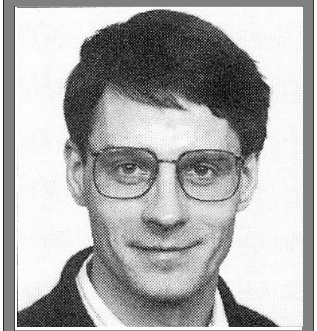
A list of seminars is given in Appendix B.

Figure 11-3. *Research Organization.*
Research secretariat *(above):*
Per-Olof Fjällström, Sture Hägglund, Lillemor Wallgren.
Research committee *(below):*
From the back: Toomas Timpka, Anders Törne, Andreas Björklind,
Anders Haraldsson, Tore Risch, Bengt Lennartsson,
Krzysztof Kuchcinski, Peter Fritzson,
Anders Aleryd, Per-Olof Fjällström, Ulf Nilsson, Sture Hägglund,
Erik Sandewall, Birgitta Franzén, Lillemor Wallgren, Nahid Shahmehri, Kristina Larsen.

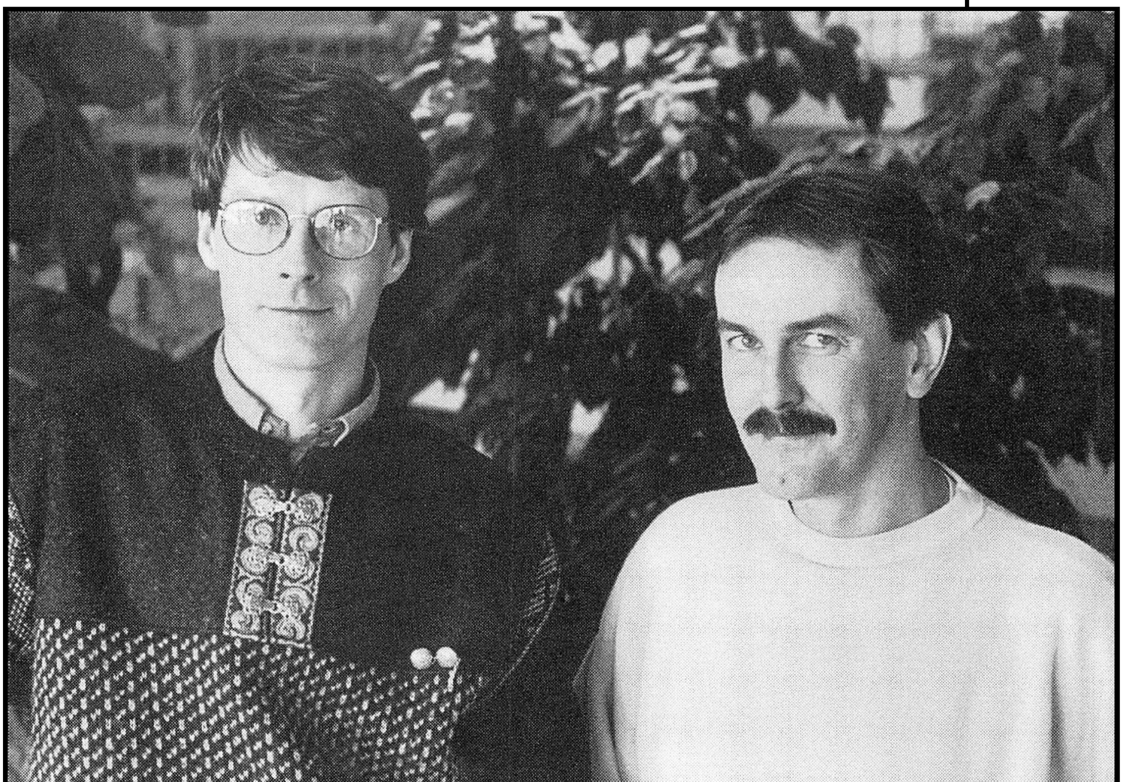
ACTLAB

12
The Laboratory for Complexity of Algorithms

Computational geometry
Analysis of algorithms
Data structures



Per-Olof Fjällström
director



The members of the Laboratory for Complexity of Algorithms.
Per-Olof Fjällström and Jan Petersson.

12.1 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.

The research in the laboratory is presently focused on computational geometry. Computational geometry 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 discipline.

The major funding for 1994 is from CENIIT (the Center for Industrial Information Technology), a special programme for interdisciplinary research within Linköping University.

12.2 Laboratory Members

Laboratory leadership and administrative staff:

Per-Olof Fjällström, Ph. D., associate professor
Madeleine Häger, secretary

Graduate students:

Jan Petersson

12.3 CENIIT project: Algorithms for Contact Searching

This project started in July 1992 and is done in collaboration with Prof. Larsgunnar Nilsson and Zhong Zhi-Hua at IKP. In applications such as computer-aided design and simulation of systems consisting of moving or deforming parts, it is necessary to accurately and efficiently determine where contacts between (and within) geometric models of physical objects occur. For example, in the simulation of a car collision a substantial part of the computation time is used to detect contact or penetration. Since such simulations may take 10-200 hours of CPU time even on high performance computers, efficient contact-searching algorithms are obviously required. The goal of the project is to develop and experimentally evaluate algorithms for contact searching.

The contact searching process basically consists of two phases. In the first phase, potential contact areas are identified. In the second phase, the potential contacts are carefully examined to determine whether or not a contact has occurred. Within ACTLAB, we have focused our attention on the first phase.

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

Early in the project we observed that detection of potential contacts can be achieved by *(orthogonal) range searching*, which is a well-known problem in computational geometry. Several sequential range-searching algorithms have been proposed but little was known about the usefulness of these algorithms in the context of contact searching. We have therefore undertaken an experimental evaluation of various range-searching algorithms; the result of this work is reported in [FP93].

As contact searching is a time-consuming process it is important to investigate to what extent parallel computers can be used to speed up the computations. Work in this direction was initiated in 1994 by an investigation of the range-searching problem on SIMD mesh-connected computers. Very little has previously been done in this area; the results of our investigation are reported in [Fjä94]. We intend to continue our work by developing contact-searching algorithms for MIMD computers.

12.4 DIMACS project: Parallel Algorithms for Geometric Dominance and Proximity Problems

Considerable efforts have been devoted to the design of parallel algorithms for combinatorial problems. However, compared to e.g. numerical applications, parallel computers are fairly rarely used in combinatorial applications. There are several reasons for this situation. For example, most algorithms are for less realistic models of parallel computation, and it is not clear to what extent such algorithms can be efficiently implemented on existing parallel computers. Also, these algorithms are usually tuned to (worst-case) asymptotic efficiency, and may not be particularly useful in practice. Thus, there is a need for experimental and implementation-oriented research.

To encourage efforts in this direction, the Center for Discrete Mathematics and Theoretical Computer Science (DIMACS, a consortium of Rutgers and Princeton Universities, AT&T Bell Laboratories, and Bellcore) has initiated a program called *The Third DIMACS International Algorithm Implementation Challenge*. During 1994, we participated in this program. Our DIMACS project, *Parallel Algorithms for Geometric Dominance and Proximity Problems*, is described below.

The need for a more practical approach is commonly recognized also in computational geometry. For the DIMACS challenge, we have therefore studied algorithms for geometric problems in a realistic model of parallel computation, the SIMD mesh-connected computer. The problems concern combinatorial properties of point sets in Euclidean space, and arise in a variety of application areas, such as statistics, operations research, pattern classification, cluster analysis, and geometric optimization. In addition to development and analysis of algorithms within a theoretical framework, the study includes implementation and experimental evaluation. To this end, we have used the MasPar MP–1 at the National Supercomputer Center (NSC), a massively parallel computer that is close to our model of computation. Our results indicate that the running time on the MP-1 are comparable to the running time on the most powerful workstations.

At the time of this writing, the DIMACS challenge is coming to an end. The project has resulted in two papers, which were presented at a workshop ([Pet94a], [Pet94b]).

We intend to continue the study of proximity problems, possibly in the context of geographic information systems.

12.5 External contacts

P-O Fjällström and Jan Petersson: Participated in the ALCOM Workshop *Algorithms: Implementation, Libraries and Use,* Saarbrücken, Germany, August 1993.

Jan Petersson: Participated in the *The Third DIMACS International Algorithm Implementation Challenge*, Rutgers University, Piscataway, NJ, October 1994.

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 since Fall 1992:

Data Structures and Graph Algorithms.
Computability and Complexity Theory.

12.6 References

Fjällström, P-O. [Fjä94]. Range Searching on a Mesh-Connected Parallel Computer. Submitted for publication.

Petersson, J. [Pet94a]. Parallel Algorithms for Geometric Dominance Problems. Presented at *The Third DIMACS Int. Algorithm Implementation Challenge,* Rutger Univ., Piscataway, NJ, Oct., 1994. Submitted for publication.

Petersson, J. [Pet94b]. A Parallel Algorithm for the All-Nearest-Neighbors Problem. Presented at *The Third DIMACS Int. Algorithm Implementation Challenge,* Rutgers Univ., Piscataway, NJ, Oct., 1994. Submitted for publication.

Fjällström, P-O., Petersson, J. [FP93]. Evaluation of Orthogonal Range-Searching Methods for Geometric Contact Searching. Presented at the ALCOM Workshop *Algorithms: Implementation, Libraries and Use,* Saarbrücken, Germany, Aug., 1993. Submitted for publication.

Fjällström, P-O. [Fjä93]. Evaluation of a Delaunay-based method for surface approximation. In *Computer-Aided Design*, 25:11, pp 711-719.

Fjällström, P-O. [Fjä91]. Polyhedral Approximation of Bivariate Functions. LiTH-IDA-R-91-24. Also in *Proc. Third Canadian Conf. on Computational Geometry,* Vancouver, Canada, Aug., 1991.

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

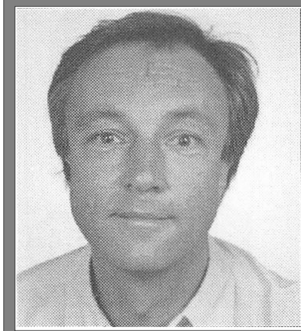
Fjällström, P-O., Katajainen, J., Levcopoulos, C., Petersson, O. [FKLP90]. A Sublogarithmic Convex Hull Algorithm. In *BIT,* 30:3, pp. 378-384, 1990.

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

ASLAB

13 The Laboratory for Application Systems

Human-computer interaction
Knowledge-based systems
Software engineering methodology



Sture Hägglund
director

Some of the members of the Application Systems Laboratory.
Back row: Mikael Lindvall, Jens Lind, Jonas Löwgren, Sverker Johansson, Johan Jenvald, Sture Hägglund, Göran Forslund, Torbjörn Näslund, Niklas Ohlsson, Mikael Eriksson.
Front row: Kristian Sandahl, Pär Carlshamre, Rita Kovordanyi, Rego Granlund, Birgitta Franzén, Eva Ragnemalm, Barbara Wasson.

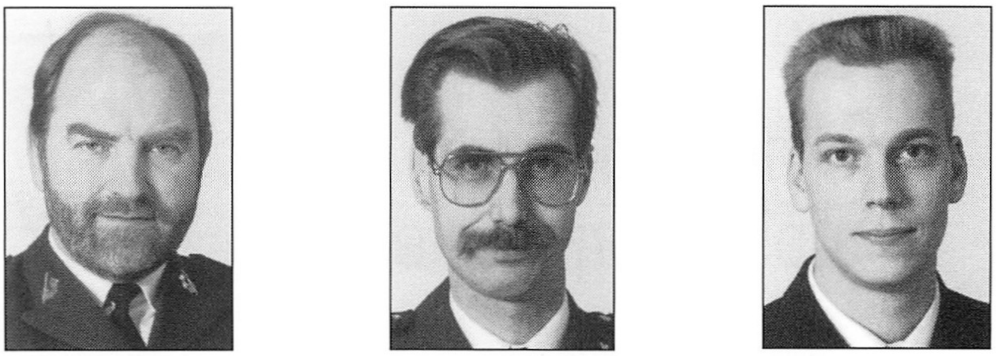


Figure 13-1. *From the left: Per-Arne Persson, Sverker Johansson and Johan Jenvald, joint graduate students with the Swedish War College in the area of Command and Control Systems.*

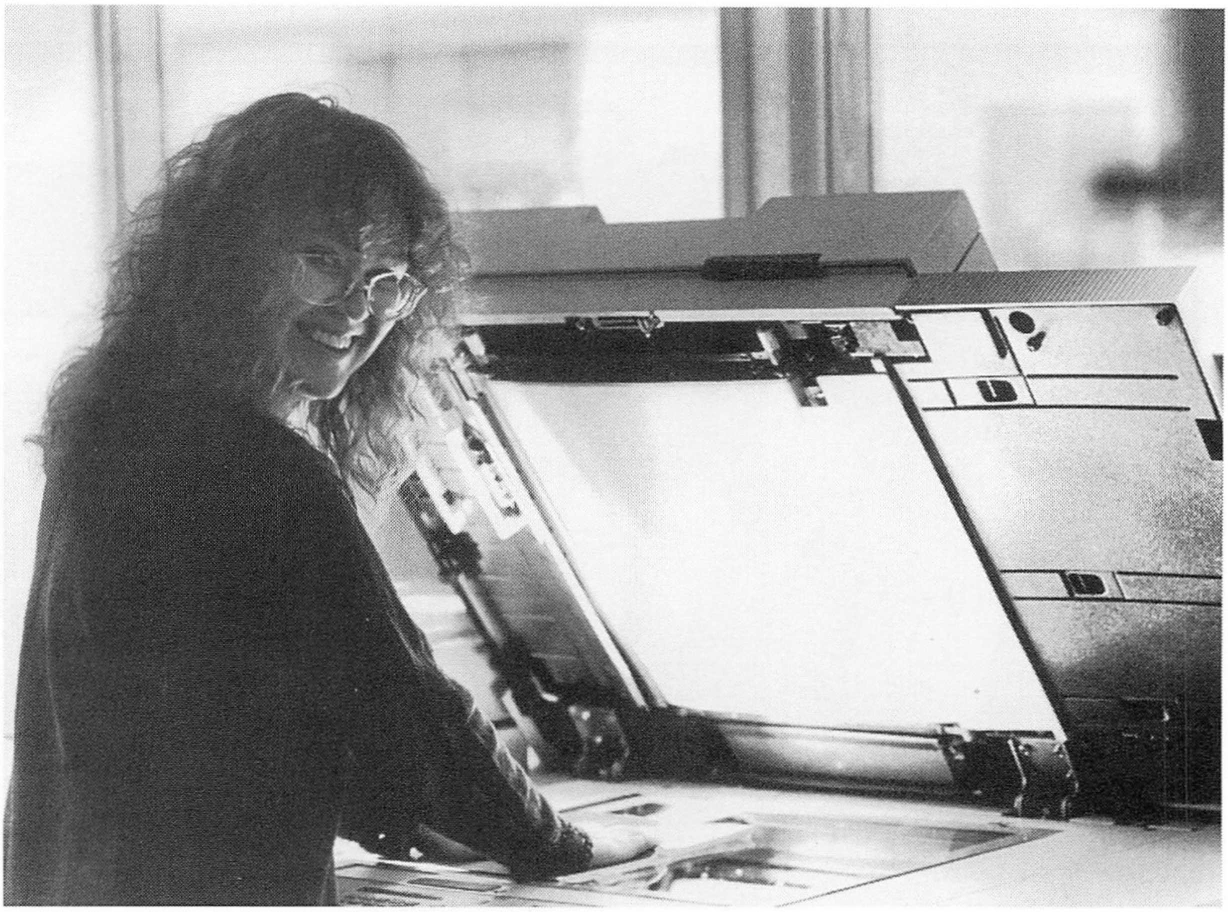


Figure 13-2. *Eva L. Ragnemalm.*

13.1 Introduction

The Application Systems Laboratory (ASLAB) was one of the first two research groups formed in the Department in 1980. Its research has, from the beginning, been oriented towards the study of theory, methods and tools, in particular knowledge-based approaches to the development and maintenance of applications software aiming at a significant increase in productivity, maintainability, understandability and user control. Current research has been broadened to a wider spectrum of approaches, but still focuses on methods and techniques for effective human-system cooperation. Projects usually take an experimental approach and emphasize participation in application-oriented projects with industry and the public sector.

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

- *Usability-Oriented Systems Development,* the *Usability Matters* group (Löwgren) focuses on studies of methods, tools and techniques for usability engineering of systems in professional practice, with an emphasis on the human-computer interface. Research projects undertaken should be governed by two goals. First, to produce scientific results about the intersection of academically established knowledge and professional environments. Secondly, the professional practice addressed shall be informed and improved by our collaboration.
- *Industrial Software and Knowledge Engineering*, the ISKE group (Sandahl), bases its work on empirical material found in industrial and governmental contexts, where complex software systems are developed. Two major projects are performed in the group, Industrial Knowledge Management and Industrial Software Engineering. On a high level, the projects differ in the sense that the research in the former project involves active development of a series of reference systems whereas the latter project uses ongoing large-scale projects as research material.
- *Cognition Technology (*Hägglund*),* includes support for collaborative decision making, including knowledge-based tools, techniques for knowledge communication and computer-supported transfer of competence. Examples of technical approaches are expert critiquing and intelligent tutoring systems. The cognition technology area is concerned with engineering knowledge and the ability to develop information processing systems based on an understanding of human cognitive processes. The area includes
  a) Computing systems based on, and dependent on, an insight and understanding of human cognitive abilities and limitations, and
  b) Development of computing systems which perform or support cognition-oriented tasks.

In addition, the laboratory hosts activities dealing with

- *Knowledge-Acquisition Tools, and Command and Control Systems* (Eriksson). These activities are closely related both to the main groups in the laboratory and to external collaboration activities.

13.2 ASLAB personnel

The following list presents the current staff and persons contributing to project activities in ASLAB during 1993–1994.

Laboratory leadership, administrative and technical staff:

Sture Hägglund, Ph. D., professor.

Birgitta Franzén, administrative assistant.
Lena Larsson, administrative assistant (1994).

Researchers and graduate students:

Brant Cheikes, Ph. D. (visiting researcher until fall 1993).
Henrik Eriksson, Ph. D. (at Stanford 1991- spring 1994).
Jonas Löwgren, Ph. D.
Kristian Sandahl, Ph. D.
Barbara Wasson, Ph. D. (visiting professor from fall 1994).

Pär Carlshamre, M. Sc., Lic.Eng.
Mikael Ericsson, M. Sc.
Göran Forslund, M. Sc.
Rego Granlund, M. Sc.
Jonni Harrius, M. Sc., Lic. Phil. (until fall 1994).
Joachim Karlsson, M. Sc., (starting fall 1993).
Rita Kovordányi, M. Sc.
Jens-Olof Lindh, M. Sc., (starting fall 1994).
Mikael Lindvall, M. Sc., Lic.Eng.
Torbjörn Näslund, B. Sc., Lic. Phil.
Niclas Ohlsson, M. Sc., (starting spring 1994).
Eva L. Ragnemalm, M. Sc.
Ivan Rankin, B.A., Lic. Phil.

Magnus Baurén, B.A., (Dept. of Communication Studies).
Johan Jenvald, M. Sc., (Swedish War College).
Sverker Johansson, M. Sc., (Swedish War College).
Svein-Ivar Lillehaug, M. Sc., (Dept. of Medical Informatics).
Per-Arne Persson, B.A., (Swedish War College).

Associated researchers

Robert Glass, Ph. D. h.c. visiting professor.
Kevin Ryan, Ph. D., visiting professor.
Iris Vessey, Ph. D., visiting professor.
Yvonne Wærn, Ph. D., professor of Communication Studies.

Other project personnel:

David Byers, M. Sc., undergraduate student.
Niklas Frost, M. Sc., undergraduate student.

Associated persons:

This list includes persons who have actively contributed to ASLAB projects, 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.)

Hans Alberg, Ericsson Telecom AB.
Marianne Almesåker, FMV.
Henrik Artman, Dept. of Communication Studies.
Bo Bergman, Professor, Quality Technology.
John Brewer, Ph. D., Pharmacia Biosystems AB.
Ann Christin Eriksson, Ericsson Telecom AB.
Lisbeth Harms, Ph. D., VTI.
Roger Hedenäng, EP Frameworks AB
Ulf Lindh, Ericsson Radio Systems AB.
Carl-Fredrik Mandenius, Professor, Pharmacia Peptide Hormones AB.
Karin Mårdsjö, Ph. D., Institute of Tema Research.
Tommy Nordqvist, Telub AB.
Östen Oskarsson, Ph. D., consultant.
Hans-Erik Pettersson, Ph. D., VTI.
Martin Rantzer, Ericsson Infocom Consultants AB.
Greger Sahlberg, Pharmacia Biosystems AB.
Rein Schandersson, VTI.
Anne Steinemann Shepherd, Ph. D., Stanford University.
Ingemar Widegren, The Defense Research Establishment.
Bengt Österlund, Ph. D., Pharmacia Biosystems AB.

13.3 The Usability Matters research group

Jonas Löwgren

The research is focused on methods, tools and techniques for usability-oriented systems development in professional practice. Every research project we undertake is governed by two goals. First, we aim at producing scientific results about the intersection of academically established knowledge and professional environments. Secondly, the professional practice we address shall be informed and improved by our collaboration.

The UM group is actively involved in the formation of a Europe-wide research consortium oriented towards design support. The consortium is coordinated by professor Manfred Tscheligi of Vienna. We are also taking part in the Amodeus Esprit research consortium - see below.

In addition to Jonas Löwgren, this group currently consists of Torbjörn Näslund, Pär Carlshamre, Mikael Ericsson, and David Byers, (temporary employment as software designer). Other researchers involved in projects within the group are Yvonne Waern and Magnus Baurén in the Department for Communication Studies.

Usability evaluation in iterative systems development.

Torbjörn Näslund

The Cocis project, which is funded by Nutek under the Information Systems programme, addresses the issue of usability inspection in iterative systems development. We have earlier demonstrated that inspection reports are easy to generate in professional development projects but rarely have expected impact on the resulting product. This phenomenon is sidestepped in the academic literature on usability evaluation. In Cocis, we have undertaken a longitudinal study of a large-scale industrial development project. Our specific intervention in the development project is to provide usability inspection reports. From a research point of view, we investigate, through qualitative methods, the wide range of factors affecting the organizational use of the usability inspection resource. The project is in the analysis stage, and we expect the results to provide a rich understanding of the complex events involved in professional iterative systems development. The results will be presented in a forthcoming Ph. D. dissertation by Näslund.

Usability engineering in collaborative systems development

Pär Carlshamre, Jonas Löwgren

In the Delta project, we address the issues of usability-oriented method takeup and collaboration between system developers and technical communicators. In collaboration with Ericsson Infocom, we developed a method extension based on usability engineering principles and involving technical communicators in the joint development process. The method extension was validated by use in a customer project at Infocom; we gathered research data from the customer project by participant-observation. Our research results indicate, among other things, that the organizational and conceptual boundary between user interface and user documentation is arbitrary and artificial. We were also able to offer empirically based modifications to the academically well established usability engineering paradigm. The Delta project was funded by Nutek under the ITYP programme and turned out to be a financial success; Infocom is currently turning the method extension into a product and Nutek will receive a considerable payoff on their research grant. The work has also resulted in a licentiate thesis by Carlshamre in 1994.

Critiquing support for user interface design

Mikael Ericsson, Jonas Löwgren, Magnus Baurén, Yvonne Waern

The Codek project, which is funded by the Council for Research into Social Science and the Humanities (HSFR), looks at knowledge-based critiquing techniques as support for user-interface design. Much research is currently devoted to inventing technical means of supporting designers using AI techniques. However, there are virtually no results on the usefulness and acceptability of such support techniques. In Codek, we are developing a Wizard-of-Oz environment in which we will conduct experiments with different approaches to design support. Our independent variables are interaction strategy and critiquing modality; we will study acceptability as well as design performance, cognitive load and learning effects. The long-term goal of the project for us is to develop support tools based on the experimental results and study their use in professional design.

Design expressions for users and systems

Jonas Löwgren, David Byers

Finally, we participate (together with Toomas Timpka and Cecilia Sjöberg in the MDA Laboratory) in the Esprit based research action Amodeus, which is coordinated by professor Phil Barnard of Cambridge and comprises twelve research sites in seven European countries. In Amodeus, our task is to investigate technical means for integrating argumentative design notations. We have developed a theoretical framework and a number of conceptual prototypes. Our main activity of this year is to develop a platform for integrative design tools, including a coherent information representation and reasoning mechanisms. Our participation in Amodeus is funded from the CEC.

13.4 Industrial Software and Knowledge Engineering

Kristian Sandahl

The members of the group devoted to Industrial Software and Knowledge Engineering (ISKE) all base their work on empirical material found in industrial and governmental contexts, where complex software systems are developed. The reason for joining the areas of software engineering and knowledge engineering is the prospect of combining the strengths of both fields, which even today are widely developed in parallel with a prejudiced unwillingness of exchanging ideas. Two major projects are performed in the group, Industrial Knowledge Management and Industrial Software Engineering. On a high level, the projects differ in the sense that the research in the former project involves active development of a series of reference systems whereas the latter project uses ongoing large-scale projects as research material.

13.4.1 Industrial Knowledge Management

Kristian Sandahl, Göran Forslund, Jens-Olof Lindh, Jonni Harrius (-93/94)

The main assumption behind the research is the widely recognized fact that all organizations in industry and government are critically dependent on their ability to store, distribute and utilize the human knowledge being produced and consumed within their business domains. We call the intentional implementation of this ability Knowledge Management. Our intention is to provide knowledge about how software supporting Knowledge Management can be realized in applications where neither traditional expert-system technology nor information retrieval systems are capable of simultaneously providing the flexibility and range of knowledge needed for a heterogeneous group of end-users.

The long-term research strategy is to initiate research projects about applying Knowledge Management technology in various reference domains with external partners. By generalizing over this first-hand experience and the second-hand experience from commercial systems being subsequently developed by our partners, we are able to propose several design hypotheses, which give an empirically founded basis for more focused research. Examples of such results are the Active Expert methodology (reported in Sandahl's doctoral dissertation 1992), the knowledge-linker tool architecture (reported in Henrik Eriksson's doctoral dissertation 1991) and a knowledge representation allowing flexible utilization of human decision (to be reported in Forslund's licentiate thesis scheduled for Spring 1995).

In total the group has an individual experience base of ten implementation projects in various domains in both academic and commercial settings. Our current efforts include:

- **Decision support for the planning of local traffic safety**. Lindh is currently engaged in a three-year project together with the Swedish Road and Transportation Research Institute (VTI) with the goal of finding a systematic way of reasoning with information of different quality.
- **Protein purification planning**. We are currently following up three commercial systems developed by Pharmacia Biotech AB, our major research partner during 1988-1991. The goal is to provide knowledge about maintenance of complex knowledge-bases.
- **Support for design of biochemical processes**. Harrius performed initial studies in Spring 1994 and we are now defining a long-range research project together with Pharmacia Peptide Hormones AB.

13.4.2 Industrial Software Engineering

Kristian Sandahl, Mikael Lindvall, Joachim Karlsson, Niclas Ohlsson

Existing formal theories are too narrow in scope and too small in scale to substantially contribute to industrial production of software, software engineering. Industrial practitioners are also faced with a massive flow of new methods marketed with impressive claims supported with weak if any empirical evidence.

Our goal is to find empirically founded improvements of methods, tools and techniques currently used for quality assurance and cost reduction for industry software. The research progresses in four steps:

1. observation and documentation of current industry projects
2. explaining and refining found relationships amongst entities of process, product and resources
3. experimenting enhancements
4. modifying methods, tools and techniques

We have managed to establish an interest for this research at two branches of Ericsson (ERA, ETX), SAAB and FMV. Currently, three major projects are performed:

- **Software Maintainability.** An ERA-project in the area of mobile telecommunication performed with an object-oriented method has been followed with the goal of assessing the benefits and cost of explicitly handling traces between entities of domain analysis, requirements specification, design specification and code. The hypothesis is that tool enhancement with regard to traceability will contribute to more effective maintenance of the product. This will be the focus of continued research as the product, shipped in June 1994, is being subjected to maintenance activities. (*Lindvall*)

- **Software Reliability.** Based on various studies of AXE-projects at ETX, a joint research project with IDA and IKP Quality Technology is aiming at identifying the most critical factors of the development process affecting the reliability of the resulting product. A detailed quantitative study of the functional design at module level has been completed, and we are about to collect data from approximately five larger projects through questionnaires. We also intend to perform a participatory observation of a planned project of reasonable size in order to support the interpretation of the data collected. The project is sponsored by ITM through a consortium consisting of ETX, SAAB Military Aircraft and FMV. (*Ohlsson*)
- **Requirements Engineering.** Finding and fixing defects originating from requirements specifications are extremely costly. Our research objective is to minimize the number of defects in requirements specifications by prioritizing the customers' requirements. An ERA-project using the Quality Function Deployment (QFD) method, where one activity is to prioritize the customers requirements, is followed. An evaluation of that method by checking the result with a customer panel is to be carried out during the Spring -95. (*Karlsson*)

The project also has two external graduate students, Roger Hedenäng at EP Frameworks (Metrics for reuse of Object-Oriented Software) and Marianne Almesåker at FMV (Reliability assurance of safety-critical software). International cooperation includes Robert Glass, Computing Trends, Victor Basili, University of Maryland, Norman Fenton, City University London, Kevin Ryan, Limerick University and Alan Goerner, University of Missouri.

13.5 Cognition Technology group

Sture Hägglund

The cognition technology group primarily studies knowledge-based approaches to computer-supported decision making and related issues of knowledge transfer, tutoring and training. In particular we are interested in expert critiquing approaches, where the supporting argumentation for a proposed decision is as important as the proposal itself. An area of growing interest is the application of this kind of understanding for tutoring and training purposes. In the area of knowledge-based systems, we cooperate closely with the Industrial Software and Knowledge Engineering group.

The group consists of Rego Granlund, simulation technology for collaborative process management, Rita Kovordányi, creative spatial problem solving, Eva Ragnemalm, intelligent tutoring and training, Ivan Rankin, expert critiquing systems and Barbara Wasson, Ph. D., visiting professor, intelligent tutoring. Associated with the group is Svein-Ivar Lillehaug, also a graduate student in the medical informatics department.

Expert critiquing

Ivan Rankin, Jonni Harrius, Sture Hägglund

In an Expert Critiquing System (ECS), 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 an 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.

An appealing characteristic of expert critiquing systems is that they can be employed in a wide variety of situations with a broad range of supportive functions. They can, for example, complement problem-solving systems with or without full (i.e. global) knowledge of the problem space by being able to comment on particular sections of the problem/solution at hand. They can also function in situations where no (near-)optimal solution can be given at all and still be able to provide useful comments.

The interesting kind of critiques are those that require more than individual comments to partial solutions, i.e. where a multi-sentential text is an obvious form of communication from system to user. Text generation has been a hot research topic for over a decade, though generalizable results are still slow in coming: research is often described in terms of a particular system or as a set of linguistic phenomena the generator covers.

Ivan Rankin's work in this area discuss the kinds of *communicative strategies* a text generator is intended to provide. For example, what does an ECS do when it runs into conflict with the user over a user-given proposal? Should it concede to a proficient user or should it argue its case? What should an ECS do if it finds itself in agreement with the user's plan of action? Should it simply agree or try and be more cooperative by making suggestions as to how to achieve the goals the user and system consider best? Jonni Harrius presented a knowledge representation model and a system architecture for expert critiquing in his licentiate thesis (1993).

Simulation environments for collaborative process management.

Rego Granlund, Sture Hägglund

In this project we study how to model and support decision making in command and control applications. The idea is to start from a description of the pedagogic goals for a training session and to provide automated tools for scenario generation and in particular for simulating agents needed in order to create a realistic communication with the environment. This work is done in cooperation with, among others, The Defense Research Establishment (Ingemar Widegren) and Department of Communications Studies (Yvonne Waern, Henrik Artman).

In a first study, we are looking into the problem of supporting or mechanizing the actors that provide the environment for the decision makers to be trained (Granlund). For this purpose, we are modelling the actors both in terms of the roles they play in support of a simulation game and in terms of their qualifications for acting in a specific role. We may for instance have more or less experienced people available and we may also consider automation or semiautomation of a given actor.

A special problem of interest is the distribution of roles over one or more individuals. In a training situation, one person can be assigned to play several roles and in this case the system should support the user in the management of the knowledge states of each actor in the system. A related, and even more interesting problem, is to support real decision makers in situations where a crisis is rapidly developing and the responsibility for controlling and commanding is changing from one to several decision makers and vice versa. The system should then support the distribution of information and responsibilities to the people involved in an intelligent and efficient way.

In this area, we are also engaged in an international CSCW research programme, COST 14 CoTech, studying computer-supported collaborative process management. This research is concerned with group decision making in the context of a real-time process or situation.

Spatial problem solving through artificial imagery.

Rita Kovordányi

In our work with knowledge-based systems we have become aware of the limitations of traditional approaches based on pure symbolic reasoning. The incapability of contemporary AI systems to simulate complex human behavior can, according to a cognitive scientific point of view, be attributed to the symbolic formalisms used and the somewhat ad hoc architectures underlying these systems. Humans show a natural tendency to draw or image diagrams when faced with a difficult problem. As imagery is a powerful tool in human creative cognition, it should also be emulated in artificial problem solving systems. The long-term objective of this project is to design a human-like interactive problem solver, which will be working with spatial (diagrammatic) representations.

A necessary step in reaching this goal is the construction of an algorithmically adequate model of human vision and imagery; a model which has its underpinnings in cognitive neuroscience. In particular, the mechanisms underlying the interpretation and reinterpretation of mental images has to be mapped out. Current research is focused on understanding and modeling the role of selective attention in this context.

Intelligent tutoring systems.

Eva Ragnemalm, Barbara Wasson, Brant Cheikes

In a study initiated a few years ago in the context of Nutek's programme for process control applications (DUP), we began to investigate how to use simulations to train process-control operators. The aim was to improve the resulting understanding of the involved processes by extending the traditional mathematical simulation models with more experience-based knowledge (Ragnemalm, Cheikes). Of particular interest was the possibility of relating 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.

These studies led us to the topic of student modelling in intelligent tutoring systems (Ragnemalm). This project is concerned with the problem of modelling the knowledge state of a student in computer-based training of diagnosis skills in process control applications. For this purpose we study the use of a Learning Companion, which is a simulated partner interacting with the trainee during the training session. The ambition is to model and guide the development of a student from novice to expert during a session with a training simulator. Ragnemalm has carried out experiments in order to understand both the kind of communication that takes place during such a training session and the feasibility of designing a computerized Learning Companion. This work will be presented in a licentiate thesis in April 1995.

Together with the Medical Informatics Department, Svein-Ivar Lillehaug is carrying out a project with the goal to extend the design of, and implement an intelligent learning environment for ventilator therapy based on cognitive apprenticeship. This involves investigating the integration and utilization of various approaches and techniques such as: representation of domain and pedagogical knowledge, knowledge-based simulation, critiquing expert systems, instructional planning and user modelling. From a user's perspective the application can be described as an open-ended exploration in an interactive simulation, with an intelligent tutoring environment that provides support such as diagnostic feedback, explanation and coaching, and where problems are sequenced according to the learner's evolving mental model.

The group also includes Barbara Wasson, who received her Ph.D in computational science from the University of Saskatchewan, Canada, in 1990. The dissertation focused on content planning for intelligent tutoring systems. Her research interests lie in the area of instructional learning environments and teaching strategies - in particular, the combination of techniques from artificial intelligence with an explicit representation of pedagogical knowledge. Currently she is on faculty at the University of Bergen, Norway and is visiting ASLAB for the better part of three years, starting in 1994.

13.6 Other activities

13.6.1 Tool Support for Knowledge Acquisition

Henrik Eriksson, Niklas Frost

Developers can use tools that support the design of domain-oriented knowledge acquisition tools to create environments in which domain specialists can enter and edit knowledge bases. Such *metatools* take as input high-level definitions of knowledge acquisition tools, and produce as output descriptions of the target 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 tools, and that such specialized knowledge acquisition tools can be developed at low cost.

As part of his thesis work, Dr. Eriksson developed a metatool, DOTS, that supports the generation of knowledge acquisition tools from an architectural description. 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 1991–94, Dr. Eriksson has been a visiting researcher at the Knowledge Systems Laboratory (KSL), Section on Medical Informatics at Stanford University, Stanford, California. He participated in an ongoing research project, PROTÉGÉ-II, at the KSL. The PROTÉGÉ-II project aims at developing computer-based tools that support software developers in the design of knowledge-based systems. In this approach, the development of a knowledge-based system is divided into (1) the design of a problem solver that accomplishes domain tasks, and (2) the definition of a knowledge base that represents the expertise required by the problem solver for various inferences in the problem-solving process. In PROTÉGÉ-II, problem solvers are composed from primitive problem-solving methods, which can be reused in several projects.

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 metatool support for the design of these domain-oriented knowledge acquisition tools. Dr. Eriksson developed the metatool DASH, which is part of the PROTÉGÉ-II suite of tools. DASH takes as input a domain ontology and produces as output a textual specification of the target tool, which can be used by the Meditor knowledge acquisition tool run-time system.

Currently, ASLAB collaborates with the PROTÉGÉ-II group on the development of new tools. As part of his project work, Niklas Frost has developed a critiquing tool that analyses the tool generated by PROTÉGÉ-II, and highlights shortcomings and inconsistencies in the tool and its specification (i.e., the input ontology). In addition to research on knowledge acquisition and knowledge acquisition tools, Dr. Eriksson is working on development methodologies, in particular methods for requirement engineering. This work is primarily carried out in the context of the MDA group.

13.6.2 Decision support in C^{3}I systems

Johan Jenvald, Sverker Johansson, Per-Arne Persson, Sture Hägglund

A special activity is undertaken in the area command and control information systems, in particular for military applications. Three graduate students from the Swedish War College are currently active in Ph. D. projects. These are Johan Jenvald, M. Sc., information fusion and presentation, Sverker Johansson, M. Sc., intelligent planning in C^{3}I systems and Per-Arne Persson, B.A., cooperation and coordination in C^{3}I systems. This research is carried out in cooperation with other groups in our department (such as RKLLAB in the area of AI planning techniques and LIBLAB in the area of information science and cooperation technology). Another contact is with the Defense Research Establishment (FOA), where part of the project work is carried out. In a joint project with FOA's department for human-system interaction where Sture Hägglund is employed part-time as adjunct researcher, we study the design of C^{3}I systems as well as the use of simulation for training of decision making in military staffs.

13.7 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 in joint projects:

- Ericsson Radio Systems in the area of software engineering methodology.
- Ericsson Telecom, Saab Defense and FMV in the area of software reliability.
- Pharmacia, Uppsala and Stockholm. Cooperation on knowledge-based systems in support of experiment planning in bio-technology.
- The National Defence Research Establishment (FOA) in the areas of cognitive engineering, simulation in command and control systems and techniques for evaluation of system designs.
- Ericsson Infocom Consultants in the area of methods for integrated design of human-computer dialogues and user handbooks.
- The Road Traffic Institute (VTI) in the area of knowledge-based systems for promoting safety in road traffic systems.
- CelsiusTech and Telub in the area of usability engineering in systems development.

In joint projects we cooperate with researchers in other departments, for instance the Tema Research Department (Waern, Mårdsjö et al.), Quality Technology (Bergman, Helander et al.), Medical Informatics (Wigertz et al.) and Applied Physics (Mandeinius et al.). Within our department, cooperation is also extensive, e.g. with CAELAB, LIBLAB, NLPLAB and MDA where joint projects are carried out.

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 Maryland, University of Missouri, University of Colorado at Boulder, George Washington University, University of Oregon and Pennsylvania State University. In Europe we have cooperation with e.g. City University, London, University of Karlsruhe, Xerox EuroPARC and Univ. of Limerick, Ireland. We also participate in a European COST CoTech project on Computer Supported Cooperative Work and in the ESPRIT project AMODEUS 2.

Our research in the software engineering 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 and Victor Basili for shorter visits. In this connection, several external courses and seminars (software tutorials, SOFT) have been arranged.

13.8 Publications

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

Ph. D. dissertations 1993-94:

Zhang, T. (1993). *A Study in Diagnosis Using Classification and Defaults. Systems with an Active Expert Methodology.* Ph. D. thesis, Linköping University.

Licentiate theses 1993-94:

Harrius, J. (1993). *An architecture and a knowledge representation model for expert critiquing systems*. Licentiate thesis No. 383, Linköping University.

Carlshamre, P. (1994). *A collaborative approach to usability engineering: Technical communicators and system developers in usability-oriented systems development*. Licentiate thesis Thesis No. 455, Linköping University.

Lindvall, M. (1994). *A study of traceability in object-oriented systems development*. Licentiate thesis No. 462, Linköping University.

International publications 1993-94:

Carlshamre, P. (1993). Looking back on my first year as a maintainer. *The Software Practitioner*, 3(6):10–13.

Carlshamre, P. (1994). Technical communicators and system developers collaborating in usability-oriented systems development: A case study. In *Proc. 1994 ACM 12th Annual Int. Conf. on Systems Documentation (SIGDOC'94)*, pp. 200–207, New York, NY, ACM.

Carlshamre, P., Löwgren, J., Rantzer, M. (1994). Usability meets the real world: A case study of usability-oriented method development in industrial software production. In Bradley, G. E. and Hendrick, H. W., eds., *Proc. 4th Int. Symp. Human Factors in Organization Design and Management*, pp. 427–432, NORTH-HOLLAND.

Cheikes, B. (1993). Methodological issues in the design of intelligent and cooperative information systems. In *Proc. of the Int. Conf. on Intelligent and Cooperative Information Systems (ICICIS-93)*, pp. 3-12, Rotterdam, IEEE Computer Society Press.

Cheikes, B., Ragnemalm, E. L. (1993). Simulator-based training-support tools for process-control operators. In *Proc. of the NATO Workshop on Natural Dialogue and Interactive Student Modeling*, Varenna, Italy.

Eriksson, H. (1993). Specification and generation of custom-tailored knowledge-acquisition tools. In *Proc. IJCAI'93*, Chambéry, France.

Eriksson, H. (1994). Models for knowledge-acquisition tool design. *Knowledge Acquisition*, 6(1):47–74.

Eriksson, H. (1994). A survey of knowledge acquisition techniques and tools and their relationship to software engineering. *Journal of Systems and Software*, 19(1):97–107.

Eriksson, H., Musen, M. A. (1993a). Conceptual models for automatic generation of knowledge-acquisition tools. *Knowledge Engineering Review*, 8(1):27–47.

Eriksson, H., Musen, M. A. (1993b). Metatools for software development and knowledge acquisition. *IEEE Software*, 10(3):23–29.

Eriksson, H., Puerta, A. R., Musen, M. A. (1994). Generation of knowledge-acquisition tools from domain ontologies. *Int. Journal of Human–Computer Studies*, 41(3):425–453.

Eriksson, H., Shahar, Y., Tu, S. W., Puerta, A. R., Musen, M. A. (In press). Task modeling with reusable problem-solving methods. *Artificial Intelligence*.

Forslund, G. (1994). Cooperative or autonomous agents: Considerations for knowledge acquisition and representation. *IEEE Expert*. Special issue on Cooperative Systems for Workgroups.

Forslund, G. (1995). Designing for flexibility: A case study. *Expert Systems*, 12(1):27–37.

Granlund, R. (1994). Developing models for describing actors in a distributed problem solving situation. In Langemyr, L., eds., *Proc. of the SIMS Simulation Conf.*, pp. 238–243.

Hägglund, S. (1993a). A framework for expert critiquing. In Hägglund, S., ed., *Working notes from 11th National Conf. on Artificial Intelligence (AAAI'93) Workshop on Expert Critiquing Systems*, Washington, DC, USA.

Hägglund, S. (1993b). Introducing expert critiquing systems. *The Knowledge Engineering Review*, 8(4). Guest Editor's Introduction.

Hägglund, S. (1994). On the need and support for user-centred design. In *Proc. 4th Int. Symp. Human Factors in Organization Design and Management (ODAM'94)*, Stockholm, Sweden.

Hägglund, S., Granlund, R. (1994). Computer support for collaborative c3i decision making and training. *SIGOIS Bulletin*, 15(2):5–7. Presented at the Workshop on Collaborative Realtime Process Management at CSCW'94.

Harrius, J. (1993). From problem solving to solution critiquing by reusing an existing knowledge base. In *Proc. of SCAI*, Stockholm, Sweden, May, 1993.

Kovordányi, R. (1994). Attention as an inhibiting force in 'creative' reinterpretation of ambiguous images. In *Proc. of the First Int. Summer Institute in Cognitive Science*.

Lindvall, M. (1994a). Object-oriented maintenance. In *Object-Oriented Programming - Issues and Actions*. Working notes from workshop arranged by Oregon Center for Advanced Technology Education, Feb. 28 - Mar 1, 1994.

Lindvall, M. (1994b). Object-oriented modeling in practice. *The Software Practitioner*, 4(2).

Löwgren, J. (1993a). *Human-computer interaction: What every system developer should know.* Studentlitteratur, Lund.

Löwgren, J. (1993b). Using domain knowledge to support graphical editing. In Salvendy, G. and Smith, M., eds., *Proc. Fifth Int. Conf. on Human-Computer Interaction*, Amsterdam. Elsevier.

Löwgren, J. (1994). Empirical foundations for design rationale as user-interface design support. In Opperman, R., Bagnara, S., and Benyon, D., eds., *Proc of the 7th European Conf. Cognitive Ergonomics (ECCE 7)*, pp. 305–310, St. Augustin, Germany, GMD, GMD-Studien Nr. 233.

Löwgren, J., Holmberg, L. (1993). Customizing the SIGCHI curriculum for use in Sweden. *SIGCHI Bulletin*, 25(4):8–11.

Löwgren, J., Laurén, U. (1993). Supporting the use of guidelines and style guides in professional user-interface design. *Interacting with Computers*, 5(4):385–396.

Löwgren, J., Quinn, C., Gasen, J., Gorny, P. (1994). Designing the teaching of HCI: Report from a CHI'94 workshop. *SIGCHI Bulletin*, 26(4):28–31.

Nyce, J. and Löwgren, JM. (In press). Towards foundational analysis in HCI. In Thomas, P., ed., *Social and Interactional Dimensions of Human-Computer Interfaces*. Cambridge University Press.

Näslund, T. (1994a). Supporting design communication with explicit representation of evaluation feedback. *Scand. Journal of Information Systems*, 6(2):21–42.

Näslund, T. (1994b). "Usability is extremely important - but it's somebody else's job, I hope". In Kerola, P., Juustila, A., and Järvinen, J., eds., *Proc. of the 17th Information systems research seminar in Scand. (IRIS17)*, University of Oulu, Finland.

Ohlsson, N. (1994a). Monitoring software projects. In *Proc. of Systems Engineering of Computer Based Systems*, IEEE Press. Position Paper. To appear.

Padgham, L., Löwgren, J. (1993). A user interface management approach for object-oriented database applications. *Journal of Systems and Software*, 27(3).

Ragnemalm, E. L. (1994c). Simulator-based training using a learning companion. In Brouwer-Janse, M. and Harrington, T., eds, *Human-Machine Communication for Educational Systems Design*, Vol. 129, pp. 207–212, Springer-Verlag. Proc. from the 1993 NATO-ASI in Eindhoven.

Rankin, I. (1993a). Generating argumentative discourse in expert in critiquing systems. In Hägglund, S., ed., *Working notes from AAAI'93 Workshop on Expert Critiquing Systems*, Washington, DC.

Rankin, I. (1993b). Natural language generation in critiquing. *Knowledge Engineering Review*, 8(4).

Rankin, I., Harrius, J. (1993). Critique generation - language games, argumentation and rhetorical aggregates. In Ahrenberg, L., eds., Papers from the *3rd Nordic Conf. on Text Comprehension in Man and Machine*, Linköping, Sweden.

Sandahl, K. (1993). Developing knowledge-based systems with active expert participation. In *Proc. of 5th Int. Conf. on Software Engineering and Knowledge Engineering, SEKE-93*, pp. 589–595.

Sandahl, K. (1994). Transferring knowledge from active expert to end-user environment. *Knowledge Acquisition*, (6):1–22.

Sandahl, K., Eriksson, H., Österlund, B. (1994). Tool-based knowledge management and distribution of knowledge to customer's end user in a biochemical domain. In *Proc. of the Int. Symp. on the Management of Industrial and Corporate Knowledge*, pp. 161–175, Compeigne, France.

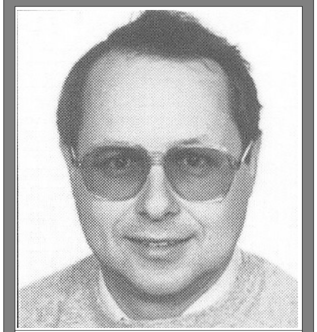
Thomas, P., Löwgren, J. (1994). From theory to action: Teaching research and practice in technology. In *Proc. of the 11th Int. Conf. Technology and Education (ICTE'94)*, London.

Wærn, Y., Hägglund, S., Ramberg, R., Rankin, I., Harrius, J. (In Press). Computational advice and explanations - Behavioural and computational aspects. In *Proc. of INTERACT'95*, Lillehammer.

Wærn, Y., Hägglund, S. (1994). Communication and cognitive science research in Linköping. In *Conf. Companion, CHI'94*.

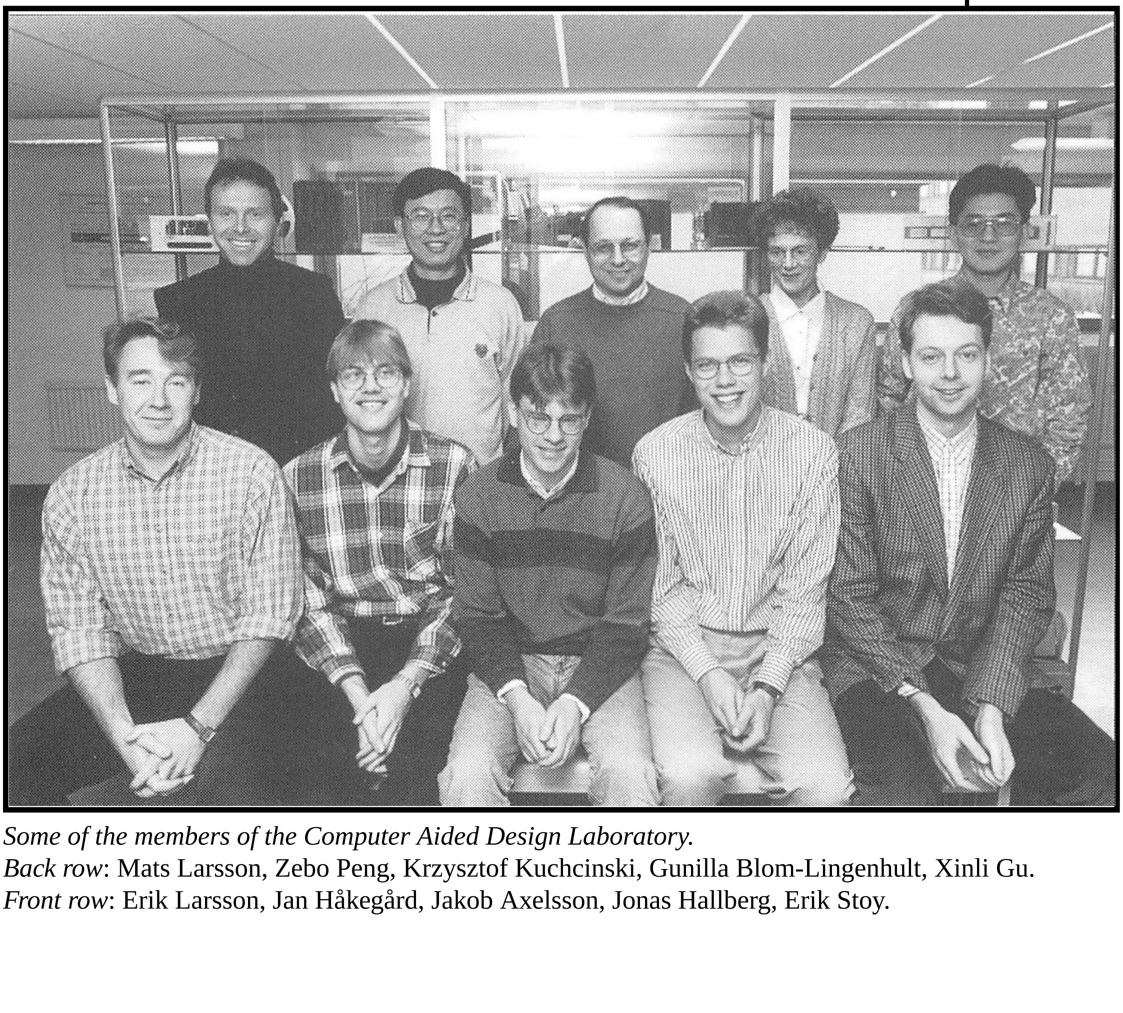
CADLAB

14 The Laboratory for Computer-Aided Design of Digital Systems



Krzysztof Kuchcinski
director

Computer-aided design
Digital systems synthesis and analysis
Hardware/software co-design
Test synthesis



Some of the members of the Computer Aided Design Laboratory.
Back row: Mats Larsson, Zebo Peng, Krzysztof Kuchcinski, Gunilla Blom-Lingenhult, Xinli Gu.
Front row: Erik Larsson, Jan Håkegård, Jakob Axelsson, Jonas Hallberg, Erik Stoy.

14.1 Introduction

The laboratory for Computer Aided Design of Digital Systems, CADLAB, is concerned with the different aspects of design automation of digital systems, especially those involving very large scale integrated circuits (VLSI). The major effort of our research work concentrates on the different aspects of digital system specification, design, test, simulation, optimization, partitioning, synthesis and related formal methods. Our primary goal is to develop methodologies and tools which will be used in a long term perspective and have a significant impact on design automation of digital systems.

To achieve this long term goal we have created a vision of a future design environment for embedded digital systems consisting of hardware and software components. The specification of such systems is given as a system level behavioral description in an imperative language such as VHDL or ADA. The system description is then translated into an internal representation, which is analysed and evaluated. Based on the evaluation results the specified system will be partitioned into hardware and software domains. Later the software part is translated using commercial compilers while the hardware is synthesized using high-level/logic synthesis tools. During the system's life time it is possible to change partitioning and generate new implementations. The testing and maintenance problem, very much neglected in previous research, also plays an important role in our framework. We consider testing as an important mechanism to increase reliability of digital systems and we are developing designs for testability methods to incorporate test considerations in the design process.

Several research projects have been started to develop different parts of our design environment. Currently the following projects are being carried out at CADLAB: "Advanced Systems Design with ASIC"- a consortium of academia and industry, "Architectural Synthesis of Digital Systems", "Hardware/Software Co-design" and "System Design for Testability". All projects are carried out in close cooperation with Swedish industry.

14.2 Laboratory Members and Guests

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

Laboratory leadership and administrative staff:

Krzysztof Kuchcinski, Ph. D., acting professor, laboratory director
Gunilla Blom-Lingenhult, laboratory secretary

Senior researchers:

Zebo Peng, Ph. D., associate professor

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

Graduate students:

Jakob Axelsson, M. Sc.
Xinli Gu, Tech. Lic.
Jonas Hallberg, M. Sc.
Jan Håkegård, M. Sc.
Erik Larsson, M. Sc.
Mats Larsson, Tech. Lic.
Erik Stoy, M. Sc.

Visitors during extended periods:

Petru Eles, Ph. D., Technical University of Timisoara, Romania
Marius Minea, Technical University of Timisoara, Romania
Alexa Doboli, M. Sc., Technical University of Timisoara, Romania

The following degrees were awarded to CADLAB members since the last report:

Xinli Gu completed his Licentiate thesis entitled *Testability Analysis and Improvement in High-Level Synthesis Systems*.

Mats Larsson completed his Licentiate thesis entitled *A Transformational Approach to Formal Digital Systems Design*.

A total of about 15 technical papers were published in international journals and international conferences during the reported time. A paper entitled *Synthesis of VHDL Concurrent Processes* by Petru Eles, Krzysztof Kuchcinski, Zebo Peng and Marius Minea got the “Best Paper Award” at EURO-DAC with EURO-VHDL 1994 conference.

14.3 Current Research

CADLAB’s research activities during 1993-1994 have been split into the following overlapping and interacting main areas:

- high-level synthesis,
- hardware/software co-design,
- test synthesis, and
- formal hardware verification.

We are developing both basic methodologies and tools in these research areas. To demonstrate the research results, prototype implementations of some design environments are also developed. The design environment for integrated hardware/software design is given in Figure 14-1. In this design environment the input to the system is given in the form of VHDL behavioral specifications which are compiled into an internal design representation. This representation constitutes a kernel for the design tools and methodologies we have developed. It is used for synthesis, hardware/software co-design and for testability analysis and improvement. It is also used in the formal hardware verification project.

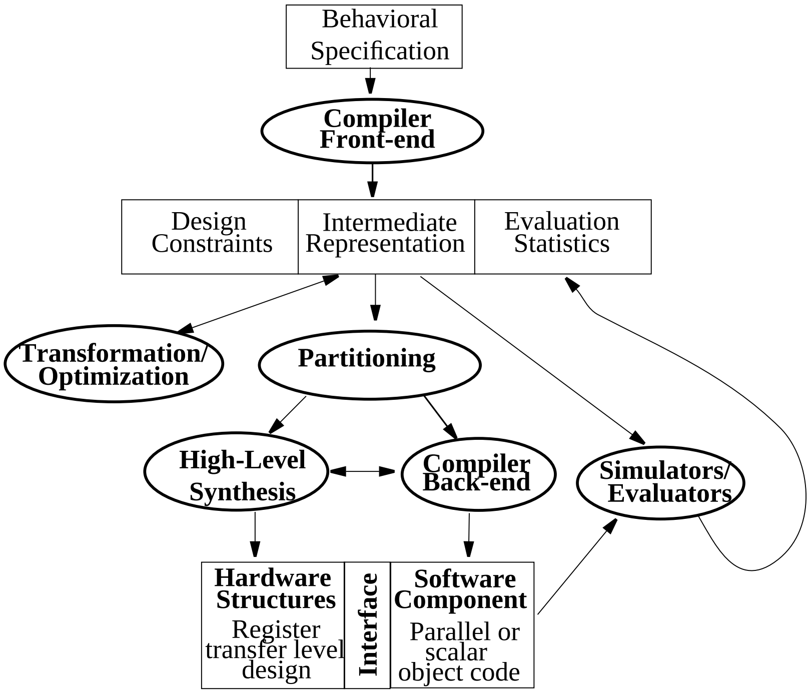


Figure 14-1. *Overview of the proposed design environment.*

The following sections give a more detailed presentation of the different projects and their main achievements during the reported time.

14.3.1 High-Level Synthesis

The objective of the research is to extend current high-level synthesis methods, which are usually limited to scheduling, allocation and module binding, to more general architectural space exploration methods. The main emphasis is put on process level concurrency and timing constraints specifications and synthesis. A large subset of VHDL is used as an input specification language and the design representation based on Petri nets captures the internal information of a design. Telecommunication examples are used for evaluation of results.

14.3.1.1 Synthesis of Concurrent VHDL Processes

Petru Eles, Krzysztof Kuchcinski, Zebo Peng

This research addresses the problem of high-level synthesis from a behavioral VHDL description that contains interacting concurrent processes [8]. Our goal is to conform to the VHDL standard semantics during synthesis and to produce hardware that operates with a high degree of parallelism. One of the most difficult issues in this context originates from the VHDL semantics of signal assignments and wait statements which are specified in terms of simulation. As stated in the language definition, signals are only updated at the end of a simulation cycle, unlike variables which are updated as soon as they are assigned a value. This means that the update of signal values must be synchronized with the execution of a wait statement by *every* process in the system and has to be performed simultaneously for all signals that change their values in that simulation cycle.

The synthesis strategies we propose preserve the partial ordering relation of operations on signals and ports from the simulation model to the synthesized hardware structure. Thus, we are achieving *simulation/synthesis correspondence* which means that both the simulation model and the synthesized hardware react with the same values (sequences of values) of the signals and ports to identical sequences of stimuli applied at the inputs.

The main objective of our approach is to preserve the computational effects of the simulation cycle with minimal additional costs and minimal impact on the performance of the synthesized hardware. To achieve this goal we have developed a method for compiling VHDL into an internal design representation which explicitly captures its essential semantics with respect to process synchronization. The developed compiler automatically generates synthesis structures which are later transformed by high-level synthesis algorithms. An example of a VHDL process and the related Petri net based internal representation is shown in Figure 14-2.

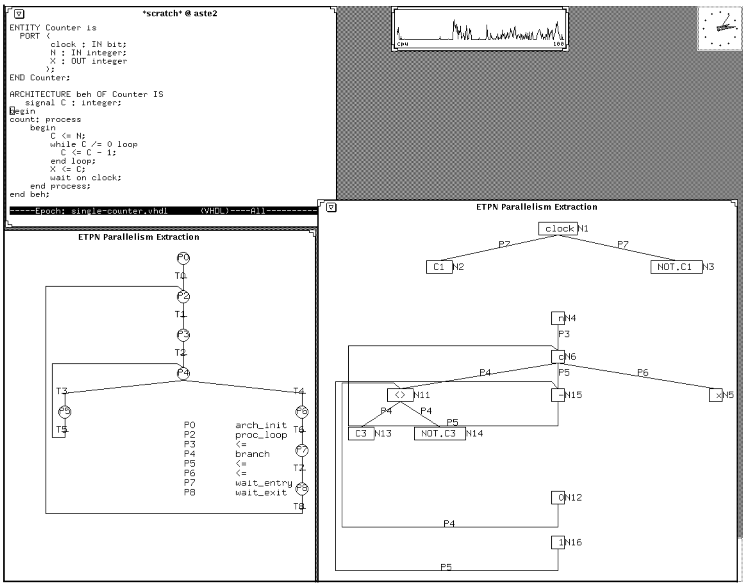


Figure 14-2. *An example of the VHDL process and its design representation.*

Our approach supports two different solutions to concurrent process synthesis, one with unrestricted use of signals and wait statements, and the other with reduced synchronization between processes. Which solution will be applied for a certain synthesis task depends on the description style adopted by the user. Selection of the description style is decided according to the features of the designed hardware. The proposed solutions have been implemented, and tested with the CAMAD high-level synthesis system.

14.3.1.2 Synthesis under Timing Constraints

Jonas Hallberg, Krzysztof Kuchcinski

High-level synthesis systems perform two main tasks: resource allocation and operation scheduling. According to this approach the time schedule of operations in the final implementation is decided as a result of optimization decisions made during the synthesis. Hence, a high level algorithmic description accepted for HLS very often contains no timing information and decisions concerning planing of operations are passed to the synthesis tool. Some design requirements can, however, impose particular restrictions on the execution time of an operation or a sequence of operations in the algorithmic description, on the execution frequency of an input/output signal, on the timing properties of interface signals, etc.

Requirements on the timing aspects of the design are incorporated as *timing constraints* in the behavioral VHDL specification submitted to the synthesis system. These constraints have to be captured as part of the internal design representation generated after compilation of the input specification. One of the major problems of the definition and implementation of a mechanism for timing constraints specification in the context of VHDL synthesis is consistency between the behavior of the simulation model and the synthesized hardware. We have identified solutions to this problem within the frame of our two synthesis strategies and have shown how simulation/synthesis correspondence can be achieved for VHDL specifications containing both interacting processes and timing constraints.

We have proposed and implemented nested minimal/maximal timing constraints in the VHDL language and its front-end compiler. We have also developed a method for back-annotation of synthesized timing into the VHDL description for post-synthesis simulation. Algorithms for consistency checking of timing constraints as well as synthesis algorithms are under development.

14.3.2 Hardware/Software Co-Design

The main objective of this research is to develop methods and tools for the description, evaluation and partitioning of application-specific computer systems which consist of both hardware and software components. The starting point for this research is the observation that high-level synthesis has reduced the gap between software and hardware development. It opens up possibilities to start both hardware and software development from the same specification which can then be partitioned and synthesized into hardware and software implementations. Our current research has concentrated mainly on the issues of intermediate design representation and hardware/software partitioning. Cooperation with Dr. Petru Eles' group from the Technical University of Timisoara in the area of using VHDL for hardware/software co-specification and development of pre-partitioning techniques for VHDL specifications has also been carried out [11].

14.3.2.1 Intermediate Hardware/Software Design Representation

Erik Stoy, Zebo Peng

Our basic approach to the hardware/software co-design problem is to first develop a *uniform* way of modeling a design in the intermediate design stages so that hardware/software trade-offs can be easily made and quickly evaluated. Based on this modeling technique, design evaluation procedures and design space exploration methods can be developed as well as techniques to allow efficient movement of functionality from hardware to software or vice versa. The design representation model therefore plays a central role in our design methodology.

Formulation of the basic modeling technique has been done. It is basically an extension of the Petri net based hardware design representation model developed by our group. This has shortened the development time and it also provides a link to existing hardware synthesis tools. Implementation of the basic design representation model and its semantics in Prolog has been done. The Prolog implementation is used as a verification of the correctness of the formalism used for the modeling technique. It will eventually be used also as a simulator. Development of evaluation procedures of hardware/software systems based on the representation model has also been started, which will be the main activities of this project in the near future.

14.3.2.2 Hardware/Software Partitioning

Jakob Axelsson, Zebo Peng, Krzysztof Kuchcinski

The hardware/software partitioning problem consists of finding a way to assign different parts of the program to different parts of the physical hardware. The criteria that guides the partitioning is different for different kind of applications, and we therefore concentrate on partitioning of embedded real-time systems, which is a particularly interesting application class for hardware/software co-design. The goal of the implementation of a real-time system is to assure that certain timing constraints, normally expressed as deadlines on tasks, are met to the largest extent possible, and this is also what we take as the goal of the partitioning. To be able to calculate if a task will meet its deadlines, we must take into account that different tasks might share physical resources, such as microprocessors or buses, and how conflicts over those resources are resolved by a scheduler. We have adopted a fixed-priority scheduling policy, for which we have also developed an efficient and general priority assignment algorithm. For partitioning, we are studying branch-and-bound algorithms. Some problems that will be addressed in the near future are how to analyze the execution time of the program in different implementations, and how to select an appropriate target architecture.

14.3.3 Test Synthesis

The project aims at the development of design for testability (DFT) methodologies and tools to facilitate testing of complex digital systems consisting of heterogeneous components. We are also developing a system view of design for testability so that problems related to testing in the production, operation and maintenance phases can be dealt with in a systematic way and solved by formal and efficient methods. The project concentrates on research issues in the areas of systems DFT architecture, computer-aided testing and synthesis for testability.

14.3.3.1 Testability Analysis and Improvement

Xinli Gu, Krzysztof Kuchcinski, Zebo Peng

The objective of this research is to develop methods and tools to detect testing problems in the early design stages and make appropriate changes in a design to make it more testable. In particular, we have been working on the development of testability analysis tools at register-transfer level. Our testability measurement is defined by controllability and observability of each node in the data path of ETPN. They are measured based on the structure (feedbacks, branches, fan-in and fan-out etc.) of a design, the depth (both sequential and combinational) from I/O ports and the testability characteristics of functional units used. They reflect the ATPG complexity and test application time for achieving high testing quality [1, 7].

A testability analysis algorithm has been developed and implemented for calculating the controllability and observability of all nodes. The results are used to guide the improvement of a design's testability. We have mainly employed partial scan and partitioning techniques to improve design testability. When partial scan technique is used, a sequential unit selection strategy is used to identify hard-to-test registers and a line selection strategy is used to identify hard-to-test lines. The registers identified are then transformed to scan registers and are connected to a scan path. For hard-to-test lines, we insert test modules to these lines and connect them to the scan path also to make them testable.

Testability analysis results are also used together with a set of heuristics to guide the selection of partitioning boundary to improve testability. The selected registers and test modules at partitioning boundary are replaced by BIST registers when BIST technique is used and/or replaced by scan registers when scan technique is used.

Experimental results with high-level benchmarks show that our partial scan scheme improves the fault coverage and ATPG time. The hardware overhead used for scan and test application cycles is less than for a full scan scheme. In several cases, the partial scan scheme achieves even better fault coverage than full scan, since the extra added logic does not improve fault coverage, but increases the test generation complexity instead. When designs are partitioned for testability, the fault coverage achieved in most cases is better than partial scan and full scan.

14.3.3.2 Board Level Testing

Jan Håkegård, Zebo Peng

The main purpose of this research is to simplify board level testing, by providing built in test support on each board. This is done through the inclusion of a test controller which autonomously performs testing of the board after having received higher level test commands. Such a test controller will not only reduce the need to use bed-of-nails fixtures in the production test phase, but can also be used during the operation and maintenance phase to perform regular self tests for localization of an erroneous board in a complete system.

A pre-study in the area has been made through the development of a simple test controller model in VHDL. The experience from this work will be utilized in the development of algorithms to detect what tests should be made at the board level, as well as for the automatic generation of a test controller targeted for the analyzed board.

The project is running in close cooperation with Ericsson Telecom AB. Ericsson not only support us in technical discussions and realistic real world examples, but also actively participates in the project through their own expert in the area, Gunnar Carlsson.

14.3.3.3 High-Level Synthesis for Testability

Zebo Peng

The main objective is to develop a new approach to integrate testability consideration into high-level synthesis. The approach is based on a stepwise refinement technique for high-level synthesis which utilizes a sequence of design transformations to generate a register-transfer level design from a VHDL behavioral specification. Three main techniques have been developed to improve testability during the design transformation process: controllability/observability balance allocation, partial scan, and condition scan. Since the application of these testability-improvement transformations is carried out together with operation scheduling, data path allocation, and control allocation, the testability factor is taken into account in the global optimization process and more testable designs are generated [12].

One of the main tasks of high-level synthesis is data path allocation, which determines the number and type of hardware modules to be included in the physical implementation and the mapping of data path operations into available hardware modules. In our approach, we start with a default allocation generated by a VHDL compiler, which assumes that each operation instance in the VHDL specification is mapped into an individual data path node. Re-allocation is then carried out basically through merger transformations, each of which folds two combinatorial nodes into one, representing the decision to share a hardware module by two operations.

When testability is not considered, selection of data path nodes to be merged are usually based on analysis of connectivity/closeness between the nodes. This often results in difficult-to-test designs because it generates many loops, and nodes with good controllability and bad observability are usually merged together, since they are close to the primary inputs. Nodes with good observability and bad controllability are also merged together for similar reason. As a result, the data path consists of many nodes which are difficult to control or to observe. In our approach, testability analysis results are used to guide the selection of merged nodes. The basic technique is to fold nodes having good controllability and bad observability to nodes having good observability and bad controllability. In this way, the new node will inherit the good controllability from one of the old nodes and the good observability from the other. This technique is called controllability/observability balance allocation, since the objective is to make the controllability and observability of the data path nodes relatively equal. Refinement of this technique is being done at the moment.

14.3.4 Formal Hardware Verification

Mats Larsson, Krzysztof Kuchcinski

High-level synthesis (HLS) systems transforms a behavioural specification into a register transfer level implementation. The final design is correct based on the assumption about correctness of all transformations and the optimization algorithm as well as all software involved in this process. Unfortunately, existing HLS systems are not based on sound semantic principles. Research in this area has so far been oriented towards optimizing a qualitative measure of the generated design in terms of area and speed and the equally important correctness measure have almost entirely been ignored.

The goal of the verification project is to remedy this lack of concern for correctness by improving the formal foundations of high-level synthesis. More specifically we want to prove the correctness of transformation rules used in an existing HLS system and also provide a formal basis for extending the set of available transformations in a sound way. A further goal is to examine ways in which proof systems can play a role in the architectural synthesis process. To achieve these goals the approach taken here is to embed the internal design representation of an existing HLS tool in a mechanized proof system [6].

By embedding a language in a proof system we mean to (re)define it in terms of the notation of the proof system. We have chosen to use the HOL proof assistant as proof system which means that we will embed our design representation in higher order logic. The motivation for using HOL is that it is a mature and well supported proof system with properties such as safe symbolic reasoning about design objects, safe extensions to the basic logic via definitions, and a programmable interface to the logic. Another reason it that we have already designed a formal transformation system based on the window inference package in HOL[2, 9] and will build on that experience in this project.

To try out these ideas we will use the ETPN design representation used in the CAMAD system [3]. ETPN is well suited for this purpose since it has a well defined execution model and the synthesis approach is purely transformational. The main difference between this and the original approach is that using the original definition of ETPN we can only capture an algorithm and optimize it under the syntactic rules that are necessary to guarantee that we maintain the meaning of the algorithm. These syntactic rules imply among other things that we must keep the data dependencies between operations. Using a formal definition of ETPN semantics we allow a designer to modify the algorithm so as to yield a more efficient implementation. This is possible since we can reason about the meaning of designs and not just about their form. This means that a transformation in a design is possible as long as it preserves the mathematical equivalence.

14.4 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 1993 the course “Introduction to Petri nets” was given by Krzysztof

Kuchcinski and Zebo Peng. The course “VLSI System Design” was given by Krzysztof Kuchcinski and Zebo Peng during autumn 1993 and winter 1994. Additionally, a series of research seminars on digital systems design automation and hardware/software specification methods were given every week by members of CADLAB.

The group established a solid cooperation with Swedish industry. In the consortium on “Advanced System Design with ASIC” we actively cooperate with Ericsson MEST/CADLAB group led by Lars-Olov Eriksson, and with Ingemar Söderquist, SAAB Missiles. These contacts are also maintained in other industry oriented projects.

In terms of international cooperation we have established a stable research project on VHDL synthesis in cooperation with Technical University of Timisoara.

14.5 References

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

Gu, X. *Testability Analysis and Improvement in High-Level Synthesis Systems*, Licentiate thesis No. 378, Department of Computer and Information Science, Linköping University, September 1992.

M. Larsson. *A Transformational Approach to Formal Digital System Design*, Licentiate thesis No. 378, Department of Computer and Information Science, Linköping University, May 1993.

Peng, Z. Kuchcinski, K. Automated Transformation of Algorithms into Register-Transfer Level Implementation, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 13, No. 2, Feb. 1994, pp. 150-166.

Stoy, E. Peng, Z. An Integrated Modelling Technique for Hardware/Software Systems. *Int. Symp. on Circuits and Systems*, London, England, 1994.

Stoy, E. Peng, Z. A Design Representation for Hardware/Software Co-Synthesis. *Euromicro 94*, Liverpool, England, 1994.

Larsson, M. (1994b). Towards Amalgamating High-Level Synthesis and Proof Systems. *Microprocessing and Microprogramming*, Vol. 40, No. 10-12. Presented as a Short Note on the *EUROMICRO Conf.*, Liverpool, England, 1994.

Gu, X, Kuchcinski, K, Peng, Z. Testability Analysis and Improvement from VHDL Behavioral Specifications. *EURO-DAC’94*, Sept. 19-23, 1994, Grenoble, France.

Eles, P., Kuchcinski, K., Peng, Z., Minea, M. Synthesis of VHDL Concurrent Processes. *EURO-DAC’94*, Sep. 19-23, 1994, Grenoble, France.

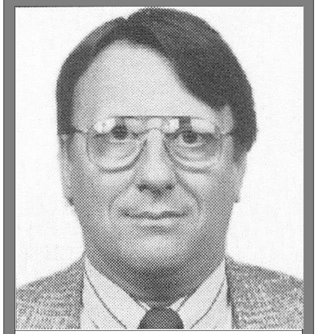
M. Larsson. An Engineering Approach to Formal Digital System Design. In T. F. Melham and J. Camilleri, eds., *Higher Order Logic Theorem Proving and Its Applications*, Vol. 859 of LNCS, pp. 300-315. Springer-Verlag, Sep. 1994.

Peng, Z. Kuchcinski, K. An Algorithm for Partitioning of Application Specific Circuits, *The European Conf. on Design Automation, EDAC-EUROASIC 1993*, Feb., 22-25, Paris, France.

Eles, P., Peng, Z., Doboli, A. “VHDL system-level specification and partitioning in a hardware/software co-synthesis environment,” *Proc. 3rd Int. Workshop on Hardware/Software Co-Design*, Grenoble, France, Sep. 22-25, 1994.

Peng, **Z**. “Testability-Driven High-Level Synthesis”, *Proc. of the 1994 Int. Conf. on ASIC*, 1994.

EDSLAB



Tore Risch
director

15 The Laboratory for Engineering Databases and Systems

Engineering Database Tools and Applications
Domain-Specialized Databases
Active, Temporal, and Real-Time Databases
Object-Relational Query Processing
Multi-Databases
Parallel and Main-Memory Databases

Some of the members of the Laboratory for Engineering Database Systems.
Staffan Flodin, Tore Risch, Gustav Fahl, Ling Lin, Martin Sköld, Magnus Werner, Anne Eskilsson, Olof Johansson and Kjell Orsborn.
Missing: Jonas S. Karlsson and Tomas Padron-McCarthy

15.1 Summary

The laboratory for Engineering Databases and Systems was established in July 1994 and is an offspring from the former CAELAB founded in July 1989. The work in the laboratory is mainly research on new database services and advanced applications in particular for supporting engineering information systems. Important concepts are distribution, heterogeneity, active databases, temporal databases, and databases in real time systems. Applicative projects are also conducted in cooperation with industry: One focuses on product modelling systems and one focuses on applying next generation database technology to computational mechanics.

15.2 The laboratory research

The field of Engineering Database Technology includes research on methods and theories for database support of engineering applications. Applications for this technology include mechanical, electronic, and software applications. Central problem areas are:

- Methods for data modelling of complex engineering systems.
- *Domain-oriented* database technology for scientific and engineering applications.
- Methods for efficient *querying* of object-oriented databases through *OO Query Languages*.
- Methods for *integration* of both decentralized autonomous databases and other external data (e.g. text, graphics, geometry objects, sensors, etc.).
- *Active databases* that notify applications when critical state changes in the database occur.
- *Real-time databases* that include real-time properties.

A large part of the research involves the umbrella project AMOS - Active Mediators Object System. AMOS serves as un umbrella project where some subprojects contribute parts of a general vision. Theoretical results are verified by incorporation in the AMOS architecture and prototypes.

15.2.1 Laboratory members

Laboratory leadership and technical/administrative staff:

Tore Risch, Ph. D., professor
Anne Eskilsson, secretary

Research staff:

Gustav Fahl, Lic. Ph.
Staffan Flodin, M. Sc.
Olof Johansson, Lic. Eng.
Jonas Karlsson, M. Sc.
Ling Lin, M. Sc.

Kjell Orsborn, Lic. Eng.
Thomas Padron-McCarthy, M. Sc.
Martin Sköld, Lic. Ph.
Magnus Werner, M. Sc.

Associated researchers:

Sture Hägglund, Ph. D., professor, LiTH
Lennart Karlsson, Ph. D., professor, LuTH
Witold Litwin, Ph. D., professor, Paris Univ. 9
Per Svensson, Ph. D., res. Director, Swedish Defense Research Establishment
Bo Torstenfelt, Ph. D., associate professor, LiTH
Anders Törne, Ph. D., associate professor, LiTH

15.3 Projects

This section presents the projects in which EDSLAB is participating. The projects are externally financed by Nutek, TFR, and by the local faculty organization CENIIT (Center for Industrial Information Technology). CENIIT provides basic funding for the graduate students and research supervision within the laboratory.

15.3.1 AMOS - an architecture for active mediators

The AMOS (Active Mediators Object System) project addresses support for future engineering information systems where autonomous, heterogeneous, and active databases and other software are distributed over fast computer networks. In such an environment *active mediators* simplify the communication between individual programs (usually being run on a workstation) and the *data sources* from which information is retrieved. The purpose of these active mediators is to locate, transform, combine, query and monitor the desired information, and therefore retain flexibility and convenience for the user in very large federations of databases and other systems. Our approach is called *active mediators*, since it includes *active* database facilities. The research is aimed towards the development of an architecture and prototype system for active mediators.

A central component of AMOS is an object-oriented (OO) query language, AMOSQL, with object-oriented abstractions and declarative queries. The language is extensible to allow for easy integration with other systems. This allows for knowledge, now hidden within application programs as local data structures, to be extracted and stored in AMOS modules.

Different application areas require specialized representation methods. AMOS also allows building specialized *domain mediators* that represent application oriented models with domain specific database representations and operators. In this way the database can be used to represent not only external data, but also domain knowledge that is presently hidden within each application. Domain mediators include product models for various engineering design disciplines, analysis and synthesis models for computational mechanics, and user interfaces for engineering applications. The AMOS system provides key technology to design domain mediators for engineering applications including: high-level modeling capabilities in an extensible object-oriented query language - AMOSQL, application-oriented data management and high execution efficiency.

A major reason for this approach is that it blends easily with the older technology: active mediators can serve as high level interfaces to existing, large and conventional databases, and so they can be introduced gradually. We foresee that as large data systems evolve by the introduction of active mediators it should become gradually easier to introduce new services into the system.

Figure 15-1 illustrates how a set of application programs access a set of data sources through AMOS. 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 sensory data source and notify (showed by arrows) applications or mediators. In the following subsections the work on each kind of mediator is described in more detail.

The initial work on the AMOS prototype is built upon a main-memory object-oriented DBMS, WS-IRIS, developed (by Tore Risch) at Hewlett-Packard Laboratories, Palo Alto, CA.

The basic work on AMOS is funded by NUTEK in the *Information Systems* programme (Sec. 15.3.2, 15.3.4, 15.3.7), by TFR (Sec. 15.3.3), and by CENIIT.

15.3.2 Object-Oriented Query Processing

Staffan Flodin

A central issue in the database area is the processing of queries. The object-oriented data model imposes new demands on the query processing capabilities of a database management system. Several such issues have been identified and novel query optimization techniques to solve them are being developed. Special attention is made towards support for queries for scientific applications and for the future OO SQL standard.

One of these new demands is the ability of the query processor to handle *late bound* function calls efficiently. A function call is late bound when the actual implementation of the function is not selected until run time. Late bound function calls are necessary if the data model includes both inheritance and overloading[1] of function names.

A method to manage late bound function calls in queries has been developed and is implemented in AMOS. The method includes resolution of the function calls required to be late bound and an efficient and optimizable execution strategy. Efficient processing of OO queries requires the ability to manage inverted late bound function calls, and the ability to manage inverted late bound function calls is a very important feature of our method.

For high efficiency the query language compiler does as much type checking at compile time as possible and thus reduces the type checking needed at run time.

[1]Overloading of function names is having the same name to denote several implementations, where the actual implementation to use is selected based on the type of the arguments.

Incremental query compilation techniques are developed that automatically recompile query language functions when the database schema evolves.

Future work will include generalizations of our method of managing late bound function calls and other optimization techniques of OO queries.

15.3.3 Integrators

Gustav Fahl

Different data sources store data in different ways. For example, they may use different data models, data formats, etc. The integrators combine heterogeneous data and present a higher OO abstraction, an *OO view*, of integrated data. Gustav Fahl is working on techniques for providing AMOS views of relational database sources.

Key problems with such integration are: how to provide object identity in the view when there is no such concept in the relational database; how to represent relational database access in AMOS query plans; how to handle the fact that the extension of types in the view depends on the state of the relational database; how to map relational structures to subtype/supertype hierarchies in the view; and how to optimize queries against the object view.

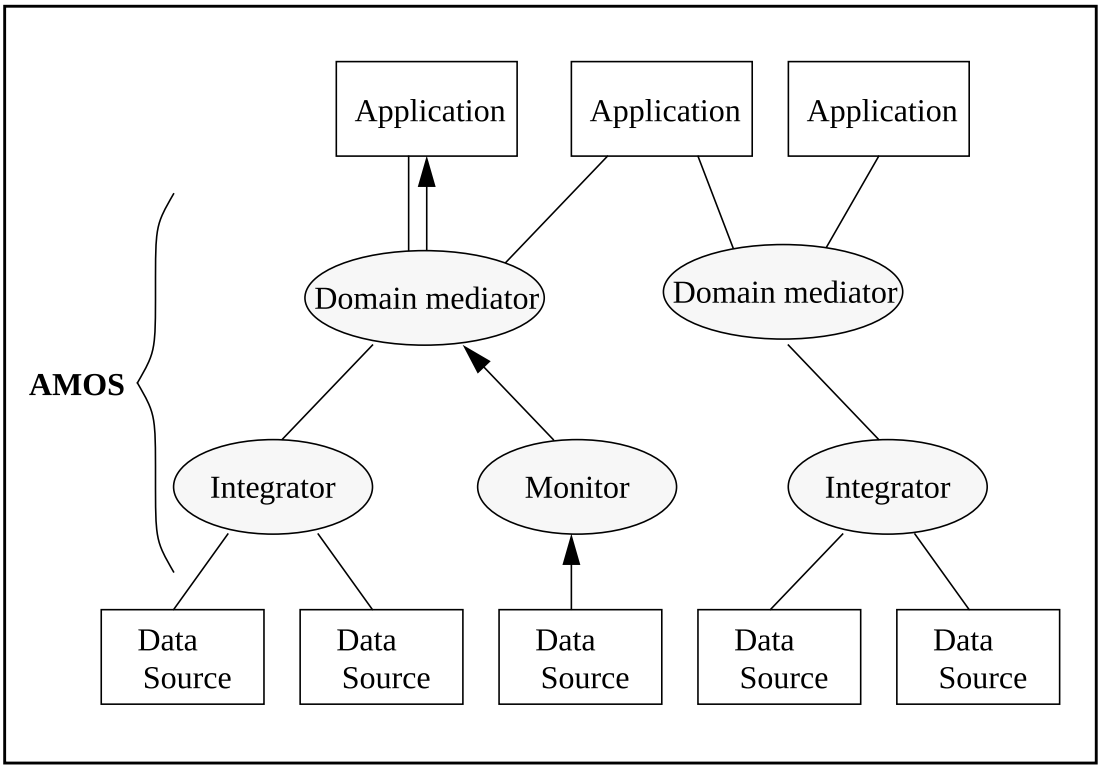


Figure 15-1. *Active mediators of different classes mediating between data sources and users/applications*

15.3.4 Active Databases

Martin Sköld

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

The query language of AMOS, AMOSQL, has been extended with active rules. Rules are defined by pairs <condition, action>, where each condition is a declarative AMOSQL query and where each action is an AMOSQL 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 overloaded on types, 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.

Since the full power of AMOSQL queries is available in rule conditions it is possible to state very complicated conditions that relate many objects in the database. To support this, a calculus for incremental evaluation of rule conditions has been developed. It enables efficient monitoring of complicated rule conditions if the number of updates to the database is few.

We are presently extending the AMOSQL rules to monitor any kind of change in the database. This includes schema changes, time, and changes to external sensors that are modelled in the database.

Extending AMOS with time and capabilities to monitor external sensors will be correlated with the work on Spatio-Temporal Databases and Real-Time Databases presented in the next sections.

15.3.5 Spatio-Temporal Databases

Ling Lin

Conventional databases were designed to capture the most recent data, as new data become available through updates, the existing data were removed from the database. For some applications, it is necessary to maintain past, present and future state of data. Temporal databases support the storage and querying of information that varies over time. Our goal is to investigate how to extend AMOS with query constructs and index structures to support applications that require temporal data, e.g. real-time control applications.

As a driving example we are initially investigating how to represent in AMOS a database model for decision support for Air Traffic Flow Management (ATFM). The example is derived from an application by S.M. Sripada et al. presented at our ADB-94 conference in June 1994. We have developed a running database model that can store spatial data like sectors, beacons, cross-points, etc. and flight data like routes, speeds, scheduled taking-off time and landing time, etc. A radar continuously feeds the database time stamped spatial positions of aeroplanes. The database model allows to ask queries on past, current and future states of flights, e.g.

- What is the current speed of flight f?
- What was the flight position of flight f at time t?
- When did flight f pass over some specific space position?
- What will be the estimated time interval for flight f to pass over sector s?
- How many flights will be in a given sector in a given time interval?
- What should the speed of the flight f be for it to land in time?

In the current implementation AMOS has been minimally extended by using its *foreign function* facilities that allows the definition of customized data representations and operators in C or Lisp. Future work includes technical problems such as new data structures for time stamps, special index structures for time intervals, a sliding “time window” to remove old data, query processing for temporal data, and active rules over temporal data, etc.

15.3.6 Real-time Query Processing

Thomas Padron-McCarthy

Real-time databases are databases with added *timeliness* requirements, for example:

- Transactions, queries, and other operations have deadlines.
- The time to execute a query or an operation is predictable, i.e. it does not vary significantly from time to time.
- The system estimates the real-time to execute queries and operations.
- Contingency plans can be defined that are queries and operations executed when the deadline approaches. In contingency plans there may be timeliness trade-offs regarding data consistency, precision, and completeness.
- Functions and queries can be defined in terms of the time to execute other queries or functions.

Since time is essential in real-time databases, it normally also needs temporal facilities, e.g. to access the clock or to timestamp and version data. 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.

15.3.7 Federated AMOS Systems

Magnus Werner

An *object-relational multidatabase query language* must present the user with possibilities to ask multidatabase queries. The simplest form is a remote query ranging over a single component database. This is not sufficient, however, since inter-database queries are needed that combine information from various component databases.

An important concept in database languages is the view. It allows a user to customize the appearance of information in the database. The view concept should be extended in a multidatabase environment. *Simple multidatabase views* allow us to achieve location transparency in a multidatabase environment by defining a view spanning multiple databases. This is desirable since most users want easy access to information without having to worry about its placement. Simple views alone will not suffice but should be extended to *structure transforming multidatabase views*. The task of structure transforming views is to resolve semantic heterogeneity inevitably present in a multidatabase environment. Types in one database may be represented as properties of some entity in another database. We thus need to be able to map between the different key components in the data model such as types, functions, values, and objects, to be able to combine information in different formats from various sources.

AMOS being a homogeneous multidatabase system gives us unique opportunities to study different issues such as inter-database query parallelism, multidatabase query optimization, and distributed active rules. We hope that distributed active rules will prove useful to maintain multidatabase view consistency and that performance gains will be achieved.

15.3.8 Parallel Main-Memory Query Processing

Jonas S Karlsson

New database applications increase the demand for high-performance databases. One way to meet this is to use massively parallel and distributed architectures. Parallelism is also good for query processing, since query processing often involves parallel operations such as applying functions to large sets of data. An important aspect is scalability, both for data and load (CPU-usage, messages).

The goal of this work is to develop technology for compiling OO queries into efficient code for parallel data base access on a parallel machine. The machine architecture we have in mind is a shared-nothing architecture with very fast point-to-point communication between the processors, and a relatively large amount of main memory on each processor. In such an architecture it is possible to store large databases in the main memory of the processors. For our experiments we are using a Parsytec transputer based parallel machine with MIMD architecture (Multiple Instruction Multiple Data). Parallelism seems very advantageous for fast access to stored data, and we are initially studying how a novel technique for parallel hashing, LH*, can be used for this.

Later in the project we plan to develop new query execution strategies to translate AMOSQL queries into optimal parallel execution plans that uses LH*.

This work is done in collaboration with Professor Witold Litwin (Paris Univ. 9) and Professor Peter Fritzson (LiTH). It is funded by NUTEK.

15.3.9 Domain Mediators for Finite Element Analysis

Kjell Orsborn

Scientific and engineering databases to support engineering applications are very important for future scientific and engineering environments. This project investigates how next generation database technology can be used to model and manage applications for finite element analysis (FEA), a central technique in for example computational mechanics.

A completely graphical interactive FEA system has been integrated with AMOS, with the analysis model stored in the database. The DBMS is also being extended to include domain operators for FEA permitting analysis of data within the database. The query language AMOSQL can be used to model and manage the structure of FEA models and the analysis process of FEA. These type of new database applications usually have a high level of complexity of both data and algorithms, and high volume of data and high requirements on execution efficiency. This puts new requirements on the database technology to be used, as for example abilities to represent and process queries including complex numerical operations. Database technology also provides new functionality to these applications as for example the ability to make ad hoc queries over models and data, and distribution of data among different engineering disciplines.

Kjell Orsborn has also developed a product model of mechanical parts in cooperation with SAAB Military Aircraft. It shows that an object-oriented query language is well suited for such models and makes it easy to make advanced queries of the model.

This project is a cooperation with Professor Lennart Karlsson (LuTH), Professor Martin Lesser (KTH) and Dr. Bo Torstenfelt (IKP, LiTH). It is funded by NUTEK in the *Complex Systems* programme.

15.3.10 Product modeling systems

Olof Johansson

A *product modeling system* (PMS) is a computer integrated development environment for a specific class of advanced products. A PMS consists of a central product model database interfaced with CAD-applications that support graphical designs of engineering models, and CAE-applications that make engineering calculations on the models.

An approach has been developed to manage the software engineering of product modeling systems. The key idea is to store a high-level PMS design specification as an extended entity relationship (EER) model in a data dictionary. The EER model is developed in cooperation with product and application experts using an object-oriented case tool, OOCASE.

Most of the source code for the PMS implementation is generated automatically, using SQL-based code generators. Our PMS-development system generates product model database schemas and user interfaces. It also generates interface modules in the native application development language of a CAD-system. Through these, a CAD application developer has access to the product model database.

By automatic generation of most of the software that is dependent on the EER model, changes in the data model can quickly be evaluated by experts and end users on a working PMS prototype.

Based on the experiences from three years of practice with the development of a power plant PMS, we expect to develop a PMS development platform architecture that allows high-level implementation independent specifications of PMS applications. The system will have facilities that enable a high degree of automated software implementation of user interfaces.

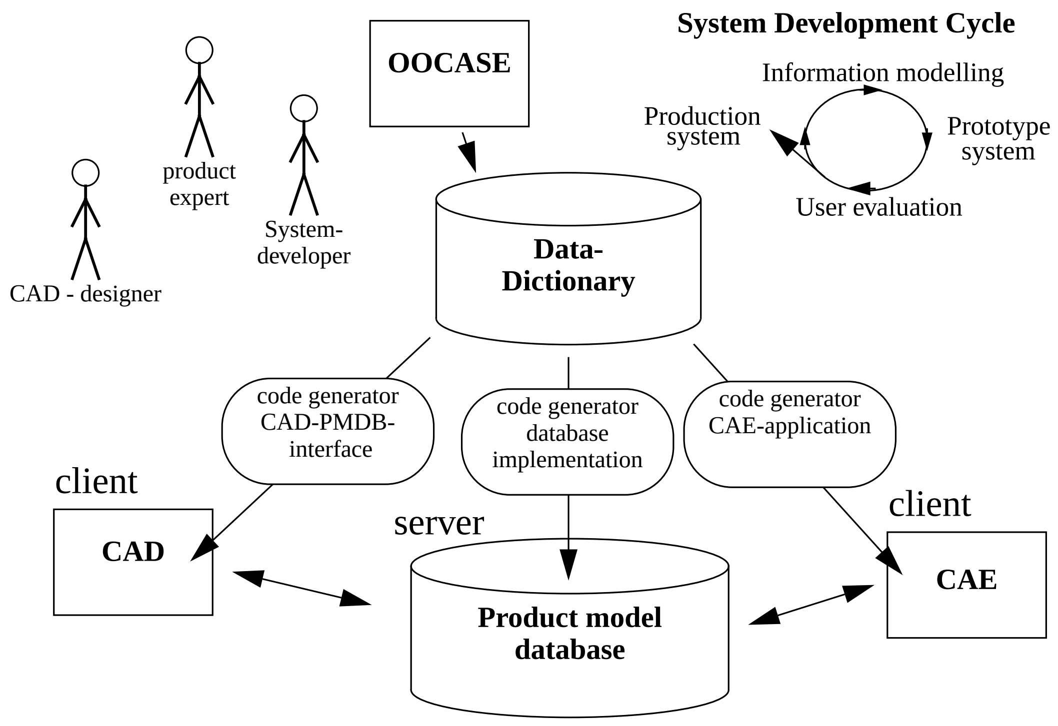


Figure 15-2: *The PMS Development System*

This project is a cooperation with ABB Stal and professor Sture Hägglund, LiTH. It is funded by NUTEK in the *Complex Systems* programme.

15.4 New Conferences

An industrial workshop on *Modern DBMS Implementation Techniques* was organized in cooperation with Softlab AB June 1-2, 1993. It had about 55 attendees from Swedish industry. The seminars were presented by Svein Olaf Hvassvold (Sintef, Norway), Witold Litwin (Paris Univ. 9, Berkeley), and Tore Risch (LiTH).

A new International Conference series on Application of Databases (ADB-94) was successfully launched in Vadstena June 21-23, 1994. ADB-94 was organized in cooperation with and sponsored by major professional organizations and many internationally recognized researchers. This new conference series aims at developing a synergy between database researchers, developers and application designers. It is intended as a forum to explore innovative applications of databases and innovative database services of specific applications.

One hundred papers from twenty-six countries were submitted to ADB-94. The program committees from Europe, US, and the Far East selected 28 papers, from 13 countries, i.e. about one paper per four submitted. Three distinguished speakers were invited. The keynote talk, *Database Applications in Telecom Systems* was given by Bernt Ericson, Director of Applied Research at Ericsson AB. The invited talks on *An Architecture for Cooperative Database System*, and on *The Singapore National Information Infrastructure Development* were given respectively by Jack Minker (U. Maryland, USA) & Michael Yap (National Computer Board, Singapore). Three tutorials were given, selected out of 10 submissions. Finally two panels debated selected database application domains, and, at the Closing Panel, the attendees gave their opinion about the conference itself. The conference had 117 participants from 15 countries. There was also significant local interest: 61 attendees were Swedish. The conference proceeding is published in the book series Lecture Notes in Computer Science by Springer Verlag (ISBN 3-540-58183-9). ADB-95 is planned to be held in Santa Clara (California) in December 1995

15.5 International Cooperation and Other Activities

Internationally, the group is participating in the EC Network of Excellence on Active Databases, ACT-NET and is a member of its student exchange program. We have also participated in several program committees of international database conferences. In March 1994 Tore Risch and Martin Sköld were invited to the international Dagstuhl Seminar on Active Databases at Schloss Dagstuhl in Germany.

Locally, the laboratory is participating in the competence center ISIS - Integrated Control and Information Systems, lead by Professor Lennart Ljung, ISY, Linköping University.

The laboratory teaches both undergraduate and graduate database courses. Graduate courses were held on Modern Database Systems, Distributed Databases, and Real-Time databases.

Two Licentiate Thesises were finished during 1994: In June Gustav Fahl finished his Thesis on *Object Views of Relational Data in Multi-database Systems* and in September Martin Sköld finished his Thesis on *Active Rules based on Object Relational Queries.*

15.6 Relevant publications since 1992

Papers:

Fahl, G., Risch, T., Sköld, M. (1993). AMOS - an architecture for active mediators. In *Proc. Workshop on Next Generation Information Technologies and Systems (NGITS 93)*, Haifa, Israel.

Flodin, S., Risch, T. Processing Object-Oriented Queries with Invertible Late Bound Functions, LiTH-IDA-R-95-10. To be presented at *21st International Conference on Very Large Data Bases*, Zurich, Switzerland, Sept. 11-15, 1995.

Johansson, O. Verktyg för objektorienterad programmering - Konstruktörsstödsystem, In proc. from *PRODUKTMODELLER'92*, Linköping, 27-28, Oct., 1992 (in Swedish).

Johansson, O. PRODUKTMODELLER - Objektorienterad datamodellering, Presented at *Workshop on Product Models (PM-93)*. Linköping, Oct., 26-27, 1993. Published in *PM-93*, ISBN 91-7871-212-2, Linköping, 1993 (in Swedish).

Johansson, O. Using an Extended ER-model based Data Dictionary to Automatically Generate Product Modeling Systems. In Witold Litwin, Tore Risch (eds.): *Applications of Databases*, *1st Int. Conf., ADB-94*, Vadstena, Sweden, June, 1994 (Proc.). LNCS, Springer-Verlag, ISBN 3-540-58183-9, 1994.

Litwin, W., Risch, T. 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.

Loborg, P., Holmbom, P., Sköld, M., Törne, A. Programming Flexible Manufacturing Systems. In *Proc. of Workshop on Adaptive Manufacturing Equipments*. Nutek 1992:53, Oct., 1992, CAELAB Memo 92-02.

Loborg, P., Risch, T. Sköld, M., Törne, A. Active OO Databases in Control Applications, *19th Euromicro Conference*,. Barcelona, Sep., 1993.

Loborg, P., Sköld, M., Törne, A., Holmbom, P. A Model for the Execution of Task Level Specifications for Intelligent and Flexible Manufacturing Systems. LiTH-IDA-R-92-22. In *Proc. of the Vth Int. Symp. on Artificial Intelligence, ISAI92*, Cancun, Mexico, Dec., 1992.

Orsborn, K. Applying Next Generation Object-Oriented DBMS for Finite Element Analysis. In Witold Litwin, Tore Risch (eds.): *Applications of Databases*, *1st Int. Conf., ADB-94*, Vadstena, Sweden, June, 1994 (Proc.), LNCS, Springer-Verlag, ISBN 3-540-58183-9, 1994.

Orsborn, K. Modeling of Product Data Using an Extensible O-O Query Language. LiTH-IDA-R-93-15. Submitted to the Int. Journal of Computer Integrated Manufacturing, Aug., 1994.

Orsborn, K., Risch, T., Torstenfelt, B. Produktmodellering med objektorienterade databaser och frågespråk. Presented at *Workshop on Product Models (PM-93)*, Linköping, Oct. 26-27, 1993. Published in *PM-93*, ISBN 91-7871-212-2, Linköping, 1993.

Risch, T., Sköld, M. Active Rules based on Object-Oriented Queries. LiTH-IDA-R-92-35. In special issue on Active Databases of *IEEE Data Engineering*, Jan., 1993.

Sköld, M., Risch, T. Compiling Active Object-Relational Rule Conditions into Partially Differentiated Relations. LiTH-IDA-R-94-10. Presented at the *Dagstuhl Seminar on Active Databases*, Schloss Dagstuhl, Germany, Mar., 1994.

Licentiate theses:

Fahl, G. *Object Views of Relational Data in Multidatabase Systems*. LiTH Licentiate of Philosophy Thesis No. 446, ISBN 91-7871-386-2, June, 1994.

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

Sköld, M. *Active Rules based on Object Relational Queries - Efficient Change Monitoring Techniques*. LiTH Licentiate of Philosophy Thesis No. 452, ISBN 91-7871-407-9, Sep., 1994.

Book:

Litwin, W., Risch, T. (eds.) *Applications of Databases, 1st Int. Conf., ADB-94*, Vadstena, Sweden, Jun., 1994 (Proc.). LNCS, Springer-Verlag, ISBN 3-540-58183-9, 1994.

Research Reports, etc.:

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

Flodin, S. An Incremental Query Compiler with Resolution of Late Binding. LiTH-IDA-R-94-46.

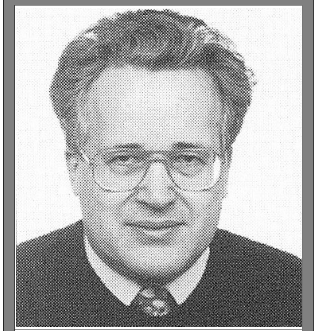
Flodin, S., Risch, T. Processing Object-Oriented Queries with Invertible Late Bound Functions. Submitted for publication, 1994.

Johansson, O. Beslutstödsystem för konstruktörer - Ett IT4-projekt, Faktarapport produktutveckling/ konstruktion, nr 4, Best.nr V020013, Sveriges Verkstadsindustrier, ISSN-1103-7067 (in Swedish).

Karlsson, J., Flodin, S., Risch, T., Sköld, M., Werner, M. AMOS User's Guide, CAELAB Memo 94-01, Dept. of Comp. Sc., Linköping Univ., 581 83 Linköping, Sweden, 1994.

EIS

16 Economic Information Systems



Birger Rapp
director

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

Some of the members of Economic Information Systems.
Back row: Björn Helander, Thomas Falk, Bengt Savén, Birger Rapp, Jörgen Andersson, Lars Poignant, Anders G. Nilsson, Henrik Eriksson.
Front row: Lennart Gombrii, Kristina Larsen, Lena Wigh, Anna Moberg, Jaime Villegas, Jan Ollinen.

16.1 General background

The division Economic Information Systems (EIS) was established in 1987 with its first full professor, professor Birger Rapp. Today the division belongs to the Department of Computer and Information Science. The research area 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. In 1992, EIS got its first adjunct professor, professor Rolf Rundfelt. Now two more adjunct professorships are going to be established within the division.

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

The research area 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 organizations, together with the effects 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 as well as training.

EIS have initiated cooperation with the Department of Management and Economic Informatics, Latvia University in Riga and the Department of Modelling and Simulation at Riga Technical University, Latvia. We also cooperate with the Department of Economic Informatics, Faculty of Economics, Vilnius University in Vilnius, Lithuania. These departments participate in our research-programme *SSIT - a method for organizational learning*.

EIS also cooperate with Lancaster University, Great Britain and Leuven University, Belgium.

16.2 Research areas

Within the division there are now five different research groups.

16.2.1 **ITOS**. ITOS stands for Information Technology and Organization Structure.

Society is in a state of constant change. There has been a tendency for workplaces to concentrate in large urban areas. Unfortunately this tendency has brought with it a worsening of social climate, increased rents for businesses and long commuting time 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. 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”. One goal of ITOS is to gain an insight into the interplay between the available techniques and the conditions under which people use these techniques, all seen from an organizational perspective.

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 the firm at the level of the society as a whole. Some of the researchers are Kristina Larsen, Jörgen Lindström, Anna Moberg, Jan Ollinen and Lars Poignant.

The projects are supported to some extent by Digital, Ericsson, Telia, Nutek and Arbetsmiljöfonden.

16.2.1.1 Selected publications

Licentiate thesis

Moberg, A. (1993). *Satellitkontor - En studie av kommunikationsmönster vid arbete på distans,* lic-avh nr 406, IDA-EIS, Linköpings universitet.

Poignant, L. (1994). *Informationsteknologi och företagsetablering - effekter på produktivitet och region,* lic-avh nr 441, IDA-EIS, Linköpings universitet.

Published

Larsen K., Moberg A., Ollinen J. (1992). Informationsteknologi och Nya Organisationslösningar - aspekter på kommunikation, *Third Annual Conf. on Computer and Information Science*, Linköping, Sweden, 1992.

Poignant, L., Rapp, B. (1993). Utlokalisering av tjänsteföretag, NUTEK R1993: 52.

Rapp, B. (1993). ‘Informationshantering på individ- och organisationsnivå”. In L. Ingelstam, L. Sturesson (eds.) *Brus över landet*, pp. 117-141, Carlssons bokförlag.

16.2.2 **SIMPRO**. SIMPRO stands for Simulation and Production.

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, particulary 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. One way to improve the possibilities for decision-makers can be training. The main project under this headings is SSIT - a method for organizational learning.

Some of the researchers are Thomas Bennet, Bengt Savén and Jaime Villegas. Two Ph D. dissertations are expected during 1995, Savén and Villegas.

The projects are supported to some extent by Samhall Klintland, Saab military aircraft and Nutek.

16.2.2.1 Selected publication

Licentiate thesis

Savén, B. (1990). *Produktionssimulering*. LiTH, Thesis No. 241 del 1: Inledning. ISBN 91-7870-704-8, del 2: Bok ISBN 91524-0944-9.

Published

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

Savén, B. (1992). Factory Simulation – concepts and one approach. *Roboties & Computer integrated Manufacturing*, Vol. 9. No. 4/5, pp. 341-345.

Savén, B. (1992). 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. (1992). The Evolution of the User Role in Simulation Studies. *Proc. of the European Simulation Multiconference,* Lyon, France, Jun. 7-9, 1993.

Savén, B. (1993). Cyklisk MRP-ett steg från MS till MPS. Plan-nytt, (5). *Swedish Production and Inventory Management Society*, Stockholm, Sweden, 1993.

Savén, B. (1994a). Beslutsstöd och simulering i verkstadsföretag – en sammanfattning av två enkätstudier. Technical Report LiTH-IDA-R-94-17, Dept. of Computer and Information Science, Linköping University, Sweden.

Savén, B. (1994b). The consensus staircase – three case studies of manufacturing simulation projects. In Morito, Sakasegawa, Yoneda, Fushimi, and Nakano, eds., *New Directions in Simulation for Manufacturing and Communications*, ORJS T-94-1.

Savén, B. (1994c). Cyclic MRP: Introducing capacity constraints into MRP. *Journal of Les Cashiers du FEPIMS.* 1994.

Villegas J., Sjölin M. (1992). Industrial Training with Simulation Games, *European Simulation Symp. Proc.*, Dresden, Germany, pp. 555. 1992.

Villegas, J. (1993). Qualification of decision makers: A process that can be supported by professional simulation games. In *Proc. of the European Simulation Symp.*, Holland-Delph, The Netherlands, 1993.

Villegas, J. (1994). Designing a management training programme for decision makers based on the “SSIT” methodology. In *Proc. of 4th Annual IDA Conference on Computer and Information Science*, Linköping, Sweden. 1994.

Villegas, J., Baertsoen, G., Henri, M-M. (1994). Two approaches in the use of simulation techniques for training purposes. *Proc. of the European Simulation Symp.* Istanbul, Turkey, Oct., 1994.

16.2.3 **ISS**. ISS stands for Information Systems and Control

This group, Information Systems and Control, is under creation. The graduate School of Skövde will support an adjunct professor in this area within EIS. Extensive research has been done in the area of Manufacturing Planning and Control by professor Birger Rapp. Two of the researchers are Jörgen Andersson and Rolf Larsson.

16.2.3.1 Selected Publications

Licentiate thesis

Larsson, R. (1992). *Aktivitetsbaserad kalkylering i ett nytt ekonomisystem,* lic-avh nr 298, IDA-EIS, Linköpings universitet.

Noghabai, M. (1993). *Värdering av strategiska datorinvesteringar- Med ett ledningsperspektiv på FMS- och KIS- investeringar*, lic-avh nr 371, IDA-EIS, Linköpings universitet.

Published

Andersson, J., Helander, B. (1993). Ekonomistyrning för små företag. Technical Report LiTH-IDA-R-93-37, EIS, Linköpings universitet.

Bennet, T., Olhager, J., Rapp, B. (1994). Computerization of manufacturing planning and control systems in Sweden 1980-1993. In *IFIP Conf. on Evaluation of Production Management Methods*, pp. 355-362, Elsevier Science Publ. B.V., Amsterdam, The Netherlands.

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

Olhager J., Rapp B. (1991). Modern Concepts for Computer-Based Manufacturing Planning and Control Systems, pp. 565-576. *Operational Research '90*. Ed. H.E. Bradley, pp. 565-576, Pergamon Press Oxford, England.

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

Olhager, J., Rapp, B. (1994). Different approaches to a flow shop problem. *European Journal of Operational Research*, 75(2):469-473. Elsevier Science Publ. B. V., Amsterdam, The Netherlands.

Olhager, J., Rapp, B. (1995). Operations research techniques in manufacturing planning and control systems. *Int. Transactions in Operational Research*, 2(1), Pergamon Press Oxford, England.

Ovrin P., Olhager J., Rapp B. (1990). Computer-Based Manufacturing Planning and Control Systems in Small Companies. *European Journal of Operational Research*, Vol. 45, pp 106-112, Elsevier Science Publ. B. V., Amsterdam, The Netherlands.

16.2.4 Accounting and Auditing.

In cooperation with six different accounting firms 3 - 6 researchers are, since 1991, offered a special “research-school” each year. The companies are KPMG- Bohlins, Deloitte & Touche, Ernst & Young, Lindebergs, SET and Öhrlings-Reveko. The researchers have mostly graduated from Business Administration at the University. The courses are offered in compact form during some research weeks. During the rest of their time the researchers are working with their accounting firm. This special education lasts for three years and is unique. In Sweden it is only offered in Linköping as a result of an initiative taken by professor Birger Rapp.

Some of the researchers are Anders Bäckström, Åsa Dössing, Bo Lagerström, Fredrik Nilsson, Peter Carlsson, Camilla Sjöström, Magnus Wiklund-Johansson and Per-Ove Zetterlund.

The projects and the “research-school” are supported to some extent by KPMG – Bohlins, Deloitte & Touche, Ernst & Young, SET, Lindebergs, and Öhrlings Reveko.

16.2.4.1 Selected publications

Licentiate thesis

Sjöström, C. (1994). *Revision och lagreglering - ett historiskt perspektiv*, lic-avh nr 417, IDA-EIS, Linköpings universitet.

Nilsson, F. (1994). *Strategi och ekonomisk styrning- En studie av Sandviks förvärv av Bacho Verktyg*, lic-avh nr 463, IDA-EIS, Linköpings universitet.

Published

Rundfelt, R. (1994). *Transnational Accounting; Swede - individual accounts*. MacMillan.

Rundfelt, R. (1994). *Tendenser i börsföretagens årsredovisningar.* Stockholms Fondbörs.

Rapp, B. (co-author) (1994). *Bilder av revision och revisorer.* Öhrlings Reveko.

16.2.5 Principal Agent Theory

An agency relation 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 or unwritten, among owners of factors of production and customers and employees. Three projects within this field have been finished during 1994. One project studies the change in agency costs due to the managers´ increase in ownership and change in leverage of the company when a management buy-out is made, (Peter Carlsson, 1994). In an other project firm relations are discussed, (Jonas Lind, 1994). Finally, professor Birger Rapp and Anders Thorstenson have published a book in 1994, “Vem skall ta risken?” (Who will bear the Risk?).

16.2.5.1 Selected publications

Licentiate thesis

Carlsson, P. (1994). *Separation av företagsledning och finansiering - fallstudier av företagsledarutköp ur ett agentteoretiskt perspektiv*, lic-avh nr 414, IDA-EIS, Linköpings universitet.

Lind, J. (1994). *Creditor - Firm Relations: an interdisciplinary analysis*, lic-avh No. 451, IDA-EIS, Linköpings University.

Published

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

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

16.3 Staff

Group leadership and administrative staff:

Birger Rapp, Econ. Dr., professor
Rolf Rundfelt, Econ. Dr., adjunct professor
Thomas Falk, Adjunct professor
Nils Göran Olve, Adjunct professor
Eva Elfinger, Secretary

Employed graduate students:

Jörgen Andersson, B. Sc.
Thomas Bennet, M. Sc.
Anders Bäckström, B. Sc.
Åsa Dössing, B. Sc.
Björn Helander, B. Sc.
Bo Lagerström, Econ. Lic.
Kristina Larsen, M. Sc.
Jörgen Lindström, M. Sc.
Mattias Lundquist, B. Sc.
Anna Moberg, Econ. Lic.
Fredrik Nilsson, Econ. Lic.
Lennart Ohlsén, M. Sc.
Jan Ollinen, M. Sc.
Lars Poignant, Econ. Lic.
Bengt Savén, Tech. Lic.
Jaime Villegas, Ph. D.
Magnus Wiklund-Johansson, B. Sc.
Per-Ove Zetterlund, B. Sc.

Graduate Students:

Gudrun Carlsson
Elisabeth Gudinge
Thomas Jansson
Rolf Larsson, Ph. Lic.
Mehran Noghabai
Maria Ohlsson
Annette Stolt

CV s senior staff

Professor Birger Rapp is born in 1940. He is married to Birgitta and they have three children. He has, since 1987, a chair in Economic Information Systems, Department of Computer and Information Science, Linköping Institute of Technology and Linköping University. Earlier he has been associate professor in Business Administration and in Production Economics and in Forest Economics at the universities of Stockholm, Linköping and Umeå. Before 1987 deputy full professor in Business Administration at Stockholm university. He has also been visiting professor at EIASM, Brussels.

He has been the president of SORA, The Swedish Operational Research Association and their representative both in EURO (The Association of European Operational Research Societies within IFORS) and in IFORS. He was the second president of EURO and is now appointed as vice president at large for IFORS 1995-1998. He has been session coordinator at several conferences and he was the general coordinator for the EURO-TIMS conference in Lausanne.

He belongs to the editorial (advisory) boards to the following international journals, EJOR, IJMSD, JORBEL and Omega. He is also a referee to several other scientific journals.

He has published books in investment theory, in production planning and in control and principal agent theory and a great many papers in international journals.

He has been a senior consultant to many Swedish companies and is the president of the Pronova Research and Development Board in Sweden.

He is also IMIT’s (Institute of Management of Innovation and Technology) local manager in Linköping.

Adjunct professor Rolf Rundfelt is born in 1939. He is adjunct professor in Economic Informations Systems, especially in Swedish and International External Accounting. He is a doctor in Business Administration, University of Stockholm, 1974. He has been lecturer at the University of Stockholm since 1966. He is a member of the Board of the Swedish Financial Accounting Standards Board and the Board of International Accounting Standards Committee and the European Accounting Advisory Forum.

His research interest is mainly External Financial Accounting and he has produced the following books in 1994:

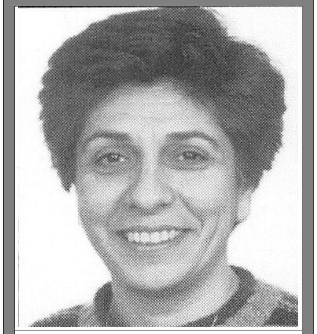
Transnational Accounting; Sweden - individual accounts, MacMillan.
Tendenser i börsföretagens årsredovisningar, Stockholms Fondbörs.

Adjunct professor Thomas Falk is born in 1944. He has a Ph. D. from Stockholm School of Economics, 1976. He is adjunct professor in Economic Information Systems, especially in Economics of Information, and senior management consultant at Cepro AB. Between 1992 - 1994 he was the president of the Swedish Institute for system development (SISU). He was director, Business Development at Cap Gemini Nordic 1989 - 1992, director, Strategic Information technology, at Digital Equipment AB 1987 - 1989, senior vice president at Svenska Handelsbanken 1985-1987, associate professor at Stockholm School of Economics 1979-1985. He has been visiting professor at the European Institute for Advanced Studies in Management (EIASM) 1981-1982 and at the Pennsylvania State University, 1977-1979. He is a member of the Royal Swedish Academy of engineering Sciences.

Adjunct professor Nils Göran Olve is born in 1947. He is adjunct professor in Economic Information Systems, especially in economic control. He has studied in Stockholm and at UC Berkeley, California. Degrees from the Stockholm's School of Economics: "civilekonom" 1970, "Ekonomie doktor" (Ph. D.) (1977), "docent" in Business Administration 1981. 1970-1986 he was employed by the Stockholm School of Economics and its affiliates as researcher, teacher and course director. During part of this period deputy full professor and department head; visiting professor at EIASM in Brussels and director of the International Teachers Programme (ITP). Since 1986 a partner in CEPRO, a consultancy firm in Stockholm. In 1993 deputy full professor at Linköping Institute of Technology; currently part-time with Swedish Institute of Systems Development.

Main professional interests: economic control systems which encourage decentralized initiatives and yet maintain and support the synergies of the whole corporation; teaching management principles to non-specialists. He has published in Swedish and English books and some 30 articles in anthologies and journals. He is also chairman of the "Friends of the Opera College", Stockholm. Unmarried.

IISLAB



Nahid Shahmehri
director

17
The Laboratory for Intelligent Information Systems

Object-oriented systems
Computer Supported Cooperative Work
Process Modelling
Workflow Management
Information Retrieval
Hypertext
Description Logics

Some of the members of the Laboratory for Intelligent Information Systems.
Back row: Nahid Shahmehri, Lena Wigh, Patrick Lambrix.
Front row: Martin Sjölin, Johan Tufvesson, Thomas Hall.

17.1 Introduction

The Intelligent Information Systems Laboratory conducts research in an area which arises from 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 by several users and applications. The traditional knowledge bases of e.g. expert systems are not suitable for such things as multiple users and large 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 IISLAB's focus of interest.

IISLAB's research consists of both an implementation or system development strand and a theoretical strand. The system development and theoretical components interact so that each provides input to the other. There is also interaction within the theoretical components. The theoretical work within the group has been presented at a number of competitive international conferences, while parts of the system have been used both by an industrial group, for prototyping, and an outside university group, for education. The system has also been released on the Internet. There has been significant interest with over 1000 people having obtained a copy of the system at the Linköping site. (The system can also be obtained from two official mirror sites in the U.S.A.)

17.2 Members

The following people have been members of the group during all or part of 1993/94:

Laboratory leadership and administrative staff:

Nahid Shahmehri, Ph. D., current director (since July 94), assistant professor.
Lin Padgham, Ph. D., former director, associate professor.
Lena Wigh, secretary.

Graduate students:

Patrick Lambrix, Lic.
Michael Jansson, Lic.
Martin Sjölin
Thomas Hall
Niclas Wahllöf

Other Members:

Ralph Rönnquist, Ph. D., until 1993.
David Partain, research engineer, until July 1994.
Mats Gustafsson, final undergraduate thesis project.
Stefan Kalmelid, final undergraduate thesis project.
Jeffrey A. Lorenzen, guest research student from the U.S.A.
Johan Tufvesson, final undergraduate thesis project.
Dermot McGahon, final undergraduate thesis project.

17.3 LINCKS System

The aim in our system development work (the LINCKS system) has been to gradually build up a sophisticated and complex information system which has characteristics necessary for exploring issues of interest, but which are unavailable in commercial or other readily available systems. These characteristics include: a very flexible basic representation, management of historical data, support for multiple users, shared information, and a declarative language for obtaining different object views. They are reflected in the features of LINCKS.

LINCKS is an object-centered multi-user database system developed for complex information system applications where editing and browsing of information in the database are of paramount importance. The focus is on sharing small information chunks which combine to make up complex information objects used by different users for different purposes. The information chunks are semi-structured in that they contain one part which is well-structured to facilitate addition of A.I. processing within the system, and one part which is unstructured and suitable for management by the user. 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.

Some of the interesting features of LINCKS are:

- Multiple user support.
- A hypertext X-windows based user interface.
- Composite objects - although all objects are built up of small pieces, the user interface presents these as single *composite objects*, allowing editing over the entire composite object using an emacs-like editor. Structured editing is editing which supports and is in some way based on the notion of the internal structure of the edited objects (documents). In LINCKS the text editor in the user interface allows viewing and editing over the entire document 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.
- Database history - the LINCKS system maintains information regarding the history of objects and actions within the system. Object history is maintained at both the single chunk and composite object level. Past versions of objects can be accessed and reactivated. The historical information 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. Another case is when one has disconnected workstations which operate on portions of the database during a period of time (perhaps quite long) when they are disconnected.

- Alternative views - LINCKS allows a user or application developer to interactively define alternative views on the underlying database objects. This is particularly useful in applications where the same information is to be used for different purposes or by people in differing roles which require a different composition and display. 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 approach gives us 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.
- Information sharing - LINCKS provides an easy way for users to share database objects or for one user to share information across all of her accessible objects. This is done by linking objects into composition objects. Thereby, any changes to that object by one user will result in updates to all compositions which have that object as a component.
- Parallel Editing Notification - As LINCKS is a multi-user system and allows information sharing, this leads to possible conflicts. Consequently, the LINCKS system maintains a list of objects being edited and issues warnings when parallel editing occurs.

In the following subsections we describe briefly some issues which are of research interest and on which we have been working during the reported period.

Propagation of Change

In his Licentiate thesis, Michael Jansson studied the propagation of change in an intelligent information system that maintains historical information. Propagation of change occurs when a database manager executes transactions repeatedly, though to different parts of a database and possibly with some small variations. A number of operations can be performed automatically via propagation, e.g. merging variants of information, undoing a prior change to some information without losing recent changes and fixing the same bug in different releases of information. Michael discusses problems which arise when propagating changes and proposes a tool to perform propagation of change in the context of the LINCKS system.

LINCKS and WWW

The World Wide Web (WWW) continues to increase in size and usage. In fact, the WWW is one of the fastest growing services on the Internet. We make our papers, articles, and other pieces of information available on the WWW, marking up their contents (text, graphics, sound, movies) in HTML. We have created a prototype WWW front-end to LINCKS. The front-end is a normal client to LINCKS and it enables users to browse a LINCKS database using any WWW browser. Further, we use LINCKS for tracking owner, source, and creation date of information pieces and documents. Instead of using *copy-and-paste* for information duplication, we are using links to the *real* source and thus maintain trace-ability. The prototype generates HTML documents on-the-fly from information pieces in the database. A document's logical structure is described by a simple grammar, as well as *where* in the database the different parts of the logical structure can be found. Moreover, by applying different views to the *same document* root object we are able to provide: abstract; table of contents (of different levels); a chapter, a section, or a subsection; the full article; with or without footnotes; or any combination of the above. We can supply the users with a choice of granularity *without* duplicating the information into different files. This work is part of Martin Sjölin's Licentiate thesis work.

During 1995, we plan to work on a new version which should overcome the short-comings in the prototype as well as extend the functionality with form support, history browsing, etc. Johan Tufvesson has started this work as his undergraduate thesis project.

17.4 Improved Workflow Management in CSCW

Many of the office information systems and the database systems give users the possibility to design, execute, and manage business process networks. In the research literature of *Computer Supported Cooperative Work (CSCW)*, these systems are referred to as *workflow systems*. The general ability of such systems is that they can route data between different programs or human resources. Workflow software is a technology that can increase the productivity of multi-person business processes. This is mainly achieved by improved information feedback and tools to monitor, manage, and control business processes. Individuals can improve their work both in the personal and organizational context.

Today's workflow management systems lack a number of important properties which give rise to a number of challenging research issues such as:

- *Modifiability*: to facilitate for the user of the system to modify existing models and also to create new models from existing ones.
- *Usability*: to consider the user's level of expertise in the design of the process description language.
- *Flexibility*: to allow the user to handle unexpected situations, i.e. exception handling.

- *Reusability*: to aid the user to find and reuse the process descriptions already existing in a workflow system.
- *Learning*: to observe the earlier processes in order to learn and make suggestions when the user is defining new processes.
- *Protection and information sharing*: to provide authorization and access control while allowing information sharing.

The long term goal of this project is to investigate how workflow systems can be improved to be more usable and flexible. The short term goal of the project is to investigate the process modelling aspects of process management including such issues as *reusability* and *learning*. We will also look at the business process reengineering aspects of the workflow management. Thomas Hall is working on this project in cooperation with Telia Research in Linköping.

There are many research issues common to process modelling and the software development process. We intend to investigate the results from this project in the context of modelling the software development process.

17.5 Information Retrieval and Hypertext

As more and more text becomes available in electronic from, it becomes increasingly difficult for both the authors and users to find the information they need. We are working on ways to locate text that discusses the same ideas, although possibly in quite different terms. Once located, these pieces of text can be joined together with hypertext links. These *conceptual links* might be between text that has many words in common, but might also be from short key phrases to fuller descriptions. Having such links available for navigation will be useful for authors to find other sections that need to be updated when a change is made. It will also be helpful for readers to find related material for further explanations or more (or less) details.

Jeff Lorenzen is working on information retrieval techniques that can be used to find these links. Several common text searching systems including *agrep* and *Smart* have been examined, but most of these rely on statistical occurrences of words. This superficial understanding of the text is sufficient for many tasks but can clearly be improved. Jeff is working on ways to merge these word-based methods with new natural language methods to improve the results.

To make these links usable, Dermot McGahon is creating a flexible hypertext interface for xlincks. The interface will allow for the management and navigation of links including one-to-many links, multiple directional links, and links with associated actions such as change notification.

17.6 Extending Standard Description Logics

Description logics are languages tailored for expressing knowledge about concepts and concept hierarchies. They are usually given a Tarski style declarative semantics, which allows them to be seen as sub-languages of predicate logic. One starts with primitive concepts and roles, and can use the language constructs (such as intersection, union, role quantification, etc.) to define new concepts and roles. Concepts can be considered as unary predicates which are interpreted as sets of individuals whereas roles are binary predicates which are interpreted as binary relations between individuals. The basic reasoning tasks are unsatisfiability and subsumption checking. A concept is unsatisfiable if it always denotes an empty set. Subsumption represents the *is-a* relation. A whole family of knowledge representation systems have been built using these languages and, for most of them, complexity results for the subsumption algorithm are known.

Description logics comprise a well established theoretical field which also has a natural coupling to applications in intelligent information systems. In his final undergraduate thesis Stefan Kalmelid describes a prototype system that connects LINCKS and a description logic system (CLASSIC) giving advantages to both kinds of systems. An advantage for LINCKS is that it receives a typing mechanism and a query language through CLASSIC. CLASSIC is enhanced as it can use LINCKS' browsing mechanism as a user interface to browse through the *is-a* hierarchy and, more importantly, the history mechanism in LINCKS allows CLASSIC to browse through previous versions of the objects.

In future work regarding LINCKS we intend to add reasoning modules to the system through the connection with a description logic system such as CLASSIC. Description logics provide a firm theoretical base to investigate the issues which appear when we try to integrate classification and other kinds of reasoning. The field of description logics has matured during the last few years with respect to solving and understanding most of the basic problems in the area. It is now ripe for more complex additions such as defaults, temporal information, and *part-of* relationships. It is in this area that we have concentrated this part of our work.

with default reasoning

Several approaches have been explored for combining description logics and default reasoning. Lin Padgham and Bernhard Nebel found that serious problems can appear in the naive combination of classification and default reasoning, 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.

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. In an evaluation of the system, we found that it has an accuracy in the same range as doctors. This work was done by Lin Padgham in cooperation with ASLAB and has led to Tingting Zhang's Ph. D. thesis.

Another approach has been taken by Niclas Wahllöf, who extended the CLASSIC system by adding a default reasoning module on top of CLASSIC. The new system uses the default information to reason about individuals.

with part-whole reasoning

The sort of reasoning we may hope to do in a description logic system supporting *part-of* relations includes such things as deducing the existence of composite objects based on the existence of their parts, answering questions such as whether a particular object is a part of some other object, and determining whether one class is a possible building block for another class. Lin Padgham and Patrick Lambrix proposed a framework for representing and reasoning about *part-of* relations. To facilitate *part-whole* reasoning they defined the notion of *compositional inclusion* as the relation on which the *part-of* hierarchy is based. Compositional inclusion is a partial ordering and supports reasoning about *part-of* in a similar way as the subsumption hierarchy supports reasoning about *is-a*. A new kind of inferencing was defined, *compositional inferencing* which lets the system infer new compositions on the basis of the existence of available parts. They also examined the usefulness of the approach in a document management application. This led to extensions of the language in both expressivity and inferencing.

with temporal information

We have also looked at a temporal description logic called T-LITE, which is based on a standard description logic and the temporal logic LITE, which was developed in our lab. T-LITE allows extensions of concepts to be time-dependent in two different ways: Firstly, individuals can belong to a concept at a particular time, but not at others. For example, Jeff is a car owner at a particular time *t1*, but not at time *t2*. Secondly, concepts can be defined in terms of the development of objects. For instance, a traffic-light might be defined as a light cycling in colour from green through yellow to red at consecutive times. This work was done by Patrick Lambrix and Ralph Rönnquist.

17.7 IISLAB's Publications 1993-1994

17.7.1 Theses

Kalmelid, S. *Combining a Terminological Reasoner with an Object-Centered Database*, Final undergraduate thesis, Department of Computer and Information Science, Linköping Univ., 1994.

Zhang, T. *A Study in Diagnosis Using Classification and Defaults*. Ph. D. thesis No. 302, Linköping Univ., Sweden, 1993.

Jansson, M. *Propagation of Change in an Intelligent Information System*, Licentiate thesis No. 381, Linköping Univ., Sweden, 1993.

17.7.2 Externally Published papers

Padgham, L. Löwgren, J. A user interface management approach for object oriented database applications. *Journal of Systems and Software*, 27(3).

Padgham, L., Lambrix, P. A framework for part-of hierarchies in terminological logics. In Doyle, J., Sandewall, E., and Torasso, P., eds., *Principles of Knowledge Representation and Reasoning: Proc. of the Fourth Int. Conf.*, pp. 485–496, Bonn, Germany, May 24-27, 1994.

Sjölin, M. A WWW Front End to an OODBMS: *Proc. of the Second Int. World Wide Web Conf.*, Chicago, USA, 1994.

Sjölin, M. LINCKS – ett verktyg för CSCW. Lars Oestreicher, ed., *Design i MDI - Konferensdokumentation till STIMDI 94*, 1994.

Lambrix, P., Rönnquist, R. Terminological Logic Involving Time and Evolution: A Preliminary Report. In *Proc. of Seventh Int. Symp. on Methodologies for Intelligent Systems, ISMIS 93*, pp. 162-171, LNAI 689.

Padgham, L., Nebel, B. Combining Classification and Non-Monotonic Inheritance Reasoning: A First Step. In *Proc. of Seventh Int. Symp. on Methodologies for Intelligent Systems, ISMIS 93*, LNAI 689, pp. 132-141, Trondheim, Norway.

Padgham, L., Zhang, T. A terminological logic with defaults: A definition and an application. In *Proc. of 13th Int. Joint Conf. on Artificial Intelligence*, Vol. 2, pp. 662–668, Chambéry, France, Jul. 29 - Sep. 3, 1993.

Wirén, M., Rönnquist, R. Fully Incremental Parsing. In *Proc. of the Third Int. Workshop on Parsing Technologies.* Tilburg, The Netherlands and Durbuy, Belgium, Aug. 10-13, 1993.

17.7.3 Internal Reports

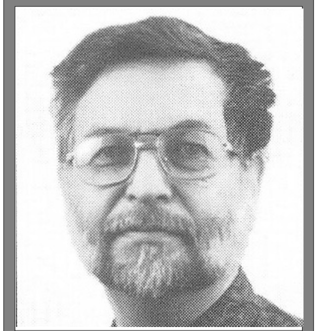
Hall, T. A Framework for User Support. *Proc. of the Fourth Annual IDA Conf. on Computer and Inf. Science*, Univ. of Linköping, Sweden, 1994.

Padgham, L., Lambrix, P. Wahllöf, N., Zhang, T. Default Inheritance in Commonsense Reasoning: Implementing, Using, Evaluating, Unpublished Manuscript, 1994.

Sjölin, M. LINCKS – A Platform for Computer Supported Cooperative Work. Demo at the Conf. on Computer Supported Cooperative Work, Chapel Hill, North Carolina, 1994.

Lambrix, P. Sjölin, M., Padgham, L. LINCKS - A Platform for Cooperative Information Systems. Unpublished manuscript, 1993.

LIBLAB



Roland Hjerppe
director

18
The Laboratory for Library and Information Science

User interfaces
Document description
Knowledge organization
Hypermedia
Information management
Networking

Some of the members of the Library and Information Science laboratory.
Upper row: Jonas Persson, Andreas Björklind.
Lower row: Bodil Mattsson-Kihlström, Lisbeth Björklund, Roland Hjerppe.

18.1 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 to and use of documents and document collections. The information technologies par excellence today are of course computers and telecommunications.

Within this very broad area the main objects of study are the issues of designing and using catalogues - tools for access to large collections of documents. The application domain within which this research so far has been carried out has been 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. Libraries can be regarded as a social 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 the interactions of these 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, and which 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. In the archives environment specific classes of old collections of records are progressively being digitized. The development of computerized access tools for the collections of traditional records is also growing. The computer-based creation, use and storage of records is furthermore increasing. All of these developments raise issues 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.). Artifacts, 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 artifacts (and *objets trouvées*) have the same functions as libraries and archives have for texts. For some artifacts (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.

Libraries, archives, museums, and national heritage institutions are the prime examples of institutionalized memory institutions. They can also be regarded as cultural repositories in that they collect, describe, conserve, and organize for access items of material culture.

In addition there are institutions and persons performing the same tasks for natural objects. Among these a distinction can be made between those that focus on dead items: rocks, shells, stuffed animals etc., and those that are concerned with living matter: genes, seeds, plants and animals, environments. In the latter case there are specific problems in that living matter has life-cycles and thus has both strict needs with regard to the immediate environment and requirements for continued renewal of the stock.

18.2 LIBLAB's Research program

The first research program for LIBLAB was written 1982. A second, revised, program, the main contents of which are included below, was formulated in 1988.

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 behavior, 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.

A third version of the program, focusing on the next five to six years and taking into consideration the changes in the environments, and the reorientations of interests at LIBLAB discussed below is in the process of being defined.

The trend towards the convergence of communications and computers, the use of networked information resources, vis. Internet and the World Wide Web, through powerful workstations as well as PDAs (Personal Digital Assistants) 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 situations of the individual user and the characteristics of the collections and their items.

The scope of the research at LIBLAB has in recent years been broadened to cover implications for archives, museums and cultural heritage institutions as well as libraries. Technical documentation is another area that also is receiving attention, mainly because it seems to be the domain in which formalisms for document description and architecture, and their applications - e.g. SGML and HyTime, are having an impact.

Publications seem to be the only class of documents that only incidentally have attributes related to spatio-temporal coordinates as an important characteristic. For most of the other types of documents discussed above it is important to specify from where and when they originate

Issues relating to access to media forms that traditionally have been broadcast: recreation/entertainment, news, opinion etc. have in the last year been recognized by LIBLAB as an emerging area meriting close attention. One of the reasons is the materialization of access through cable to hundreds of TV-channels, video-on-demand etc. Two other reasons are the multimedia description issues arising, and the long neglect in library and information science of most aspects of retrieval of recreation/entertainment.

18.3 Laboratory Members

Laboratory leadership and administrative staff:

Roland Hjerppe, M. Sc., researcher
Bodil Mattsson Kihlström, secretary, joined LIBLAB in summer 1993

Researchers and graduate students:

Andreas Björklind, B. Sc., doctoral student since 1991
Lisbeth Björklund, B. Sc, doctoral student since 1985
Jonas Persson, M. Sc., doctoral student since 1991
Åke Sivertun, Ph. D., joined LIBLAB in spring 1993

18.4 Progress Report and Current Research

The activities since 1992 are described briefly, to provide a background to present activities as well as continuity for readers of previous reports.

The activities during 1993 and 1994 have been characterized by the work on the TemaKat/IdaKat-project, further described below, and by the continued development of an undergraduate education program in Informatics.

18.4.1 Cooperation and participation in committees etc.

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, which finished its work in 1993.

Roland Hjerppe is now participating in the new Working Group 13, Virtual and Augmented environments for CSCW that was initiated in 1994.

LIBLAB has a close cooperation with the Swedish Defence Research Establishment - FOA, in the area of Geographical Information Systems - GIS. Two of the doctoral students at LIBLAB, Andreas Björklind and Jonas Persson are funded partially by FOA for their activities in this area.

Andreas Björklind has since its inception in September 1993 been a member of the Board (treasurer) of the SGML Users' group of Sweden.

Roland Hjerppe is a member of the Swedish Reference Group for EU's Libraries Program.

18.4.2 Research areas

Research activities at LIBLAB during this period has been grouped into six areas

- The TemaKat/IdaKat-project
- Geographical Information Systems - GIS
- Description of documents
- Cooperation
- Users and information access systems
- Networked information

18.4.2.1 The TemaKat/IdaKat-project

TemaKat/IdaKat has been the major research undertaking of LIBLAB during 1993 - 94. The original goal was 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 was 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 most of the prospective users of TemaKat were interviewed extensively with regard to their reading and writing habits, their use of catalogues, libraries and literature. The resulting transcribed 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 phase the findings from the first phase and the ideas and results from earlier HYPERCATalog experiments were merged and a resulting specification was used to produce a database, built on top of the RAM-DBMS WSIris by a group of students, and to initiate work on an interface.

In the third phase when the TemaKat was to be introduced and its uses to be studied, as a part of an evaluation, the supporting infrastructure, a LAN was not in place at the Themes department. To save the work done it was decided to do an implementation at LIBLAB's home department, IDA, instead. (The presence of the necessary LAN was not assured at the time of original planning for the project and IDA was thus proposed as a testing ground in the first proposal to the funding agency, which, however, decided that the Themes department should be the user group.)

The reorientation (with the concomitant name change to IdaKat) had deep consequences. The user interface that had been designed for a Mac + (which was the least common denominator at the themes) had to be completely redesigned for Open Windows on SUN workstations, taking into account larger screen and the capabilities for multiprocessing etc. The database content which was to be taken from the catalogue of the library of the Themes department (available in MARC-format) had to be replaced by the file produced at IDA of reports and other publications acquired during the years and produced by many people with no training in cataloguing. The first two attempts at designing and implementing an interface were failures for a number of reasons and at present work is going on at extending WILLOW (the Washington Information Looker Upper Layered on Windows), a Z39.50 client, to handle links and input of bibliographic information, and provide a two-layered access restriction. The actual use of IdaKat is expected to begin in the fall of 1995.

One of the, few, benefits of this not too uncommon delay in a long-term project is that Internet meanwhile has become a household word and that we have recognized that we in our HyperCatalogs vision cannot isolate the local catalogue, that TemaKat and IdaKat were designed to be, from the rest of the world's information resources.

18.4.2.2 Geographical Information Systems - GIS

The research activities in Geographical Information Systems - GIS are focused both on two distinct, but related areas concerning future GIS - generalization and architectures for spatio-temporal structuring, and on applications of traditional GIS to environmental issues, especially waste management.

18.4.2.2.1 Generalisation

The main investigator in this activity is Jonas Persson, who intends to base his dissertation on the ideas presented below.

Cartographic generalization has been used for long to obtain simpler maps and maps in smaller scales. As the spatial information systems have evolved from being map presentation systems to true information systems with reasoning capabilities it has turned out that generalization also can work as a powerful heuristic for solving some problems.

One such problem is finding the shortest path in a complex environment. The more complex the map is the greater problem. In fact the size of the problem increases exponentially compared to the map complexity. To get rid of this problem it is possible to generalize the map by filtering, straightening simplification etc. and then solve the problem on the simpler map. After that the path is projected to the original map (if necessary) to give a hint of a possible solution. The preliminary solution is then refined until a reasonable one is achieved.

In management and command control systems there are several other knowledge structures, except for the map, to keep in mind, e.g. personnel, supply, time and plans. If a similar kind of generalized reasoning could be used for them much work might be saved.

A feature with generalizations is that they reduce the amount of information. This should not be a disadvantage since it is always possible to recreate the previous states from the original data.

However a problem is how the generalizations should be performed and on what assumptions they should be based. In traditional data handling these approximations are performed on quantitative (exact) information, but in many cases these apparently precise data are not at all that precise. Qualitative reasoning is a recent trend in artificial intelligence and spatial information handling. Instead of using quantitative, numeric data, relative and approximate terms are used, e.g. north of, close to, straight and small. This way of reasoning is more like the one performed by humans when trying to solve spatial problems. By using qualitative reasoning the hope is to achieve more appropriate generalizations in a natural way.

18.4.2.2.2 An Architecture for Organising Dynamic Information About Space and Time

The main investigator in this activity is Andreas Björklind, who intends to base his dissertation on the ideas presented below.

Today's information systems are stretched to their limits in situations of continuous, large quantity flows of spatio-temporal data. Such situations are encountered in, for example, assessment and control of crises and emergencies. Meta-databases for the management of data about data have become a necessity. There are four main differences between the meta-databases for assessment and control systems and traditional information retrieval systems (e.g. bibliographic information systems). The first is the need for speed in generating and updating the meta-database. Secondly there are spatio-temporal aspects of the handled information. The third difference is the large volume and variety of information to be managed and to be transformed to symbolic form. Fourth is the collaborative aspect of the interaction.

In this project an architecture for organizing heterogeneous spatio-temporal multimedia data is proposed. In large scale assessment and control operations the scale of information flow is vast. One of the most important problems is not just to allow for the flow itself, but rather to automatically interpret the structure and contents of documents, and relate them to each other.

Common to all documents in this project are that they are coordinate based, i.e. linked to one or more points or regions in space and time. For this reason all information flowing in to the system can be indicated on a map together with a time stamp. As a consequence a geographical information system (GIS) will be required, and a GIS that has been developed at FOA in Linköping (FOA is the Defence Research Establishment in Sweden) will be used.

The international standard for Hypermedia/Time-Based Structuring Language — HyTime (ISO 10744) — an extension of SGML (Standard Generalized Mark-up Language, ISO 8879), provides a formal framework in this project. It is suitable for the descriptive tasks arising in the context of assessment and control systems, both for the planning tasks and for the meta-database design. One of the tasks in the project is to analyze the various kinds of incoming information with regard to its various characteristics. The resulting analytic information will be used to design methods of symbolic description for the meta-database. It will also be used to design appropriate indexing and classification schemes, and for creating thesauri. The transformation of the information to symbolic form, using HyTime, will enable it to be used both in reasoning and for subsequent presentation in the GIS-system.

The meta-databases in this project are concerned with the logical structure and access methods of documents in the database. The main problem is to design a meta-database that can handle a large flow of documents in which spatio-temporal data are of great concern. The meta-database cannot be fixed in its structure, but must be dynamic depending on the actual crisis. The meta-database must also include means for decision making based on various methods for spatio-temporal reasoning that can be used in other subsystems or applications that will be using the meta-database. It will be necessary to have means for selecting the information that is available in the database with respect both to time and space.

18.4.2.2.3 Applications of GIS in waste management

The main investigator in this activity is Åke Sivertun.

18.4.2.3 Cooperation

The main investigator in this activity is Roland Hjerppe, together with Erland Jungert, FOA.

Work in this area is reported e.g. in a report from the COST-14 Project CO-TECH, Working Group 3. (see “other publications”).

18.4.2.4 Description of documents

The main investigator in this activity is Roland Hjerppe.

The work on generalization of the concept of document, and the consequences for descriptions, has been reported e.g. at the Third ISKO Conference (see under “External refereed publications”).

18.4.2.5 Users and information access systems

The main investigator in this activity is Lisbeth Björklund, who is writing her dissertation based on the ideas presented below.

18.4.2.5.1 Formal methods for qualitative analysis

One area which has a long tradition in designing large public systems, is Library and Information Science. We have here two kinds of systems, used by many people, who have not been involved in their development and design, Information Retrieval systems for bibliographical databases and OPACs (On-line Public Access Catalogues) offered for access to information on library holdings. These systems were originally designed for other user groups, librarians and intermediaries, and later on adapted for use by other people. These adaptions have been guided by studies of the use of the systems, which has led to a long tradition of user studies in LIS.

User studies in this area have focused on either studying the over-all information behavior of groups of people, studying the actual use of different systems in order to improve the usability or studying the interaction of an individual and a system in order to improve the interface or the retrieval technique. The techniques used have mainly been elicitation, trying to drag out the information from the user, or discovery, finding out by intensive study of the information seeking behavior. Real user participation in design situations, e.g. the development of new OPACs are rare, although there are some recent approaches in this direction.

User studies in library and information science have changed over the years. In the 1986 ARIST chapter on Information Needs and Uses, Dervin and Nilan described a forthcoming paradigm shift. They described the traditional paradigm, where the information is subjective, processed by the user and external observation is used for making propositions on the use of systems. “What” -questions are sought answers for by the use of quantitative methods. In the alternative paradigm, exemplified by three different approaches, the user is seen as the one who is constructing information, by using systems as one tool for their understanding. The focus is on the user, and “how”-questions are studied preferably with qualitative methods, since these parameters are not measurable. In 1990 Hewins validated this paradigm shift, and pointed at the cognitive approach as the next paradigm to come. And, in 1991, Allen wrote an ARIST chapter on cognitive research in information science.

If we look at other disciplines, such as Human Computer Interaction, which has evolved in a similar way, going from system orientation to cognitive psychology, there is a shift under way, turning from the individual and the machine towards a “social paradigm”, the user as she behaves as an individual in a certain situation, in relation to other people. This has emerged in the field of Computer Supported Cooperative Work (CSCW) where the importance of the surroundings and the relation to other people in different working situations are acknowledged as an important factor in system development. In this research area (CSCW) we find many examples of very profound studies of people in their every-day work, performed in order to guide the design of computerized support systems. Approaches in this direction in Library and Information Science are rare, but Ellis acknowledges this as he writes on the problems of choosing observation methods in the studies of information-seeking patterns of academic researchers - “as information seeking is integrated with the rest of their activities in a way that makes observation almost totally impracticable.”

The overall trend in these different research areas, is that the user is in focus, as part of a social setting, and the methods used to study her must, at least partly, be of a qualitative nature, in order to capture her needs and preferences in design processes.

The aim of the dissertation is to define and describe a formal method for the management of qualitative data, i.e. from user interviews. A comparative study of this formal approach, and some traditional tools for computer supported qualitative analysis, in the light of systems design, is also part of the work. The method is developed for design of bibliographic information systems, but might be generalized to support user-oriented design of large, public systems in general.

18.4.2.6 Networked information

The whole LIBLAB group has been involved in investigating this area, since it is very tightly connected to the undergraduate courses in Informatics. This is an area which will have great impact on our next research program. Activities during the years covered by this report, has mainly been devoted to initial studies and extensive use of Internet. Some results and discussions are presented in e.g. (Björklund et.al. 1995). Much effort has been put into the presentation of possibilities and limitations of the Net to a broad audience. This has taken the form of a number of open seminars, giving people a chance to get a first real contact with Internet.

18.5 Educational activities

During this period, a lot of effort has been put into education. The basic courses in Informatics has been further developed and a new lab for our students has been set up.

18.5.1 Undergraduate Education in Informatics

By the time of the last Activity Report, we had just started giving two basic courses in Informatics, Electronic Media I, 10p., and Information Resources in Societies I, 10p. This has now been extended, and this year, 1994-95, we have about 30 students, taking 20p. of each of these courses. We have also planned a third semester on Information Needs and Uses, to be started in this autumn. Altogether, we have then reached the point where our students are able to take 60p. in Informatics, and can after finishing their B.Sc. or M.Sc., go on to our doctoral program in Informatics.

18.6 Publications

External refereed publications:

Björklund, L. Thoughts about the cooperative work of creating tools for cooperative work. In: *An invitational workshop on Social Science Research, Technical Systems and Cooperative Work,* Paris, Mar. 8-10, 1993, pp. 77-83.

Persson, J. A Resource Based Approach to Generalization in the Context of GIS. In: *EGIS/MARI ‘94. Conf. Proc. Fifth European Conf. and Exhibition on Geographical Information Systems,* Paris, France, Mar. 29-Apr. 1, 1994. Eds.: J.J. Harts; H.F.L. Ottens; H.J. Scholten. Utrecht, EGIS Foundation, 1994. Vol. 1, pp. 142-150.

Björklind, A. An Architecture for Organising Dynamic Information About Space and Time. In: *Knowledge Organization and Quality Management. Proc. of the Third Int. ISKO Conf.,* Copenhagen, Denmark, Jun. 20-24, 1994. Ed. by H,. Albrechtsen and S. Oernager. Frankfurt/Main, Indeks, 1994. (*Advances in Knowledge Organisation*, Vol. 4, 1994) pp. 358-365.

Björklund, L. The Potential of Using Knowledge Organising Tools in Collaborative System Development. In: *Know ledge Organization and Quality Management. Proc. of the Third Int. ISKO Conf.,* Copenhagen, Denmark, Jun. 20-24, 1994. Ed. by H,. Albrechtsen and S. Oernager. Frankfurt/ Main, Indeks, 1994. (*Advances in Knowledge Organisation*, Vol. 4, 1994) pp. 264-271.

Hjerppe, R. A Framework for the Description of Generalised Documents In: *Knowledge Organization and Quality Management. Proc. of the Third Int. ISKO Conf.,* Copenhagen, Denmark, Jun. 20-24, 1994. Ed. by H,. Albrechtsen and S. Oernager. Frankfurt/Main, Indeks, 1994. (Advances in Knowledge Organisation, Vol. 4, 1994) pp. 173-180.

Björklund, L., Björklind, A., Hjerppe, R. Highways and Backroads of Internet: Strategies and Tactics. In: *Information Superhighway: The Role of Librarians, Information Scientists and Intermediaries. Proc. of the 17th Int. Essen Symposium,* Oct. 24-27, 1994. In press.

Björklund, L. Motorvägar och bystigar till information och upplevelse. In: *Aktuell forskning inom biblioteks- och informationsvetenskap,* FRN, 1994. In press.

Internal publications:

LiU-LIBLAB-WP:82 **Hjerppe, R. & Jungert, E.**: Generalization of Spatio-Temporal Information in Reactive Logistic Planning. *Jan., 1993, 6p.*

LiU-LIBLAB-WP:83 **Hjerppe, R. & Jungert, E.**: Meta-databases for Multimedia Spatio-Temporal Information Flow. *Jan., 1993, 5p.*

LiU-LIBLAB-WP:84 **Hjerppe, R.**: Införande och utvärdering av en personlig/gemensam katalog: TemaKat fas 3. *Feb., 1993, 6p.*

LiU-LIBLAB-WP:85 **Hjerppe, R.**: Användningen av MARC-fält vid sökning och visning. En kartläggning av Internet-kataloger. *Feb., 1993, 1 p.*

LiU-LIBLAB-WP:86 **Björklund, L.**: Datorstödd samverkan (CSCW) vid informationssökning och informationsanvändning. *Mar., 1993, 4p.*

LiU-LIBLAB-WP:87 Björklund, L.: Formella metoder för kvalitativa användarstudier. *Mar., 1993, 4p.*

LiU-LIBLAB-WP:88 **Björklind, A.; Hjerppe, R., & Jungert, E.**: Management of Dynamic Multimedia Spatio-temporal Information for Meta-Databases. *Mar., 1993, 5p.*

LiU-LIBLAB-WP:89 **Persson, J.; Hjerppe, R., & Jungert, E.**: Fusion and Generalization of Dynamic Spatio-temporal Information for Reactive Logistic Planning. *Mar., 1993, 4p.*

LiU-LIBLAB-WP:90 **Hjerppe, R.**: Meta-databaser och heterogen distribuerad information: användarens perspektiv. *Mar., 1993, 4p.*

LiU-LIBLAB-WP:91 **Björklund, L.**: Planning for the Evaluation of TemaKat. *Aug. 1993, 7p.* (Paper presented at ECSCW 93, Workshop on Evaluation Studies, Sep. 12, 1993, Milano)

LiU-LIBLAB-WP:92 **Hjerppe, R.**: Studier av användningen av IDAKat. En avslutning på TemaKat-projektet. *Jan., 1994, 6p.*

LiU-LIBLAB-WP:93 **Björklund, L.; Hjerppe, R.**: Visualisering: verktyg, metoder och data för användning i bibliotekssammanhang. *Jan., 1994, 5p.*

LiU-LIBLAB-WP:94 **Hjerppe, R.; Öhrman, M.**: Semi-intelligent filtrering av elektronisk post. *Feb. 1994,8p.*

LiU-LIBLAB-WP:95 **Hjerppe, R.; Jungert, E.; Ohlsson, K.**: Establishing and Managing Improvised Meetings. Improving Access to Heterogeneous Information Resources in Physical Planning. *May 1994, 6p.*

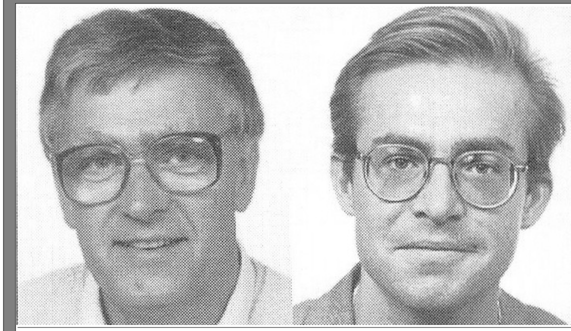
Other publications:

Andriessen, E., Hjerppe, R., Jungert, R., Kehrer, T., Lubich, H., Peiro, J.M., Rugelj, J., Santos, A., van der Velden, J. & Wilbur, S. Multimedia-Supported Cooperation. Concepts, Applications and Technical Support Requirements. *COST-14 Project CO-TECH, Working Group 3.* Jan., 1993. 113p.

LOGPRO

19
The Laboratory for Logic Programming

Foundations of logic programming
Integration of programming paradigms
Program transformation



Jan Maluszynski
director

Ulf Nilsson
director (94/95)

Some of the members of the Laboratory for Logic Programming.
Back row: Jan Maluszynski, Lars Degerstedt, Ulf Nilsson, Stefan Zemke, Johan Boye.
Front row: Fredrik Eklund, Magnus Andersson, Alessandra di Pierro, Andreas Kågedal.

19.1 Introduction

The research of the Logic Programming Laboratory (LOGPRO) focuses on foundations of logic programming and its relationship to other computational paradigms. The main areas of research have been: amalgamation of logic and functional programming, abstract interpretation and verification of logic programs, algorithmic debugging, constructive negation and the relationship between logic programming and various grammatical formalisms.

The following persons participated in LOGPRO research during 1993-94:

Leadership and staff

Jan Maluszynski, Ph. D., professor;
Gunilla Blom-Lingenhult, secretary;

Senior researchers

Wlodzimierz Drabent, Ph. D., associate professor;
Ulf Nilsson, Ph. D., assistant professor;

Graduate students

Magnus Andersson, B. A.;
Johan Boye, Ph. Lic;
Lars Degerstedt, Ph. Lic;
Fredrik Eklund, M. Sc.;
Andreas Kågedal, Ph. Lic;
Hans Olsén, M. Sc.;

Visitors

Jukka Paakki, Ph. D. (Univ. of Jyväskylä);
Alessandra Di Pierro, Ph. D. (Univ. of Pisa);
Tim Heyer, visiting scholar (Univ. of Braunschweig);
Stefan Zemke, visiting scholar (previous affiliation, Cambridge Univ.).

A total of 19 papers were published in reviewed journals and conference proceedings during the reporting period. Three licentiate theses and one co-authored book were also produced by members of the group (see publications).

19.2 Research Topics

The aim of the research of LOGPRO is to increase the usability of logic programs by focusing on specific areas of application and methods for improving the efficiency of such programs. Much of the research can also be characterized as the study of *relations* between logic programming and other areas of computer science, in order to facilitate transfer of concepts, methods and techniques between these areas. Below we focus on the *visions* and *results* of four major research projects during 1993-94.

19.2.1 Query-processing in deductive databases

It is well-known that logic programming may be viewed as an (extended) representation language for relational databases. Logic programs are therefore sometimes referred to as *deductive databases*. However, it is also recognized that standard query-processing techniques — such as that of Prolog — are often inadequate for database applications. The aim of our first project is to study the use of *fixed-point techniques* for *recursive query-optimization* in deductive databases. The principal investigator of this project is Nilsson. Engaged in this work are also Andersson and Degerstedt. Two major lines of research can be singled out:

Heuristic techniques for computation of fixed-points

Many problems in computer science, including that of recursive query-processing, may be viewed as fixed-point problems. Such problems are typically solved using variants of the, so-called, *chaotic iteration* strategy. The aim of this sub-project is to study techniques for improving convergence of chaotic iterations by taking various heuristics into account.

Query-processing in the presence of negation

Query-processing of positive logic programs is today a well-known topic. Several papers (including Degerstedt (1993) and Nilsson (1995)) have shown how the two main approaches — top-down execution with tabulation and bottom-up execution with magic templates — relate to one another. However, not so much attention has been devoted to *logic programs with negation*. As with positive programs two major approaches may be singled out:

- **Top-down strategies with tabulation**: Bol and Degerstedt (1993a) have provided a general scheme for top-down execution with tabulation for positive programs. The idea is based on a notion of a *search forest*. The main contribution of this notion is that it clearly separates between *search space* and *search strategy.* In a follow-up paper (1993b) they extend the idea to programs with negation with well-founded semantics as the underlying declarative semantics.
- **Bottom-up strategies with "magic rewriting"**: The other predominating approach to query-processing relies on a rewriting step (introducing so-called magic templates into the program) followed by bottom-up execution. Nilsson (1993) and Degerstedt and Nilsson (1994) generalize this approach from the positive case to logic programs with negation.

Degerstedt and Nilsson (1994) also show that the bottom-up approach and the top-down approach are closely related.

19.2.2 Dynamic properties of logic programs

The dynamic properties of logic programs studied in this project concern actual form of procedure calls and successes during the computation of the program. The objective is: to facilitate precise reasoning about such properties, to contribute to better compilation and to support the user by programming tools facilitating the construction of correct programs.

One aspect of our research concerns the use of assertions specifying dynamic properties for correctness proofs and for debugging. Another is efficient compilation of directional programs. The principal investigators are Drabent and Maluszynski. Engaged in the work are also Boye, Eklund and Kågedal.

Proving dynamic properties

One of the strengths of logic programming is that programs may be seen as sets of axioms and this view is sufficient to reason about program results. However the way programs are executed (the operational semantics) may be rather complicated and the dynamic properties are often difficult to understand.

In their paper "Inductive Assertion Method for Logic Programs" (Theoretical Comp. Sci., Vol. 59, 1988) Drabent and Maluszynski proposed a general method for reasoning about such properties. One specifies the properties by means of assertions, giving a precondition and a postcondition for every predicate/procedure. The method provides a way to prove that such a specification is correct.

General objectives of the present research are:

- to extend the method to the case of constraint logic programs and programs with delays;
- to provide a proof method for assertions assigned to program points, similar to the approach of Floyd and Hoare for imperative programs;
- to provide a precise comparison with other methods of dealing with dynamic properties;
- to devise ways of proof automation for nontrivial classes of programs.

To formulate extensions of the method one has to be precise about the semantics considered. The "grammatical view of logic programming" presented in the book by Deransart and Maluszynski (1993) gives a general framework for operational semantics. In this framework the execution of a (constraint) logic program is seen as two interleaving processes: construction of a skeleton tree, which is a parse tree of a context-free grammar and decoration of the nodes of the constructed portions of the tree. The decoration process consists of solving equations associated with the constructed skeleton. Different strategies of skeleton construction combined with different strategies of equation solving give rise to a space of different operational semantics. One of them is SLD-resolution with Prolog computation rules. Also, attribute evaluators used for implementation of attribute grammars can be expressed within the framework. This shows that attribute grammars are closely related to logic programs. In particular some concepts and proof methods developed originally for attribute grammars are also applicable to logic programs.

One of them is the notion of a dependency relation which can be used for modelling of data-flow during the execution of logic programs. It shows directional aspects of a given logic program, and gives some hints concerning the use of the program.

A restricted class of dynamic properties is captured by the notion of *directional type*. The objective of the present research is to develop principles for static checking of directional types based on the analysis of the dependency relation, and implement them in an experimental system.

A combination of directional type checking with algorithmic debugging techniques is a planned topic of future work.

Towards optimized compilation of directional logic programming

The research on this topic concerns the use of some dynamic properties of programs for optimization of their compilation. It focuses on two specific languages: GAPlog and Ground Prolog.

GAPlog is an extension of Prolog that allows for clean integration of logic programs with procedures written in non-logic-programming languages. The language emerged from a theoretical study of the problem. Its principles and its prototype implementation are described in Maluszynski et.al. (1993). The underlying principle is to delay the invocation of any external procedure until its arguments become ground. However, the overhead of the dynamic delays can often be avoided by analysis of the dependency relation of a given program. The relation can be automatically inferred (Boye et. al. (1993)), and used for optimizing the compilation of the program (see Boye (1993a) and Eklund (1994)).

Ground Prolog, proposed by Feliks Kluzniak, is a directional dialect of Prolog. The arguments of every predicate are classified as *inputs or outputs*. During the execution of the program the inputs of every goal selected for resolution are ground, that is contain no variables, as well as the outputs of the goal at success. The objective of the ongoing research is to develop principles of the optimized compilation of such programs. Preliminary results can be found in Kågedal (1993,1994).

19.2.3 Verification and synthesis of logic programs

One of the basic tools for proving properties of programs is mathematical induction. The problem raised by inductive reasoning is that one usually has to find a *lemma* that will make the induction step true. This lemma corresponds to the *loop invariant* of assertional verification of imperative programs, and the discovery of such an invariant (or lemma) is a highly nontrivial task.

There are several proposals in the literature suggesting that lemmas can be found using program synthesis methods. L. Fribourg and H. Olsén have investigated this approach and devised a method for extracting logic programs from proofs in order to use the program as a lemma. *Collage induction* of Olsén (1994) was introduced as an extension of *NFI-driven induction* proposed by Fribourg and Olsén (1992), and a method for extracting logic programs from proofs in *Extended Execution* was generalized. Collage induction splits a proof by mathematical induction into a proof of an *induction lemma* and a proof of an *induction goal*. From the proof of the induction lemma, a logic program is synthesized defining the relation between the hypothesis and the conclusion of the induction goal.

The above approach does not in general solve the problem of finding invariants. However, for a restricted class of problems, the least fixed-point of the program synthesized during the proof of the induction lemma, can be computed and represented finitely by a formula of linear integer arithmetic. In these cases, the formula computed represents the *strongest* invariant and any property one wishes to verify follows logically from this invariant. The proof of the induction goal is replaced by the proof of a simple logical inclusion. Basically, the inductive reasoning is encapsulated in the fixed-point computation.

This approach is related to *abstract interpretation*, where the aim, usually, is to compute overestimations approximating the least fixed-points of a general class of programs. In contrast, our goal is to generate expressions that exactly characterize the least fixed-point for a narrow class of programs. Our continued research will focus on extending the class of programs for which the least fixed-points can be represented and computed finitely.

19.2.4 Negation in logic programming

A usual way of dealing with negation in logic programming is negation as failure. It suffers a serious restriction: it can only be used as a check, it cannot produce answer substitutions. Approaches to overcome this limitation are termed constructive negation.

Drabent (1995) presents a constructive negation method based on building failed trees. For this we had to generalize the notion of a failed tree to the context of constructive negation. Our approach has two versions, for both main semantics for logic programs with negation: the Kunen semantics (3-valued completion semantics) and the well-founded semantics. The versions differ by disallowing or allowing infinite failed trees. For both cases Drabent (1995) provides soundness and completeness proofs.

A natural question is, what can be computed without constructive negation. The basic computational mechanism for negation as failure is SLDNF-resolution. It is incomplete due to above-mentioned limitation of negation as failure. Many papers studied classes of programs for which SLDNF-resolution is complete (see Drabent (1993b) for references). It is in some sense obvious that the reason for incompleteness is floundering (i.e. selecting a non ground negative literal in the computation; to such a literal negation as failure is inapplicable in most cases). However this fact has never been established formally before.

Drabent (1993b) proves that SLDNF-resolution is complete (w.r.t. the Kunen semantics) for arbitrary programs and queries that do not flounder under some fair selection rule. The proof is based on the results of Drabent (1995).

19.3 Miscellanea

International cooperation

- ESPRIT COMPULOG Network of Excellence;
- LOPSTR: Logic programming Synthesis and Transformation (ESPRIT Human Capital and Mobility);

- ABILE: Abstract Interpretation of Declarative Languages (ESPRIT Human Capital and Mobility).

Individual cooperation

- Laurent Fribourg (LIENS, Paris) spent six months with LOGPRO in 1991. He is now supervising Hans Olsén and participating in the project “*Verification and Synthesis of Logic Programs*”;
- Pierre Deransart (INRIA, Rocquencourt) with whom LOGPRO has had a long and fruitful cooperation. During 1994/95 Pierre Deransart is visited by J. Maluszynski;
- Jukka Paakki (Univ. of Jyväskylä) spent one year with LOGPRO and is continuing the cooperation.
- Wlodzimierz Drabent is also assistant professor at the Institute of Computer Science of the Polish Academy of Sciences, Warsaw.

Awards

- In 1993 U. Nilsson and S. Bonnier (previously in LOGPRO) received the Chester Carlson Research Prize. The award was presented by the Royal Academy of Engineering Sciences in recognition of the contribution of their Ph. D. theses.

Program committees of international conferences

- “Programming Languages, Implementations, Logics and Programs”, PLILP95, Utrecht (Maluszynski);
- “Int’l Conference on Logic Programming”, ICLP95, Tokyo (Drabent);
- “Automated and Algorithmic Debugging”, AADEBUG’95, St. Malo, France (Drabent, Paakki);
- “Static Analysis Symposium”, SAS94, Namur, Belgium (Nilsson);
- “Int’l Conference on Logic Programming”, ICLP94, S. Margherita, Italy (Maluszynski);
- “Int’l Conference on Logic Programming”, ICLP93, Budapest (Drabent, Maluszynski);
- “Automated and Algorithmic Debugging”, AADEBUG’93, Linköping (Maluszynski);
- “Programming Language Implementation and Logic Programming”, PLILP’93, Tallin, Estonia (Drabent).

19.4 LOGPRO publications 93-94

19.4.1 Licentiate theses

Olsén, H. (1994). *Collage Induction: Proving Properties of Logic Programs by Program Synthesis*. Licentiate thesis No. 464, Linköping Univ., Dept. of Computer and Information Science.

Degersted, L. (1993). *Tabulated Resolution for Well-founded Semantics*. Licentiate thesis No. 402, Linköping Univ., Dept. of Computer and Information Science.

Boye, J. (1993b). *Dependency-based Groundness Analysis of Functional Logic Programs*. Licentiate Thesis 398, Dept. of Computer and Information Science, Linköping Univ.

19.4.2 Master's theses

Alexandersson, J. (1993). *Abstract Interpretation of Logic Programs with External Functions: An Implementation*. Master's thesis LiTH-IDA-Ex-9323, Dept. of Computer and Information Science, Linköping Univ.

Eklund. F. (1994). *Synthesis of Directionality Information for GAPLog Programs*. Master's thesis LiTH-IDA-Ex-9418, Dept. of Computer and Information Science, Linköping Univ.

Heyer, T. (1994). *Debugging in GAPLog: A Model for Execution of Logic Programs with Delayed Function Calls.* (In Swedish.) Master's thesis LiTH-IDA-Ex-9441, Dept. of Computer and Information Science, Linköping Univ.

19.4.3 Refereed publications

Bol, R. Degerstedt, L. (1993b). Tabulated Resolution for Well-Founded Semantics. In D. Miller, ed., *Proc. of the 1993 Int. Logic Programming Symp.*, MIT Press.

Bol, R. Degerstedt, L. (1993a). The Underlying Search for Magic Templates and Tabulation. In D.S. Warren, editor, *Proc. of 10th International Conference on Logic Programming*, pp. 793-811, MIT Press.

Boye, J. (1993a). Avoiding Dynamic Delays in Functional Logic Programs. In M. Bruynooghe and J. Penjam, eds., *Programming Language Implementation and Logic Programming 93*, LNCS 724, pp. 12-27. Springer-Verlag.

Boye, J., Paakki, J., Maluszynski, J. (1993). Synthesis of Directionality Information for Functional Logic Programs. In P. Cousot, M. Falschi, G. File, and A. Rauzy, eds., *Proc. of the Workshop on Static Analysis, WSA'93*, LNCS 724, pp. 165-177, Springer-Verlag.

Degerstedt, L., Nilsson, U. (1994). Magic Computation for Well-founded Semantics. In J. Dix, ed., *Non-monotonic Extensions of Logic Programming*, LNAI, Springer-Verlag. To appear.

Deransart, P., Maluszynski, J. (1993). *A Grammatical View of Logic Programming*, MIT Press, 1993.

Drabent, W. (1995). What is Failure? An Approach to Constructive Negation. *Acta Informatica*, 32(1), pp. 27-59.

Drabent, W. (1993b)]. Completeness of SLDNF-Resolution for Non-floundering Queries. In *Proc. Int. Logic Programming Symp*. Poster abstract. (Extended version submitted for publication.)

Drabent, W. (1993a). SLS-resolution Without Floundering. In L. M. Pereira and A. Nerode, eds., *Proc. 2nd Int. Workshop on Logic Programming and Non-Monotonic Reasoning*, pp. 82-98, MIT Press.

Eklund, F., Kågedal, A. (1994). Optimization of GAPLog Programs. Presented at *10th Logic Programming Workshop*, Zurich, Switzerland.

Fribourg, L. Olsén, H. (1992). A Unifying View of Structural Induction and Computation Induction for Logic Programs. In *Proc. of Logic Program Synthesis and Transformation (LOPSTR 92)*. Springer-Verlag. Workshops in Computing series.

Fribourg, L. Olsén, H. (1994). Direct, Dual and Contrapositive Proofs by Induction. In *Proc. of International Workshop on Logic Program Synthesis and Transformation, LOPSTR'94*. Preprints, Pisa University.

Kågedal, A. (1993). Improvements to Compile-Time Analysis for Ground Prolog. In M. Bruynooghe and J. Penjam, editors, *Programming Language Implementation and Logic Programming 93*, pp. 92-107. Springer-Verlag. LNCS 724.

Kågedal, A. (1994). A Dataflow Analysis Method for Ground Prolog. In Pascal Van Hentenryck, editor, *Proceedings of the 11th International Conference on Logic Programming*, pp. 746-747. The MIT Press. Poster abstract.

Kuchcinski, K., Drabent, W., Maluszynski, J. (1993). Automatic Diagnosis of VLSI Circuits Using Algorithmic Debugging. In Peter Fritzson, editor, *AADEBUG'93, Proc. of the 1st Int. Workshop on Automated and Algorithmic Debugging*, pp. 350-367. Springer-Verlag. LNCS 749.

Maluszynski, J., Bonnier, S. Boye, J., Kågedal, A., Kluzniak, F., Nilsson, U. (1993). Logic Programs with External Procedures. In K.R. Apt, J.W. de Bakker, and J.J.M.M. Rutten, eds, *Logic Programming Languages, Constraints, Functions, and Objects*, pp. 21-48. The MIT Press.

Nilsson, U. (1995). Abstract Interpretation: A Kind of Magic. *Theoretical Computer Science*. To appear.

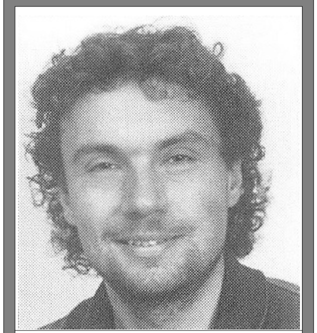
Nilsson, U. (1993). Goal-directed Bottom-up Evaluation of Normal Logic Programs. In *Proc. of the International Logic Programming Symposium*, Vancouver. MIT Press. Poster abstract.

Paakki, J. (1994). Multi-pass Evaluation of Functional Logic Programs. In *Proc. of 21st ACM Symposium on Principles of Programming Languages*. The ACM, Portland, Oregon.

MDA

20 People, Computers, and Work

Computerization in work life
Computer Supported Cooperative Work
Hypermedia
Action Research
Qualitative methods



Toomas Timpka
director

Some of the members of the group for People, Computers and Work.
Back row: Niklas Hallberg, Toomas Timpka, Mikael Johansson.
Front row: Cecilia Sjöberg, Inga-Lill Andersson, Lena Bjerlöw, Ewa Rauch.

20.1 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. Work-life informatics as a research area includes:

- the role 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;

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.

20.1.1 Action research

The notion of action research is the basis for the activities of the MDA-group. In action research, the aim is to establish forms and processes of development that people can use to change their own lives and working conditions. 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 and accepted experts; and
- cyclical research process. Description, action, and evaluation phases succeed each other in an uninterrupted cycle.

For this to be performed, a strong connection between social structure, design and organizational action is assumed. Hence, experiences from a variety of development activities are fed into the discussion of information system design, evaluation and use.

20.1.2 Computer Supported Cooperative Work and Hypermedia

The principle behind hypermedia is to use computers to easily make and share links between pieces of stored information, let this information consist of text, images, graphics or sound. With hypermedia technology, the user is in total control of the interaction with the computer. Consequently this technology, more than most, can be made to fit into habitual work practices and particularly, 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 of 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 gives the potential to design computer systems which address issues that 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 practising in the application field. We have in the MDA-group found reflection through action research, essential to find the forms for such an inter-disciplinary development process.

20.1.3 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 is 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.

20.1.4 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 of capturing observable behavior. However, one major problem lies in analysis of the recorded material. Use of qualitative methods are from our perspective an interesting alternative. 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 in many situations is the most efficient medium for documentation and communication of ideas.

20.2 The MDA group

The members of the MDA-group at the Departments of Computer Science and Community Medicine are (December 1994):

Toomas Timpka M. D. Ph. D. (director)
Henrik Ericsson Ph. D. (co-director)
Ingalill Andersson (secretary)
James M. Nyce Ph. D. (visiting associate professor)
Kent Lindquist Ph. D.
Cecilia Sjöberg Ph. Lic. (Ph. D. student)
Lena Bjerlöv B. Sc. (M. Sc. student)
Niklas Hallberg M. Sc. (Ph. D. student)
Mikael Johansson B. A. (Ph. D. student)
Kristina Kinnunen B. Sc. (Ph. D. student)
Tom af Klercker M. D.
Svein-Ivar Lillehaug M. Sc. (Ph. D. student)
Ewa Rauch M. Sc. (software engineer)

The MDA-group has during 1993-94 been a partner of the EU/AIM (Advanced Informatics in Medicine) consortium A2005 “DILEMMA”. This consortium involves partners from England, France, Germany, Portugal, and the Netherlands, and is coordinated 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 resulted in a partnership for the MDA-group in the EU/ESPRIT BRA (Basic Research Action) consortium 7040 “AMODEUS II”. During 1993-94, the MDA-group has also continued its research activities at the national level, and thus initiated projects in the NUTEK ‘Information Systems’ program as well as in the NUTEK/AMFO ‘Computer Supported Cooperative Work’ program.

20.3 Research 1993-94

The MDA-group is presently engaged in two research areas: Computer Support for Cooperative Work (CSCW) in the MEDEA/PRIMUS 2000 projects, and development of methods for participatory software design to be used in work life in the DILEMMA /Argumentative Design projects.

Computer Support for Cooperative Work and hypermedia in Primary Health Care.

From 1988, the main focus for the MDA-group has been to study and develop computer support for the work of a primary health care team. The first phase of the project, “Datorstö(r)d” or “MEDEA”, was reported in July 1992 by, in addition to scientific reports, production of a video, exhibition of a prototype at a national computer fair

(Data Office '92), and numerous articles in newspapers and trade journals. In 1993-94, the development work continued in cooperation with the staff at Torkelbergsgatan Primary Care Center, Linköping. The interaction between the patient and a health care team was chosen as the point of departure for the PRIMUS 2000 project, being the critical common source for the continuous improvement of service quality for the patient and work content for the care providers. The project addresses both the organizational and the technological aspects of health care services development in a primary health care context. It builds on experiences gained from the MEDEA and the Codesk/KnowledgeNet projects performed at the Royal Institute of technology, Stockholm.

20.3.1 PRIMUS 2000

In most countries, health care is the largest branch of the public sector, both in terms of spending and employees. To manage a constantly growing demand, national health care policies focus on strenghtening primary care and installing health care markets. In parallel, the internal organization of health care services have been made the object of scrutinization and dynamic change, because health care, like other service organizations, struggles against weak development of productivity. In service organizations, clients' subjective experiences are brought forward together with staff climate, since progress is measured against output of an 'acceptable service quality' only. Another central issue concerning services with relevance for health care is the insufficient utilization of information technology. As a complement to the traditional perspective on information systems in health care as supporting transaction processing, PRIMUS 2000 is developed an alternative framework based on the needs of a market-sensitive service organization (Timpka & Marmolin 1994, Timpka & Marmolin 1995). Information technology is planned to be introduced in the project in three steps:

1. efficiency increase by introduction of computerized medical records;
2. support of the total quality of the existing services by supply of medical "evidence", care plans and "telematic-based" quality circles; and
3. development of new distributed health care services near the patient.

Here, patients are no longer seen as targets, but instead engaged contributors and reformers in health care. Therefore, they are in the design of the PRIMUS 2000 environment brought in focus and regarded as active consumers with direct influence on the structure of the health care service. The visualization of patient-related work-flow and emphasis on teamwork may show to be the difference from the abstract entity-relationship paradigm that make information systems truly support productivity in health care. Yet, the introduction of patient-centered CSCW in health care is not a well-marked innovation which is introduced without difficulty. Office automation has passed several distinguishable stages during its development, but only recently the technology and underlying infrastructures have together reached a level of 'work-group computing' which factually can support the social and organizational goals that these systems have originally had. However, when CSCW begins to gradually appear at workplaces, it also requires the workplace to appear in the in CSCW systems. For instance, the structure of management decision-making must not be violated in the system. Also, many present CMR systems in primary care are inflexible and do not allow for the following of other than standard areas, such as child immunizations and cervical cytology. Therefore, to secure the utility of CSCW in health care, it is necessary not to discontinue the input of requirements from practice.

The PRIMUS 2000 design of computer-supported organizational learning should not be considered as a suggestion for a sudden revolution of medicine, but instead as a necessary operational model for an integrated response to the external pressures markets and governments have laid on health care organizations and practitioners. It links key decisions taken at a health care unit intended to have impact on financial performance and working conditions to assurance of clinical and patient-satisfaction outcomes, and thus only brings together concepts already well-known in medicine, with established knowledge from business administration and work sociology. Nevertheless, a broad evaluation of its long-term effects will be the next step in the development of the PRIMUS 2000 exemplar for organizational learning.

20.3.2 The MEDEA clinical hypermedia system

Development of MEDEA has showed that a focus on pragmatics leads medical hypermedia design in new and unexpected directions. For example, before the exploratory phase in the development process, the designers had not expected to have to focus the system design on transfer of text and images from paper to hypermedia. In fact, during the project the content of the corpus, and in particular the linking within the hypermedia corpus was found to be the bridge between practice and context, and technical design. Hypermedia links were not only to be easy to find and rapid to follow, but every link was to be adapted to situations encountered during work – for the informed selection of one action over another. Specifically, the supply of locally validated "evidence-based" knowledge was what the technical design had to deliver and maintain. Further, the early phases of the study also showed that hypermedia sources in health care need to be incorporated into and support collective care activities where co-operation and work-sharing occur between different categories of practitioners. Integration of present medical hypermedia systems with electronic communication facilities and shared databases towards groupware is thus to be anticipated. Concerning MEDEA, the system is presently built into the PRIMUS 2000 environment for Computer-Supported Co-operative Work (CSCW) including, e.g., electronic conferencing and a computerized medical record.

In the experimental phase of the MEDEA design, it was found important also for medical hypermedia that the access points to the hypermedia corpus reflect the way practitioners use and want access to information. Moreover, the introduction of an overall structure to the non-sequential task level of the corpus, with overview nodes as part of a defined structure, was found necessary to avoid loss of orientation. The defined corpus structure thus enhances usability in two ways. First, it helps structure corpus entrances so the corpus supports a problem-oriented approach to clinical decision-making. Second, it helps manage and maintain the corpus, since changes in work tasks and routines have to be reflected in the information material. In the MEDEA design, information regarding the structure of the corpus is integrated within the corpus content. Other designs however suggest separation between a "browser" and the source text and images. Yet, to resolve this design question, and other issues regarding the semantics and syntax of medical hypermedia, further study of how clinical texts are read and used in practice by both novice and experienced practitioners is needed.

20.3.3 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 has been to extend the available development methods to Action Design, where an interdisciplinary group functions as a reference source for system development.

Action Design is also a “methodological toolbox” for requirements engineering inspired by Activity Theory. It includes detailed recommendations at 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. Activity theory is introduced as a theoretical frame of reference for these phases, which are crucial for meeting demands on the performance of information systems in medical settings.

20.3.4 The need for requirements engineering in software development

Insufficient requirements analysis has often been suggested as a problem in software development. Therefore, to explore knowledge, practices and attitudes, key software professionals from four countries were surveyed regarding the early phases in the software development process (Timpka & Johansson 1994). For data collection, semi-structured interviews were performed and video-taped. As reference, documentation of the Action Design methodology was used. Two separate qualitative analyses of the data were made. The first was to identify central concepts and attitudes related to requirements engineering. In the second analysis, concrete circumstances surrounding decisions to use requirements engineering methods were investigated. The study concluded that investigations are needed to further analyse the problems software development projects have to deal with in practice. This knowledge can then be used to develop methods and tools which support design practices where existing structural, procedural, and building on this, technical obstacles are overcome.

20.3.5 Argumentative Design

The aim of Argumentative Design (ArD) is to modify and integrate the design rationale approach to capturing of the argumentation during a design process into a feasible method for collaborative requirements specification in working-life (Sjöberg 1994). The development of ArD has a technical component in the development of visualization methods for argument structures, and an empirical component in the study and evaluation of requirements analysis processes in industry and public sector settings.

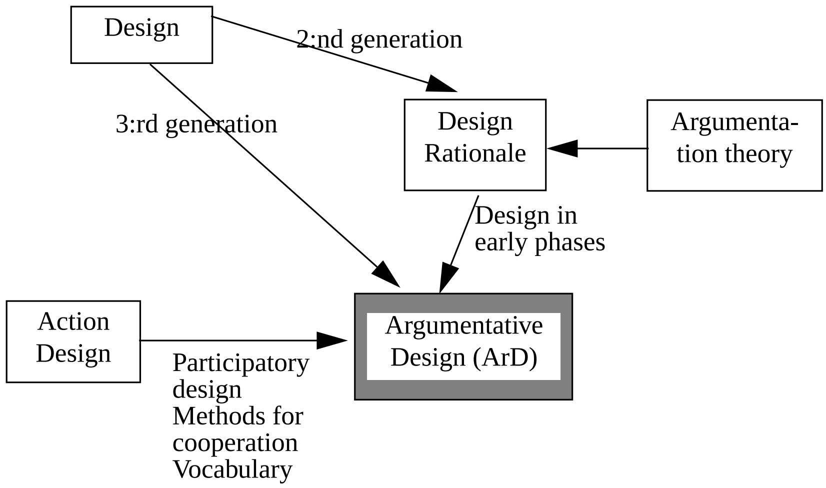


Figure 20-1. *Theoretical contributions to ArD*

In detail, the prototype design rationale approach, Argumentative Design (ArD), has been empirically refined and situated in the academic literature (Figure 20-1). In addition, data has been collected from a series of design meetings of the PRIMUS 2000 project (Figure 20-2).

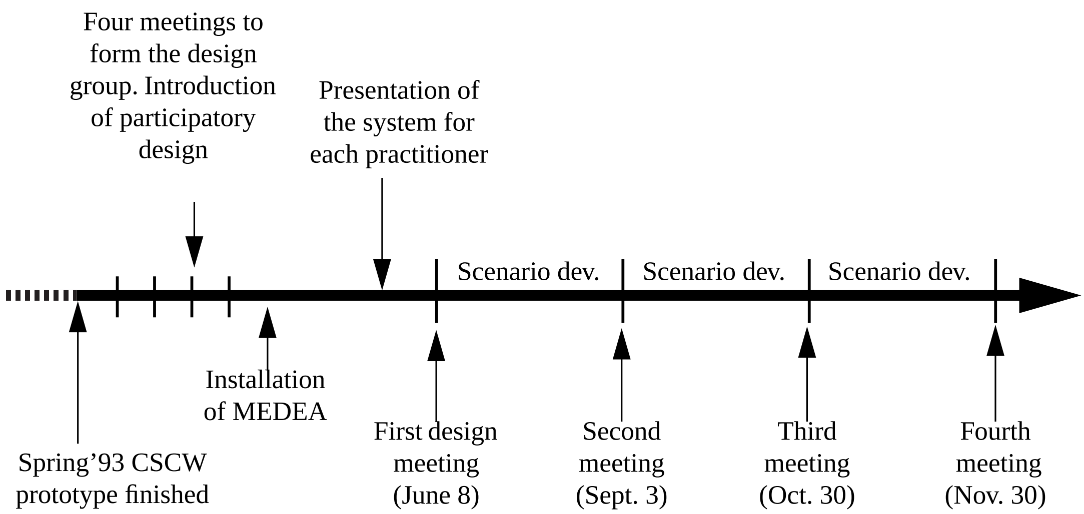


Figure 20-2. *Timeline for data collection performed 1993-94*

As a result, the present version of ArD is based on both the Scandinavian tradition of systems development as well as Retell's ideas regarding a "second generation" of design methodologies. It can be viewed as an extension of other design rationale approaches in mainly two ways. First, it is a widening of the perspective. We see the main usage of ArD in the earlier phases of system development. It is not to be used in the design specification or specifically in the Human-Computer Interaction area. Second, ArD expands the concepts involved. Objectives, problems, and needs in the customer organisation are considered in parallel with technical possibilities, and with the help of visualization of possible alternative consequences, an argumentation for the requirements is composed. The final ArD method will provide mechanisms which not only encourage debate but also result in productive closure. Previously available design rationale approaches are thus extended in three important ways. Since it is aimed at use in multi-disciplinary groups (Sjöberg & Timpka 1994), ArD is more inclusive and attends to local process factors that only problem based approaches do not. Second, it considers strategic planning and customer quality aspects (Timpka & Sjöberg 1994). Third, because it supports “levelling the playing field” in the documentation of requirements argumentation (Timpka, Sjöberg & Svensson 1994), as it brings into the systems process points of view that otherwise might not be heard or in other ways excluded.

20.3.6 Requirements engineering for Knowledge Based Systems

In the DILEMMA consortium, the MDA-group has developed an approach towards Knowledge Based Systems (KBS) development which emphasizes fitting into the clinical organization, utility and safety. To do this, the Logic Engineering technique developed at ICRF, London, and Action Design methods were analysed and integrated into the DILEMMA method for system requirements engineering and design (Timpka, Rauch & Nyce 1994). The theoretical foundation making this integration possible was found in activity theory, which holds a notion of both shared cognition and situated organizational practice. This theory has previously been used, for instance, to model 'organizational memory' in primary health care. The main contention from the study is that in the DILEMMA method, Action Design contributes to the integration tools for organizational validation (Are we building the right system, according to the organizational needs?), while Logic Engineering provides the methods for system soundness and safety (Are we building the system right, technically?).To discuss the possibilities and potential of the approach, it has to be compared in detail against the needs for improvement of KBS methods:

Use of a coherent methodology for modelling of clinical problems. Action Design provides to the DILEMMA method for problem detection video and critical incident instruments, a problem structuring instrument, and the ArD instrument for problem analysis. In addition, to support structuring of clinical problems, recently a taxonomy of problems occurring during daily clinical work has been reported by other authors. The problem model is related to a health care business model and organizational goals. Together these tools give a comprehensive approach to problem modeling in most clinical settings.

Clarification of needs through rapid prototyping. Logic Engineering with its direct connection between specification and design provides a platform for early user testing of KBS software. Here Action Design contributes with tools for user interface design, essential in the clinical workplace setting. It is necessary to remember that prototyping for needs clarification is part of the system requirements phase, and should be separated from the system design phase.

Use of appropriate methods and models. Group knowledge modeling though concept mapping techniques have been used in the KBS area as well as in health care practice, so experience from use of the method in situations similar to that described in this paper is available. Yet, the concrete approach needs to be evaluated in practical system development settings. An important point to consider here is the external representation format, that is the design of the graphical language describing the models. Only if the semantics of the external model can be communicated to care providers, and if this is possible remains to be empirically tested, then can group concept mapping be considered an appropriate modeling method.

Evaluations as a part of design. Integral evaluation, that is testing, validation and verification during systems development, is fundamental to both Logic Engineering and Action Design. The latter two measures are especially important in safety-critical environments, such as, intensive care. To take advantage of material from repeated evaluations, the design knowledge in the DILEMMA method is documented in the ArD design rationale. Only this way can prototyping become a structured means for evolutionary systems development in a software life-cycle model.

A professional approach to maintenance and support. Over a software life-cycle, maintenance is known to cause 60% of total software costs. However, the DILEMMA method described here is basically a requirements and design specification method. Yet, it supports maintenance in that it suggests integrated documentation of both the evolving software product and of the development process. Further, maintenance costs have recently been found to be related to software complexity. Software correctness problems are prevented in the DILEMMA method, basically, through the formal logic-based verification and validation in Logic Engineering.

20.3.6.1 Experiences from Action Design use

A general experience from employment of Action Design in the MEDEA design process is that the process indirectly came to visualize the tradition for technology development in working-life in that the method challenged the habitual routines (Timpka, Sjöberg & Svensson 1994). Similar experiences have been reported from other co-development and participatory design projects in non-health care settings. In co-development, practitioners are brought into product design processes to adjust work practices in the customer organization in order to properly apply a system, and thus reduce system design cycle time. However, product designers using these methods have been reported as being seen as “outlaws”, breaking traditional organizational boundaries, and finding themselves in loyalty conflicts between their employers and the cooperating practitioners. Therefore, to be able to take advantage of the possibilities inherent in product co-development, it has been suggested that the entire organizational structure surrounding software design needs to be changed. The difference between product co-development and the use of Action Design in the MEDEA project is that in the latter, the structure for the practitioner participation, from the very beginning, was negotiated into the product development process at management level in the design organizations, and through agreements between management and employees in co-determination groups in the health care organization. In effect, the MEDEA process has moved further from product co-development to participatory design. In participatory design, practitioners are directly involved in the system design, which could be seen to imply even more complicated structural problems. Nonetheless, as opposed to previous participatory software design efforts, the

MEDEA project produced a product and thus “succeeded” as more than a mere local process. Several differences to previous projects can be identified to explain this. First, contrary to these projects, all categories of practitioners in the health care organization participated, which may have relieved tensions surrounding the design process. Second, qualitative studies of work practices were performed before any design attempt was made. Very few “surprises” arose during the later design phases, and the design team often had a more complete overview of work practice in primary care and its problems, than had any individual practitioner or manager. Third, a respectful attitude was taken towards the existing hierarchies, decision structures, and communication channels in working-life. Thus, no major decision was considered final in the design group if it had not been propagated and validated in the surrounding design and health care organizations. Yet, the most important factor might have been that separate product development activities were performed. The organization of marketing strategies for products from participatory design seems to be the issue which most immediately needs more attention for participative strategies to succeed also at the market level. A modified version of a structured product specification method, Quality Function Deployment, is therefore underway to be included in Action Design (Johansson 1994).

20.3.6.2 Methodological considerations

In action research, scientific rigor and relevance are central, but sometimes, competing concepts. There is, per definition, a striving for relevance as the focus is on analysis of action in naturalistic settings. To avoid unnecessary interference from scientific instruments in these settings, rigor is mainly forced onto the interpretation of data by feeding back all inferences to be reviewed by the participants/actors in the study. Yet here, criticism of action research has been formulated for not sufficiently building and reporting the entire background, including intention, of the interventions, which thus makes it impossible for outside readers to draw conclusions from their outcomes. In our studies, however, the Action Design method documentation has been used to specify project plans, with interventions and their background described for each design phase. Thus, a “social intervention experiment” can be said to have been performed, which allows for discussions of social outcomes and consequences on more than an ad hoc basis. In other words, even though they were performed in naturalistic settings, the study of design decisions was made in a defined social structure, thus making laboratory-type control of social interaction obsolete.

20.3.6.3 People, computers, work and health care services

In several recent articles and editorials, the question of why the dissemination of medical Decision Support Systems (DSSs) is progressing slowly has been discussed. Heatherfield and Wyatt have suggested that clinical requirements and problems should be given more attention, while Fox has emphasized engineering aspects and formally specified designs to increase validation and safety. Yet, it is hard to see that any of these proposals would radically increase the dissemination rate. It may be these authors might be searching in the wrong place for solutions and explanations (Timpka 1994). The reason for the slow dissemination is perhaps not to be found within human-computer interaction, software engineering or even within clinical medicine. It may, instead, be a natural consequence of the priorities which need to be made when organizing health care services. The dissemination pattern may come from the services, the stages in which technical support is introduced are reversed in comparison to traditional product manufacturing. Hence, new technologies are used to support the efficiency and quality of an existing service before development and support of new “service products” is considered. To consider this argument, first, the significance of health care as a service organization has to be honoured. Then reports from other areas within the service sector can be used to frame the process of medical DSS dissemination. Awareness of the “reversed product cycle” in services and its implications for technology development could then lead to better informed medical DSS development projects and more realistic expectations as to their outcome. The present experience from PRIMUS 2000 is that in order to come to practical use, development of medical DSS has to be considered realistic by all and not only by the research community. More importantly, the new technology has to be accepted, sanctioned and maintained by health care practitioners, managers, and relevant sectors of the computer industry. Neither, researchers emphasising requirements engineering, nor design validation and verification are wrong, but today the service aspects of health care are more important to take into account for DSS development and dissemination. It is necessary to stop thinking in terms of “technical products”; instead, medical DSS development should be planned in the wider context of services efficiency and quality, and ultimately, in terms of entirely new types of health care products.

20.4 Courses and supervision

The MDA-group provides post-graduate courses and supervision of Master's students in our fields of interest. The intent is to bring to the students an inter-disciplinary approach to research, and to expose them to problems faced by practising professionals outside the University. 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 is provided, 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. For the academic year 1994/95, a new course, Participatory Design, is planned.

20.5 MDA publications 1993-94

Licentiate thesis:

Sjöberg, C. *Voices of design: argumentation in participatory development*. Licentiate Thesis. Linköping: Department of Computer Science, Linköping University, 1994.

Journal articles:

Kinnunen, K., Timpka, T., Forsum, U. Decentralised diagnostics of beta haemolytic streptococci group A - introduction of a developmental model for microbiological consultations in primary health care., *Scand. J Prim Health Care 1995*. In press.

Nyce, JM., Timpka, T. Work, Knowledge and Argument in Specialist Consultations: Incorporating Tacit Knowledge into System Design and Development. *Med Biol Eng Comp*, 1993;31:S13-7.

Timpka, T. The health care crisis: End or beginning of the hay-day of medical decision support systems. Editorial. *Meth Inf Med*, 1994;33:332-5.

Timpka, T., Hensing, G., Alexanderson, K. Dilemmas in sickness certification among Swedish physicians. *Euro J Pub Health 1995*. In press.

Timpka, T., Johansson, M. The need for requirements engineering in the development of clinical decision support systems: a qualitative study. *Meth Inf Med*, (33):227–233.

Timpka, T., Rauch, E., Nyce, JM. Towards productive knowledge based systems in medicine: a methods perspective. *Artificial Intelligence in Medicine*. 1994;6:501-19.

Timpka, T., Sjöberg, C., Svensson, B. The pragmatics of clinical hypermedia: experiences from five years of participatory design in the MEDEA project. *Comp Meth Prog Biomed 1995*. In press.

Book chapters and papers in conference proceedings:

Alendahl, K., Timpka, T., Sjöberg, C. Computerized knowledge bases in primary health care: a curse or a blessing for health promotion, prevention and patient quality? *Proc. of MedInfo ‘95*, Amsterdam, The Netherlands, Elsevier, 1995. In press.

Johansson, M. Identification of quality characteristics for methods in DSS requirements engineering using Quality Function Deployment (QFD). In *Proc. of IRIS’94*, Univ. Press, Oulu, Finland.

Sjöberg, C., Timpka, T. Participatory design in practice: forgetfulness as a design resource. In Bradley, G., ed., *Human Factors in Organizational Design and Management IV,* Elsevier. Amsterdam, The Netherlands.

Sjöberg, C., Timpka, T. Inside multi-disciplinary design in medical informatics: experiences from the use of an argumentative design method. *Proc. of MedInfo ‘95*, Amsterdam, The Netherlands, Elsevier, 1995. In press.

Timpka, T. Teknik och lärande i serviceverksamheter: Datorstöd för primärvården. I: Lennerlöf L (red.). *Människor, datateknik, arbetsliv*, Stockholm, Allmänna Förlaget, 1993.

Timpka, T. Organizational learning in the continuous development of health care: making use of information technology to increase the total service quality. In Bradley, G., ed., *Human Factors in Organizational Design and Management IV,* Elsevier, Amsterdam, The Netherlands.

Timpka, T. Situated clinical cognition. Editorial. *Artificial Intelligence in Medicine.* 1995;7. In press.

Timpka, T., Arborelius, E. The primary health care nurse’s dilemmas: a study of knowledge use and need during telephone consultations. In Smith JP. ed., *Readings in advanced nursing,* pp. 127–142. Oxford, England, Blackwells Scientific.

Timpka, T., Marmolin, H. Expertext systems at ambulatory clinics. In *Proc. of World Congress on Expert Systems’94*, New York, Pergamon Press. (Invited paper.)

Timpka, T., Marmolin, H. Beyond computer-based clinical remainders: improvement of the total service quality by small-group based organizational learning in primary care. *Proc. of MedInfo ‘95*, Amsterdam, The Netherlands, Elsevier, 1995. In press.

Timpka, T., Nyce, JM., Sjöberg, C., Johansson, M. Action Design: From Modeling to Support of the Software Process. Blum BI, Chang L (eds.). *Proc. of Software Engineering and Knowledge Engineering '93,* San Francisco, USA, IEEE Press, 1993.

Timpka, T., Nyce, JM., Sjöberg, C., Rehnwall, H., Herbert, I. Bar code technology in health care: Using a business model for study of technology application and dissemination. *Proc. of MIE '93*, Elsevier, Amsterdam, 1993.

Timpka, T., Sjöberg, C. Voices of design: the dynamics of participatory information system design. In *Proc. of Participatory Design ‘94*, Chappel Hill, CPSR.

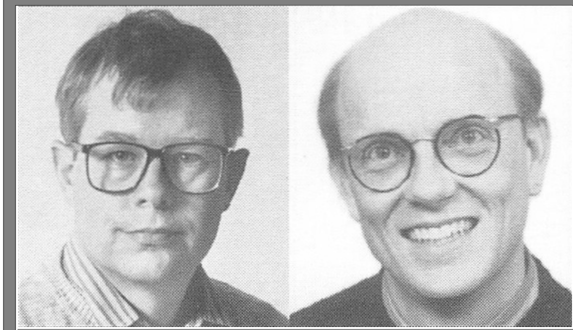
Timpka T., Svensson, B., Molin, B. (1995). Development of community nursing: analysis of the central services and practice dilemmas. *Int J Nurs Stud 1995*. In press.

Reports:

Timpka, T., Sjöberg, C., Alendahl, K., Johansson, M., Rauch, E., Hallberg, N., Herbert, I. Action Design version 2.1. CEC/AIM “DILEMMA” Report, Linköping, The MDA-group, Linköping University, 1994.

NLPLAB

21 The Laboratory for Natural Language Processing



Lars Ahrenberg
director

Nils Dahlbäck
director (94/95)

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

Some of the members of the Natural Language Processing Laboratory.
Back row: Nils Dahlbäck, Svetlana Linkova.
Front row: Peter Ingels, Barbara Ekman, Åke Thurée, Magnus Merkel, Stefan Svenberg, Lena Strömbäck.

21.1 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).

We collaborate with the following partners in our research; SICS, Ellemtel (The PUSH project), Word Work AB, Modus Språkteknologi AB, IBM Sweden, Microsoft Ireland (The Translation Support project).

NLPLAB is a member of the European Network on Excellence in Language and Speech (Elsnet)

Laboratory leadership, administrative and technical staff:

Lars Ahrenberg, Ph. D., professor (on leave 94-95)
Nils Dahlbäck, Ph. D., acting professor (94-95)
Barbara Ekman, secretary
Bernt Nilsson, research engineer

Employed research staff:

Nils Dahlbäck, Ph. D., assistant professor
Richard Hirsch, Ph. D., lecturer
Arne Jönsson, Ph. D., assistant professor (on leave 94-95)
Åke Thurée, M. Sc., research assistant
Mats Wirén, Ph. D., lecturer (94-95)

Graduate students:

Peter Ingels, B. A., M. Sc.
Magnus Merkel, B. A., Fil. lic
Lena Strömbäck, M. Sc., Fil. lic
Stefan Svenberg, M. Sc.

Guest researcher:

Svetlana Linkova, M. Sc. (94-95)

21.2 Overview of Current Research

21.2.1 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 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 dialogues between humans.

21.2.1.1 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 are also concerned with syntactic constructions, the speech acts likely to occur in interactions with the system and the ways in which 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 21-1.

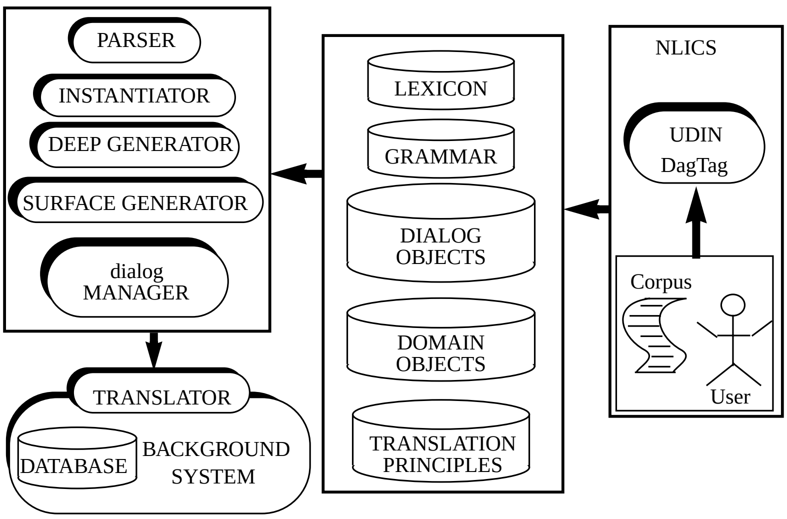


Figure 21-1. *The LINLIN architecture. The processing modules appear in the upper left part of the figure, while the customizable knowledge bases appear in the middle. The right part represents various tools needed for customization of the system.*

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 that 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. The discourse representation consists of three dynamic structures. The first one, the score-board, keeps information about salient objects and properties which are 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 is combined 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 interaction is interpreted from the information conveyed in the speech act directly; no reasoning about users' intentions or goals is utilized. Speech act information is assembled into Initiative-Response units which form a basis for interpreting the segment structure. A simple context free grammar can model the interaction and the rules are selected based on information about properties of objects describing the information provided by the system. Referring expressions are handled by copying information from the previous segment to the current segment which is in turn updated with information from the background system.

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), Dahlbäck and Jönsson (1992) and Jönsson (1991, 1993a, 1993b)

The principles for dialogue management utilized in the Dialogue Manager are applied to written natural language interaction for simple service systems. We are currently investigating its applicability for multi-modal interaction. There are indications that the principles also apply to multi-modal communication for simple service systems (Jönsson, 1995; Stein & Thiel, 1993).

21.2.1.2 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 these kind of dialogues 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, 1993).

We have previously studied a number of different real or simulated background systems, to provide an empirical basis for the development of the LINLIN system described above. This work is described in a number of publications, e.g. Dahlbäck & Jönsson (1989, 1992), Dahlbäck (1991a, b), Dahlbäck, Jönsson, Ahrenberg, (1993).

This work, as well as similar studies by others, indicates that dialogues with computers in written natural language differ from dialogues between people. It is still, however, an open question as 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 an on-going project we have collected a corpus of 60 dialogues 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 continuing, but the results obtained thus far indicates that there are small or no differences between the dialogues with people and with computers. Consequently, the characteristics of so-called ‘computerese’, i.e. the sub-language used when interacting with a computer, seem to stem more from the characteristics of the communication channel and the task situation, than from the believed characteristics of the communication partner.

21.2.1.3 Empirical studies of computational models of discourse

Research on computational models of discourse can be motivated from two different standpoints (cf. Dahlbäck & Jönsson, 1992). One is to develop general models and theories of discourse for all kinds of agents and situations. The other approach is to account for a computational model of discourse for a specific application, say a natural language interface. It is not obvious that the two approaches should present similar computational theories for discourse. Instead the different motivations and approaches should be considered when presenting theories of dialogue management for natural language interfaces.

There are two general classes of theories on dialogue management in the natural language community. One is the plan-based approach. Here the linguistic structure is used to identify the intentional state in terms of the user’s goals and intentions. These are then modelled in plans describing the actions which may possibly lead to their fulfilment.

The other approach to dialogue management is to use only the information in the linguistic structure to model the dialogue expectations, i.e. utterances are interpreted based on their functional relation to the surrounding interaction. The idea is that these constraints on what can be uttered, allows us to write a dialogue grammar to manage the dialogue.

The plan-based approach is not only a model for dialogue in natural language interfaces but also aims to account for general discourse. The dialogue grammar approach, however, is more limited (though there are researchers that claim that this method could also be used as a general model of discourse, both within computational approaches (e.g. Reichman, 1985), and in other areas of discourse analysis (e.g. Stubbs 1983)).

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 a segment which is both general and precise enough for computer applications. In these circumstances we found 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 introduce 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, 1991a, 1992) 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, Jönsson 1993a). 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, 1993a).

One problem with comparing the two approaches to dialogue management is that they have been developed using different empirical bases. To overcome this, we are currently engaged in a project whose aim is to empirically compare the two approaches, by analysing a set of dialogues using both models. We will collect a corpus of dialogues from human-computer interaction, both written and spoken, and analyze the dialogues using both a coding scheme for our dialogue grammar model, as well as with a scheme for a plan or intention based model, similar to the one used by Grosz and Hirschberg in their empirical work on discourse structure (Hirschberg and Grosz, 1992, Grosz and Hirschberg, 1992). The dialogues will come both from our own corpora, as well as from other researchers in Sweden and abroad. We are interested both in issues such as coding reliability and applicability for the different approaches, as well as the usefulness of the assigned structures for anaphora resolution and answer generation. This work is still in progress, and will continue until the summer of 1996. Parts of the work will be presented at the 1995 AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation (Ahrenberg, Dahlbäck and Jönsson 1995).

21.2.2 Intelligent Help Systems (The PUSH Project)

The PUSH project (Plan and User Sensitive Help) is a joint effort between Ellemtel, SICS (Swedish Institute for Computer Science), The Department of Computer and Information Science at Stockholm University, and our group. The aim of the project is to develop an advanced help system for a software development method used by Ellemtel. We are taking part in the empirical investigations for the project. The initial studies of the users needs' and requirements' are finished, and we are now planning an empirical comparison of different help methods using Wizard of Oz studies and other techniques.

In another part of this project we are studying the relationship between users' abilities to navigate in hypertext systems and their cognitive abilities, especially spatial abilities.

21.2.3 Grammar Formalisms and Linguistic Analysis

Most NLP-applications require that a fairly large subset of the constructions of a natural language can be recognized and interpreted. We have used constraint-based (or unification-based) grammar formalisms for the representation of grammatical knowledge in most of our projects. In addition, we have studied and tested a large number of constraint-based formalisms from the computational as well as the linguistic perspectives. In this section we report on two projects dealing with grammar writing and grammar formalisms. One is focused on making constraint-based formalisms more general and flexible, the other is concerned with their application to Swedish grammar.

21.2.3.1 A flexible unification formalism (FLUF)

Constraint-based grammar formalisms in their purest form use term or graph unification as the sole information-combining operation e.g. DCG or PATR-II. However, on descriptive grounds several extensions have been proposed, for example set values, negation, finite disjunction, infinite disjunctions expressed by regular paths (functional uncertainty), and generalization of feature structures. These extensions are intended to preserve the declarativity of the formalism and the monotonicity of unification. Other proposed extensions, such as defaults or multiple inheritance make the formalisms non-monotonic. Recent suggestions to control processing by means of annotations to rules make them non-declarative.

The increase of expressivity of constraint-based formalisms has, generally speaking, made them less tractable and less efficient for practical purposes. Since so many variants have been proposed, it has also become difficult to compare achievements and re-use resources that have been developed at different sites.

Lena Strömbäck has explored various possible and proposed extensions to constraint-based formalisms. In her licentiate thesis, Strömbäck (1992a), she surveys some of the recent systems and compares them in different respects, the most important being expressiveness, tractability, flexibility and predictability. While the first two are commonplace for representation languages, the other two are equally important. 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.

In more recent work, a formalism, FLUF, has been developed that is flexible in the desired sense, i.e. the user can define the constructs that he wants for a particular purpose, including feature structures, sets, lists and trees (Strömbäck, 1994a, 1994b, 1994c). The definitions of any construct can also be modified according to the user's needs. Multiple inheritance and non-monotonic constraints such as the constraints on coherence and completeness used in Lexical-Functional Grammars can also be defined. A prototype implementation of the system has been made, using narrowing to handle user-defined constructions (Hanus, 1994). While this makes the system inefficient for practical purposes, it is possible to improve the behaviour of the system by providing a library of special-purpose unifiers for those constructs that can be expected to be used often.

Thesis work

The work on flexibility in constraint-based formalisms is the subject of Lena Strömbäck's thesis work. This work is performed within the project General Unification led by Lars Ahrenberg and sponsored by TFR.

21.2.3.2 Topological frames and constraint-based grammars

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 topological frame, or *schema*, as he called it, 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 for every subtype of the category to which they apply. Thus, the clause schema is true for 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, 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 topological frames. The frames 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 topological frame 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. Moreover, schemas and LP-rules can be combined in a grammatical description if, say, the scope of LP-rules is restricted to fields.

In recent work we are looking at how topological frames can be integrated in a sign-based framework such as HPSG (Ahrenberg, 1994). It is natural to view the topological frame as providing a description level of its own, as is done in some recent versions of HPSG (e.g. Kathol, 1993). Topological frames are simple structures, and constraints placed on them from different parts of a phrase can simply be conjoined. This is in contrast to other proposals which rests on complex joins of word order domains when two or more signs combine to form a larger phrase.

21.2.4 Incremental Natural-Language Processing

This project concerns fully incremental language processing, that is, efficient handling of arbitrary changes to some input, such as a text. The long-term objective of the project is to generalize the notion of full incrementality to a bidirectional system, in which parsing and generation are instances of a uniform deduction process, and to demonstrate the utility of this in connection with applications such as language-sensitive text editing.

Fully incremental language processing is needed in order to efficiently handle a sequence of arbitrary cumulative and non-cumulative changes, as they occur in display editors and other text-production systems, and possibly in spontaneous speech. This notion contrasts with the standard conception of incrementality in natural-language processing, which is concerned with piecemeal left-to-right addition of linguistic material. Left-to-right incremental processing can be handled by monotonic algorithms, such as chart parsing and its deductive analogue. In contrast, fully incremental parsing is non-monotonic: if, for instance, a new word is inserted in the midst of the text, then some lemmas become invalid.

Linguistic support for text production potentially requires both parsing and generation to be fully incremental. For example, grammar checking immediately and in real time within a text-editing environment requires incremental parsing as a text is changed by a user. Linguistic structure editing — say, changing a complex noun phrase from singular to plural in a single operation — requires incremental generation from a changed linguistic structure to a modified surface expression.

To achieve consistency between (interleaved) incremental parsing and generation, a bidirectional system is needed. The reason for this is that bidirectionality (in its strong form) allows not only the same grammar and the same processor but also the same linguistic structures to be used in both directions, thereby guaranteeing that the results of the two processes are consistent. (This does not preclude approaches to handling of ill-formed input that relax certain constraints in order to “fit” a parse to a given input.)

In his thesis work, Wirén generalized the notion of parsing as deduction to non-cumulative text production, in which changes can be made to already parsed material. During 1992-94, Wirén continued the work at the Department of Computational Linguistics in Saarbrücken, supported by the German Science Foundation (DFG) through the Sonderforschungsbereich 314, project N3 (BiLD). He was then employed at IDA 1 July-30 November 1994, being funded by the Swedish National Board for

Industrial and Technical Development (NUTEK) and the Board of Research in the Humanistic and Social Sciences (HSFR).

The work has been carried out at three levels during this entire period:

1. Formal basis

To provide a theoretical basis for work on incremental algorithms, a formal measure of "the set of things actually changed" has been established. The measure is defined on the basis of a subtraction operation on the results obtained by *batch-mode* processing of the inputs before and after a change, respectively. Since the resulting measure captures what a sound and complete incremental algorithm *must* do, it can be used as a basis for proving the correctness of such an algorithm. Furthermore, as Wirén (1994) has shown, it can be used as a basis for a measure of incremental complexity. This means that it becomes possible to determine to what extent an incremental algorithm is bounded by the size of a change, rather than the size of the entire current input.

The definitions have so far been given relative to a chart-parsing framework. The definition of minimal change in (typed) unification grammars is an open problem, and no algorithms for fully incremental feature-constraint solving have been put forward. However, Teege (1994) has defined a subtraction operation for description logics. Presumably, this work could be used as a starting-point for a definition of difference in the context of feature structures. Such a measure could in principle also be used also for generation.

2. Algorithms

A first incremental chart-based algorithm was specified by Wirén in his thesis work. A refined algorithm which is polynomially bounded by the size of a change was then given by Wirén (1994). Intuitively, a bounded incremental algorithm only processes the region where the input or output changes during an update cycle.

Although the goal of generalizing full incrementality to a bidirectional system has yet to be fulfilled, the general directions of this work have been set out. The intent is to make use of Neumann's (1994) bidirectional model, where the same grammar and processor are used for parsing and generation. Essentially, Neumann adapts Earley deduction to generation, thereby obtaining a uniform, tabular (chart-based) framework which works in both directions. Adopting Wirén's previous approach, this makes it possible to achieve incrementality in both cases.

3. Implementations

Previous work has involved several experimental implementations, the most comprehensive of which is called *Lips* (Dahlén 1991). In particular, this system adapts the original algorithms for fully incremental chart parsing to on-line processing under a simple text editor, thus providing a system which parses a text simultaneously as it is entered and edited. Implementation of the bounded incremental algorithm is underway.

21.2.5 Robust Natural Language Analysis

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 of modularity and customization. In other projects we are addressing other aspects of natural language understanding from the same basic view on the process. A problem for natural language interfaces is that the user is likely to input strings that they 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 study of the user inputs to the simulated systems *Cars* and *Travel* from our dialogue corpus, all instances of input that display some form of deviance from what you would expect your grammar to account for, were noted and classified. The results are shown in tables 1 and 2.

Table 1. Some problematic types of constructions for a parser

Type	Example
Lexical transgressions	
Corrupted words	Visa alla blar som kostar mindre än ...
Segmentation error	Visa alla bilarsom kostar mindre än ...
Grammatical transgressions	
Missing words	Visa alla bilar kostar mindre än ...
Extra words	Visa visa alla bilar som kostar mindre än ...
Word substitutions	Visa alla bilar som kostar mindre en ...
Agreement errors	Visa fem bilarna som kostar mindre än ...
Elliptical constructions	
Telescoping	Pris dessa bilar?
Contextual ellipsis	Toppfart och acceleration

We are currently developing a system for robust understanding that should be able to cope with these deviances. The system combines an incremental chart-parser for a PATR-like formalism (Wirén, 1992) with Hidden Markov Modeling of word forms and word sequences. This hybrid methodology we expect will take care of most of the lexical and syntactic deviances. For the remainder, parse fitting may prove helpful. As for the elliptical inputs that require contextual information for their interpretation, we use a simple discourse model, as in previous research (Jönsson, 1993a). The project is sponsored by HSFR and NUTEK through their Language Technology Program.

Table 2. Frequency of problematic types of constructions (n = 520)

Type	Number of instances
Lexical transgressions	**143 (28%)**
Corrupted words	92
Segmentation error	51
Grammatical transgressions	**40 (8%)**
Missing words	7
Extra words	0
Word substitutions	17
Agreement errors	15
Other	1
Elliptical constructions	**337 (64%)**
Telescoping	199
Contextual ellipsis	138
Total	**520**

Thesis work

This work is part of Peter Ingels thesis work and a follow up to his Master's Thesis which applied solely chart- and grammar-based techniques to the problem of robust analysis (Ingels, 1992; 1994).

21.2.6 Generation and translation of documentation

At NLPLAB there are two ongoing projects within the field of translation and generation of technical documentation. One is focused on translation and translation aids and the other on multilingual generation. The translation aids project carries the title "Linguistic engineering for the generation and translation of documentation" and the participants here are Magnus Merkel, Lars Ahrenberg, Bernt Nilsson and Stefan Svenberg. The second project, "Conceptual Text Representation for Automatic Generation and Machine Translation", is run by Lars Ahrenberg and Stefan Svenberg.

21.2.6.1 Translation aids and text characteristics

A characteristic feature of many types of manuals, in particular computer manuals, is the high frequency of recurrent (or repeated) translation units. This is a fact that can be exploited in translation with quite simple tools, e.g. a computerised phrase-book where recurrent units are stored with their translations. In a pre-study (Merkel 1992) we found cases where up to 43 per cent of the total text in a handbook was made up of recurring identical sentences and (translatable) phrases. The same study showed that 20 per cent of the text in another handbook was made up of sentences that were already translated in the previous version. Recurrent phrases found in both documents comprised 15 per cent of the text. Taken together this meant that 31 per cent of the handbook could have been automatically translated with the aid of a translation memory of sentences and phrases of the previous version of that same handbook. When the recurrent phrases and sentences within the handbook were also taken into consideration, the analysis yielded that 52 per cent of the text was repetitious, either internally or externally.

One basic hypothesis of the project is that an interactive memory-based translation system, i.e. a system providing a terminology database, a database of previously translated units (sentences, phrases and perhaps paragraphs) coupled with a set of tools that derive internal and external recurrence profiles for a given text material would give substantial improvements to the translation process, particularly in speed and certain aspects of quality, such as terminological and stylistic consistency. The aim of the project is to determine the advantages and drawbacks of memory-based systems and to find good designs for them. We also study and develop analysis tools that can predict the behaviour of a memory-based system on a given body of text and other support tools for the translation process. Finally, we hope that the project can suggest ways of integrating memory-based systems and rule-based systems. Some of the initial findings are reported in Merkel (1993a, 1993b), Merkel et. al. (1994).

An important task of the project is to study the effects on the target text of the translation method used and compare translations made by means of memory-based systems with manual translations. This requires the development of methods and tools for evaluation of translation, which is not an easy task. To test design alternatives we need to consider the format and content of the databases as well as the matching algorithms used in database search. In this connection we investigate different methods and tools for data acquisition and the possibility of making database search sensitive to language-independent information encoded in the source document, e.g. semantic or functional properties of a paragraph encoded descriptors in SGML (Standard Generalized Mark-up Language). The analysis tools primarily support the identification of translation units of a text body, where strings as well as linguistically more interesting units such as lemmas, terms, phrases, patterns and constructions are considered. The analysis tools can be used diagnostically in several ways. For example, a text profile can be generated showing what parts of it are covered to what extent by recurrent items at various levels of abstraction, and the recurrent items can be checked for counterparts in an existing translation memory. Both kinds of information are relevant for deciding what efforts and resources are needed for the translation of the given text body.

While the purpose of this project is to develop tools for automatic analysis, it should be stressed that the tools usually require human intervention and guidance.

The translation support tools we are investigating fall into four different categories (see Ahrenberg & Merkel (1994) for a more detailed discussion):

1.*Diagnostic tools* that characterize texts and text-types in terms of parameters that have a direct bearing on the performance and usability of various computer-supported methods of translation; by applying such tools to a representative sample of texts for a given text-type, a set of text profiles is obtained that reveal characteristics of the text type and that can support decisions as to what kind of computer support should be used in the translation process;

2. *Alignment tools* that establish correspondences between source and target texts, on various levels such as chapters, divisions, headings, paragraphs, sentences, phrases and words;

3. *Data acquisition tools* that retrieve data from bilingual corpora, which can be exploited in the actual translation process; and

4. *Evaluation tools* that are used in the evaluation of translations and checking of properties such as consistency of terminology, variation in phraseology and conformity with a given style-guide.

21.2.6.2 An English–Swedish Translation Corpus

A prerequisite for the exploration and use of bi-texts is that you have one at your disposal. In a Swedish context, English and Swedish are by far the two most common source and target languages to consider. As there were no such English-Swedish translation corpus available at the start of the project, an important part of it is to create one and align the texts at least at the paragraph level.

The text corpus consists of five different sets of translations from English into Swedish. There are two different sets of computer program manuals from two companies (1 and 2) where the major difference lies in the method of translation. Text 1 has been translated manually and text 2 has been translated with the aid of a memory-based translation tool. Text 3 will comprise of technical documentation but from another domain than computers and computer programs. Text 4 is a selection of legal documents from the EC which is not technical documentation in the strict sense, but as the production of these translations is done in an industrialised manner, they are interesting as empirical material representative for another text genre. Finally there are two novels in the corpus, which will function as reference material to the rest of the material. The choice of the corpus texts has been made with regard both to the availability of the texts in electronic format and to achieve a balance of different text types and translation methods. Depending on what will actually be marked up and aligned at the end of the project, the size of the corpus will be between four and six million words (i.e., two to three million words per language).

21.2.6.3 Characteristics of texts

The tools will by themselves form a valuable result from the project, but of equal importance is the information that can be extracted from the translation corpus. Questions that we hope to find tentative answers to are: What is the effect of using a memory-based tool for translation as compared with a purely manual translation? To what extent are translations consistent when it comes to terminology and phraseology? What regular correspondences can be identified for a given text type between interesting linguistic units in English and Swedish? And, how large a part of a text can be expected to be covered by those correspondences?

21.2.6.4 Conceptual 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 text. It is then used for producing the text in some target language.

In the project “Conceptual Text Representation for Automatic Generation and Machine Translation”, Stefan Svenberg has been working on a system for multi-lingual generation from an inter-lingual text representation (Ahrenberg & Svenberg 1992, Svenberg 1994). The project works with descriptive texts found in technical service and instruction manuals. The primary goals of the project are to develop a suitable representation of such texts, and to find algorithms that can generate the text in Swedish and English.

Some sample manuals have been studied which revealed a clear pattern governing how facts about the technical domain are expressed in the texts in terms of words and syntax. A knowledge base has been employed to represent the facts as structured concepts. Multi-lingual generation is based on a principle of compositionality; the linguistic expression for a concept is a function of the linguistic expressions of its parts. The principle is general enough to hold for any language. A grammar is employed whose purpose is to relate structured concepts to functions, and a lexicon gives expressions for the primitive unstructured parts. Generation means setting up and applying the functions which will result in an expression for a concept. The same process applies for all languages, only the expressions and the results of the functions differ which makes it possible to generate into all target languages simultaneously. The expressions are standard phrase structure trees for sentences that can trivially be converted into a string of words, possibly including tags for a mark-up language or for a type setter.

To generate a text means to generate a number of sentences. To guarantee a coherent text, underlying goals and intentions must be given attention. These can be accounted for in terms of the reader's assumed information needs at any given point in the text. Such needs are the same regardless of language. The information needs can in turn be modelled as queries to the knowledge base. Thus, a query paired with a domain fact from the knowledge base gives an inter-lingual representation for a sentence, and a string of them gives an inter-lingual representation for the entire text. Some rhetorical relations are included as well.

An experimental system has been developed that features a frame system as a knowledge base, and a Prolog-like theorem prover implemented in Common Lisp. The theorem prover tries to prove by example that the input concept for generation actually has corresponding phrase structure trees in the target languages.

References:

Ahrenberg, L. (1990). A Grammar Combining Field Structure and Phrase Structure. *Proc. of the 13th Int. Conf. on Computational Linguistics*, Helsinki, Aug. 20–25, 1990, Vol. 2, pp. 1–6.

Ahrenberg L. (1992). On the integration and scope of segment-based theories of discourse. Paper presented at the *Third Nordic Conf. on Text Comprehension in Man and Machine*, Linköping, Apr. 21-23, 1992.

Ahrenberg, L. (1993). Topological frames in sign-based grammars. *NODALIDA, '93 - Proc. of ‘9:e Nordiska Datalingvistikdagarna’*, pp. 9-17, Stockholm, Jun. 3-5, 1993.

Ahrenberg, L., Dahlbäck, N., Jönsson, A. (1994). Coding Schemes for Studies of Natural Language Dialogue. To appear in *Working notes from AAAI Spring Symp.*, Stanford, Mar. 27-29, 1995.

Ahrenberg, L., Jönsson, A., Dahlbäck, N. (1990). Discourse Representation and Discourse Management for a Natural Language dialog System. In *Proc. of the Second Nordic Conf. on Text Comprehension in Man and Machine,* Stockholm, Apr., 1990.

Ahrenberg, L., Jönsson, A., Thurée, Å. (1993). Customizing interaction for natural language interfaces workshop on pragmatics in dialogue. In *Proc. of the XIV:th Scand. Conf. of Linguistics and the VIII:th Conf. of Nordic and General Linguistics*, Göteborg, Sweden.

Ahrenberg, L., Merkel, M. (1994). Translation Corpora and Translation Support Tools. *Languages in Contrast. Text-based cross-linguistic studies.* Lund University Press.

Ahrenberg, L., Svenberg, S. (1992). Conceptual text representations for multi-lingual generation and translation. *Proc. of the Scand. Conf. on Computational Linguistics*, Bergen, pp. 7-18, Nov. 28-30, 1991.

Dahlbäck, N. (1991a). *Representations of Discourse: Cognitive and Computational Aspects.* Doctoral diss., Linköping Studies in Arts and Science No. 71 and Linköping Studies in Science and Technology No. 264.

Dahlbäck, N. (1991b). Empirical Analysis of a Discourse Model for Natural Language Interfaces, *Proc. of the Thirteenth Annual Meeting of the Cognitive Science Society, (CogSci'91)*, pp. 1-6, Chicago, Illinois.

Dahlbäck, N. (1992). Pronoun Usage in NLI-dialogues. A Wizard of Oz Study. *Proc. of the Third Nordic Conf. on Text Comprehension in Man and Machine*, pp. 27-42, Linköping, Sweden.

Dahlbäck, N. Jönsson, A. (1989). Empirical Studies of Discourse Representations for Natural Language Interfaces. *Proc. of the 4th Conf. of the European Chapter of the Association for Computational Linguistics,* Manchester, pp. 291–298, Apr. 10–12.

Dahlbäck, N. Jönsson, A. (1992).An Empirically Based Computationally Tractable Dialogue Model. *Proc. of the Fourteenth Annual Meeting of the Cognitive Science Society, (CogSci'92)*, pp 785-790, Bloomington, Indiana.

Dahlbäck, N, Jönsson, A., Ahrenberg, L. (1992). Wizard of Oz studies - Why and How. *Knowledge-Based Systems*, 6(4), 258-266. Revised version paper published also in *Proc. of the Int. Workshop on Intelligent Systems*, Orlando, Florida, Jan. 4-8, 1993.

Dahlén, U. (1991). '*LIPS. Ett system för interaktiv och inkrementell parsning av naturligt språk*'. Master's Thesis, LiTH-IDA-Ex-9136, Dept. of Computer and Information Science, Linköping Univ., Linköping, Sweden.

Grosz, B. Hirshberg, J. (1992) Intonational Features of Local and Global Discourse Structure. *Proc. The DARPA Speech and Language Workshop,* Morgan Kaufmann.

Hanus, M. (1990). The Integration of Functions into Logic Programming. From Theory to Practice. Technical Report, Max-Planck-Institut für Informatik, Saarbrücken, Germany.

Hirshberg, J. Grosz, B. (1992) Some Intonational Characteristics of Discourse Structure. *Proc. Int. Conf. on Spoken Language Processing (ICSLP'92.)*

Hirsch, R. (1994a). Achieving reference: 'A rope of no use'. In Geiger, R. A., ed., *Reference in a Multidisciplinary Perspective Philosophical Object, Cognitive Subject, Intersubjective Process*. Hildesheim: Georg Olms Verlag. To appear.

Hirsch, R. (1994b). Gesture universals: Material and formal constraints. In Sebeok, T. A. and Umiker-Sebeok, J., eds., *Advances in Visual Semiotics*. Univ. of Indiana Press, pp. 475-490, Bloomington, Indiana.

Hirsch, R. (1994c). Interaktive Argumentation: Theorie und Daten 'Auf der Suche nach einer Theorie interaktiver Argumentation'. In Wohlrapp, H., ed., *Wege der Argumentationsforschung*. Stuttgart/ Bad Cannstatt: frommann-holzboog Verlag, pp. 109-119.

Ingels, P. (1992). *Error detection and error correction with chart parsing and relaxation in natural language processing*. Master's Thesis. Dept. of Computer and Information Science, Linköping Univ., LiTH-IDA-Ex-9269.

Ingels, P. (1994) Robust parsing with charts and relaxation. *NODALIDA '93 - Proc. of '9:e Nordiska Datalingvistikdagarna*', pp. 123-132, Stockholm, Sweden, Jun. 3-5, 1993.

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

Jönsson, A. (1993a). *Dialogue Management for Natural Language Interfaces – An Empirical Approach.* Ph.D. Thesis No. 312, Dept. of Computer and Information Science, Linköping Univ.

Jönsson, A. (1993b). A Method for Development of Dialogue Managers for Natural Language Interface. In *Proc. of the Eleventh National Conf. of Artificial Intelligence (AAAI'93)*, Washington DC, USA, Jul. 13-15, 1993.

Jönsson, A. (1995). A Dialogue Manager for Natural Language Interfaces. In *Proc. of the Pacific Association for Computational Linguistics, Second Conf.*, The Univ. of Queensland, Brisbane, Australia.

Kathol, A. (1993). Linearization and Coordination in German. In Papers in *Syntax*, *The Ohio State University Working Papers in Linguistics No. 42*, pp. 117-152, Sept., 1993.

Merkel, M. (1992). *Recurrent Patterns in Technical Documentation*. Research Report LiTH-IDA-R-92-31, Dept. of Computer and Information Science, Linköping Univ.

Merkel, M. (1993a). When and why should translations be reused? In *Papers from the XIII VAAKKI Symp. 1993*, Vaasa, Finland.

Merkel, M. (1993b). Semiotics at Work: Technical Translation and Communication in a Multilingual Corporate Environment. In the *Proc. of NODALIDA* ('Nordiska Datalingvistikdagarna'), Stockholm Univ., Co-author: Arne Larsson, Nokia Telecommunications.

Merkel, M. Nilsson, B. Ahrenberg, L. (1994). A Phrase-Retrieval System Based on Recurrence. In *Proc. from the Second Annual Workshop on Very Large Corpora*, Kyoto, Japan. Aug., 1994.

Neumann, G. (1994). *A Uniform COmputational Model for Natural Language Parsing and Generation*. Ph.D. thesis, Univ. of the Saarland, Saarbrücken, Germany.

Reichman, R. (1985). *Getting Computers to Talk Like You and Me*, Cambridge, Mass.; A Bradford Book, The MIT Press.

Stein, A. Thiel, U. (1993) A Conversational Model of Multimodal Interaction in Information Systems. In *Proc. of the Eleventh National Conf. of Artificial Intelligence (AAAI'93).*

Strömbäck, L. (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, L. (1992b). Unifying Disjunctive Feature Structures. *Proc. of COLING-92*, Nantes, pp. 1167-1171, Aug. 23-28, 1992.

Strömbäck, L. (1994a) Achieving Flexibility in Unification Formalisms. *COLING 94 Proc.*, Vol. II, pp. 842-846, Kyoto, Japan. Aug., 1994.

Strömbäck, L. (1994b). *FLUF: A Flexible Unification Formalism - the Idea*. Technical Report LiTH-IDA-R-94-12. A short version accepted to the *15th Int. Conf. on Computational Linguistics*, Kyoto, Japan, Aug. 5-9, 1994.

Strömbäck, L. (1994c). *FLUF: A Flexible Unification Formalism - Syntax and Semantics*. Technical Report LiTH-IDA-R-94-13.

Stubbs, M. (1983). *Discourse Analysis*. Oxford: Blackwell.

Svenberg, S. (1994). Representing Conceptual and Linguistic Knowledge for Multi-Lingual Generation in a Technical Domain. In *Proc. of the 7th Int. Generation Workshop*, Kennebunkport, Maine, USA, pp. 245-248, June 21-24 1994.

Teege, G. (1994). Making the Difference: A Subtraction Operation for Description Logics. *Proc. KR*, Bonn, Germany, 1994.

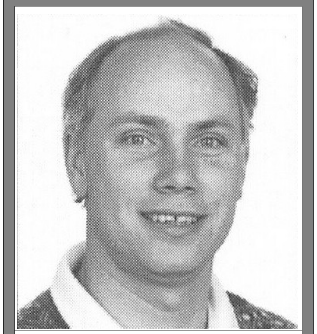
van Noord, G. (1989). BUG: A Directed Bottom-Up Generator for Unification-Based Formalisms. Working Papers in Natural Language Processing 4. Katholieke Universiteit Leuwen.

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

Wirén, M. (1994). Minimal change and bounded incremental parsing. In *Proc. of the 15th Int. Conf. on Computational Linguistics*, pp. 461–467, Kyoto, Japan. Aug., 1994.

Wirén, M. Rönnquist, R. (1993). Fully Incremental Parsing. In *Proc. of the Third International Workshop on Parsing Technologies.* Tilburg, The Netherlands and Durbuy, Belgium, Aug. 10-13, 1993.

PELAB


Peter Fritzson
director

22 The Laboratory for Programming Environments

Scientific and engineering computing
Debugging and testing tools
Compiler generation, incremental compilation, parallel hardware
Object Oriented Modelling
Distributed systems
Industrial evaluation

Some of the members of the Programming Environments Laboratory.
From the back: Henrik Nilsson, Johan Herber, Bengt Lennartsson, Mikael Pettersson, Johan Ringström, Johan Fagerström, Rickard Westman, Niclas Andersson, Tommy Persson, Gunilla Blom-Lingenhult, Lars Viklund, Franz, Lars Willför, Mariam Kamkar, Peter Fritzson, Patrik Hägglund, Eva Toller, Kent Engström.

22.1 Introduction

PELAB, the Programming Environment Laboratory of the Department of Computer and Information Science at Linköping University, is concerned with research in software engineering, i.e. tools and methods for the specification, development and maintenance of computer programs. Some examples are: programming languages, 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 of programming environment research is rather pragmatic. We are primarily interested in developing and investigating new methods and tools that have potential for 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 two years:

- Programming environment support for high-level scientific modeling in equations and compilation to efficient code
- Semi-automatic debugging support through algorithmic debugging and strictification for lazy functional languages
- Semi-automatic debugging and testing support through inter-procedural and inter-process data flow analysis – dynamic slicing
- Compiler generation from denotational and natural semantics specifications
- Parallel programming and compiler generation for data-parallel hardware
- Scheduling and generation of parallel code for solving ordinary differential equations
- Real-time functional language implementation and applications for massively parallel computers
- Design and implementation of a multiparadigm parallel language combining object-oriented and rule-based programming.

New activities have recently been started in the areas of software testing and evaluative research concerning large industrial software systems.

22.2 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.

The work in PELAB is supported by NUTEK, The Swedish Board for Industrial and Technical Development, by TFR, and by the European Commission.

Laboratory leadership and administrative staff:

Peter Fritzson, Ph.D., professor.
Gunilla Lingenhult, secretary.

Laboratory members having a Ph.D. degree:

Bengt Lennartsson, Ph. D., associate professor.
Johan Fagerström, Ph. D., associate professor.
Mariam Kamkar, Ph. D., assistant professor.

Graduate students:

Henrik Nilsson, Tech. Lic
Mikael Pettersson, Ph. Lic.
Johan Ringström, Tech. Lic.
Lars Viklund, Tech. Lic.
Niclas Andersson, M.S.E
Vadim Engelson, M. Sc.
Johan Herber, M.S.E.
Patrik Nordling, M. Sc.
Eva Toller, B.A. (also affiliated with FOA)
Patrik Hägglund, B. Sc.
Rickard Westman, M.S.E.

External graduate students:

Ulf Cederling, Växjö
Jukka Mäki-Turja, Västerås

Guest Researchers:

Ernst Heinz, University of Karlsruhe, Germany, August-September 1994
Dr. Olav Hansen, Technical University of Munich, Germany, from March 1995

Full-time teachers

Rober Bilos, Tech. Lic.
Lars Strömberg, Tech. Lic.

Master students

Kent Engström
Tommy Hoffner
Kristina Holmqvist
Patrik Krajina
Beshar Zuhdy, Erasmus student from Edinburgh, May-Oct. 1994.

Other members:

Yelena Turetskaya, guest student
Lars Willför, programmer

22.3 Degrees and papers

The following degrees were awarded to PELAB members during 1993/94:

Mariam Kamkar completed her Ph.D in April 1993.
The title of the thesis is: *Interprocedural Dynamic Slicing with Applications to Debugging and Testing*.

Johan Ringström completed his licentiate degree in May 1993.
Compiler Generation for Parallel Languages from Denotational Specifications.

Lars Viklund completed his licentiate degree in June 1994.
Contributions to a High-level Programming Environment for Scientific Computing

Henrik Nilsson completed his licentiate degree in September 1994.
A Declarative Approach to Debugging for Lazy Functional Languages

Fifteen papers were accepted for publication or presented at international conferences by PELAB members during 1993/94. In addition to these, eight journal papers were accepted or published, and six technical reports were written.

22.4 Current Research Projects

Our long range goal is to enhance and simplify the programming process, both during the *specification*, *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 supports the 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. Five of our projects use this approach. The goal of one project is to compile language specifications written in DML, Denotational Meta Language, into efficient language processors which generate quadruples. During 1992-93, this resulted in a compiler generator that can produce efficient stand-alone compilers implemented in C. A more recent project has produced a first version of a system called RML (Relational Meta Language and system), which can generate efficient implementations from Natural Semantics specifications. A third 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 fourth project aims at constructing a high-level programming environment for scientific computing (called ObjectMath), which provides support for high-level mathematical modeling in equations, and transformation to efficient code, instead of manual implementation in Fortran. A recent development is generation of parallel code from mathematical models. The goal of the fifth project is to provide a framework in which programmers can work in a variety of styles, free intermixing constructs from different paradigms. The techniques supported by this project include parallel programming, the object-oriented approach, and rule-based programming. Another project concerns parallel implementations of functional languages (Erlang) for real-time systems. A cooperation with FOA and Ericsson deals with parallel programming applied to radar imaging (CARABAS).

The debugging aspects of program maintenance were the secondary focus of the DICE and PEPSy projects. 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. The program slicing technique is also important as a debugging support technology. This is the subject of an additional project.

Additional support for the testing phase of software maintenance is the focus of a new project on software testing, in cooperation with Ericsson. A related project concerns evaluative research on tools and methods used to construct and maintain large software systems.

An overview of most of the ongoing research projects is given in the following sections. The people involved are listed at the beginning of each section, in the order supervisor(s) first, followed by graduate student(s).

International Project Cooperation

PELAB took part in the industrial Esprit project GIPE II (Generation of Incremental Programming Environments II) until the project ended in Dec. 1993. Within that project we were doing work on debugging tools and programming environments. PELAB is currently taking part in seven industrial Esprit projects. Our main Esprit project is currently PREPARE (Programming Environments for Parallel Architectures), where work is being done on compilers and environments for massively parallel computers. PELAB is also a member of the GPMIMD-II project which establishes parallel computing centers in Europe cooperating through the ZEUS (Centers of European Supercomputing) network. Through this project, the National Supercomputing Center in Linköping has joined the ZEUS cooperation. In addition to that, PELAB is taking part in five Europort projects, two in Europort I (LSDYNA3D and FLOW3D), and three in Europort II (ERLANG, BEARS and TRAFFIC). Of these, PELAB's efforts are mostly concentrated on two projects: parallel implementation and applications in Erlang, and parallel methods and applications for bearing simulation (BEARS).

22.4.1 Algorithmic Debugging

Peter Fritzson, Mariam Kamkar, Henrik Nilsson

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 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 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. Previously this technique was limited to programs without side-effects, and had 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

Earlier 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. The results are described in the Ph.D. thesis 1991 by Nahid Shahmehri, and in several papers. 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.

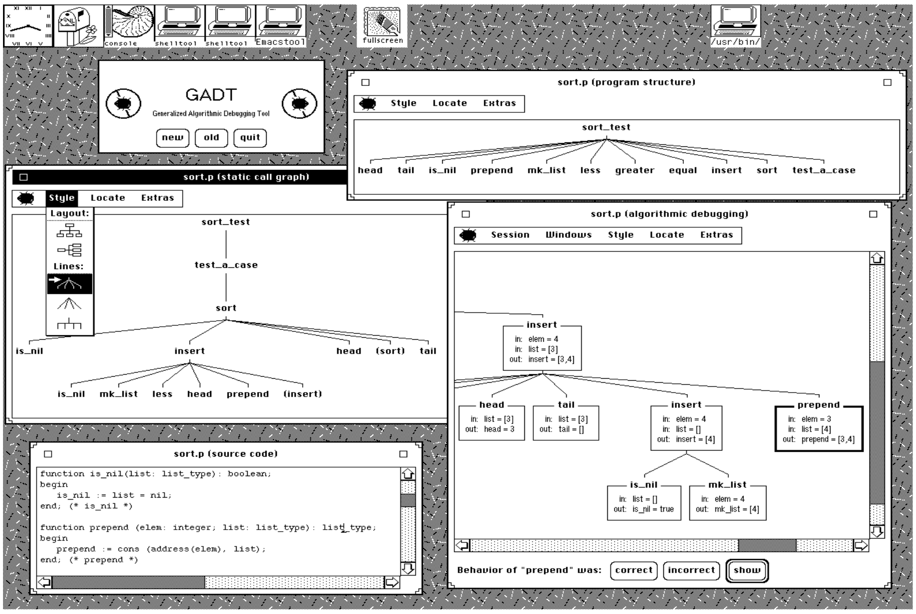


Figure 22-1. *The graphical user interface of the GADT system, including a graphical display of the execution tree. The bug localization algorithm searches the execution tree while interacting with the user when necessary*

A prototype Pascal algorithmic debugger including transformations has been implemented, integrated with the DICE system. The graphical user interface of the system, called GADT (Generalized Algorithmic Debugging Tools) is shown in Figure 22-1.

Focusing Bug Localization through Program Slicing

Program slicing, a program 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.

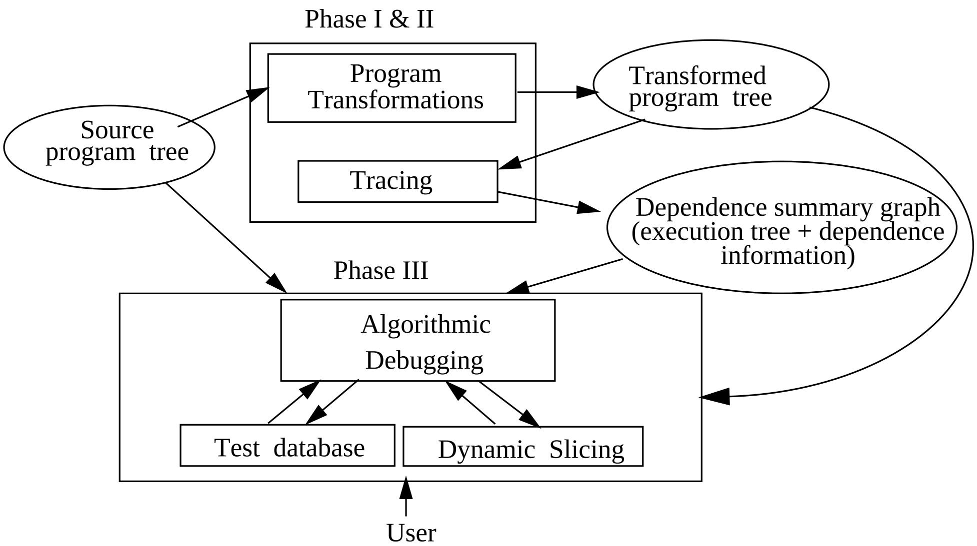


Figure 22-2. *The functional structure of the generalized algorithmic debugging system (GADT) when integrated with dynamic slicing. Arrows denote information transfer.*

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 was implemented according to these ideas 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.

Our research is currently aimed at alleviating the above problems. Piecemeal tracing, a new and much more efficient tracing scheme, has been developed. By building partial traces on demand by re-executing the program as needed, we hope to make the memory requirements far less pressing.

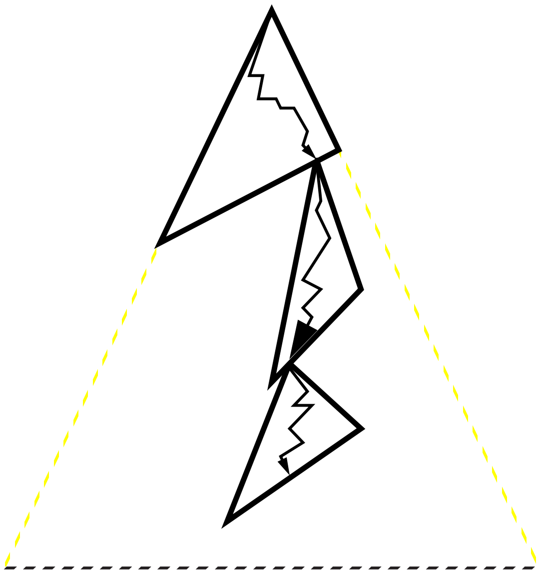


Figure 22-3. *Piecemeal tracing is a technique that trades space for time, by decoding the execution tree in small pieces at a time, re-executing parts of the program, if needed.*

We are also looking at other debugging strategies, besides algorithmic debugging, but still based on the strictified trace, and we are looking at methods for handling specific debugging problems due to advanced source level constructs, higher order functions and monad-structured programs.

This project is currently being done in cooperation with Jan Sparud and Thomas Johnsson from the Computing Science Department at Chalmers University of Technology, which is very well known in the area of lazy functional programming and implementation of lazy functional languages.

22.4.2 Interprocedural and Inter-Process Dynamic Slicing

Mariam Kamkar, Peter Fritzson, Patrik Krajina

During the past ten years several variants of an analysis technique called program slicing have been developed. Program slicing has applications in maintenance tasks such as debugging, testing, program integration, program verification, etc., and can be characterized as a type of dependence analysis. A program slice with respect to a specified variable at some program point consists of those parts of the program that may directly or indirectly affect the value of that variable at the particular program point.

A *static* program slice is computed through static data and control flow analysis and is valid for all possible executions of a program. In contrast, *dynamic* program slicing considers only a particular execution of a program. The main application of dynamic program slicing is hence in program debugging. This is due to the fact that program debugging is used to analyze a particular execution of a program, the one that denotes the existence of a bug in the program. Given an incorrect value of a variable of interest, dynamic program slicing can present a dynamic slice to the debugger for further investigation.

As an example consider the program in Figure 22-4(I) which computes sum of the integers `1` to `n`. Figure 22-4(II) shows a static slice of this program with respect to the value of variable `sum` that is computed at statement `printf("%\n", sum)`. This slice consists of all statements in the program that are needed to compute the final value of `sum` in any execution, in this example static slice is equal to the original program. Figure 22-4(III) shows a dynamic slice of the program in Figure 22-4(I) with respect to the final value of `sum` for the specific test case `n=0`.

```
main()
{
  int N, i, sum;

  iread(0, &N);
  sum = 0;
  i = 1;
  while (i <= N) {
    sum += i;
    i++;
  }
  printf("%i\n", sum);
}
```

(I)

```
main()
{
  int N, i, sum;

  iread(0, &N);
  sum = 0;
  i = 1;
  while (i <= N) {
    sum += i;
    i++;
  }
  printf("%i\n", sum);
}
```

(II)

```
main()
{
  int N, i, sum;


  sum = 0;







  printf("%i\n", sum);
}
```

(III)

Figure 22-4*: (I) Example program. (II) Static slice of the program with respect to the final value of variable* `sum`*. (III) Dynamic slice of the program with respect to the final value of variable* `sum` *for input* `n=0`*.*

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 dependencies between statements in the procedure. This graph is subsequently collapsed, and the dependencies are transferred to an interprocedural *summary graph*, which is substantially more compact. The method is described in the Ph.D. thesis by Mariam Kamkar, and has been integrated with the previously mentioned generalized algorithmic debugging tool (GADT) to improve bug localization. An overview and practical evaluation of some static and dynamic slicing systems was recently done by Tommy Hoffner as his master thesis work.

Inter-process Dynamic Slicing

The concept of dynamic slicing has been introduced originally for sequential programs. Distributed programs introduce several problems which do not exist in sequential programs (e.g., non-reproducible behaviors, non-deterministic selection of communication events, etc.). Several methods for dynamic slicing of distributed programs have been proposed. Previously we have developed an interprocedural dynamic slicing algorithm, which works at the procedure abstraction level, and which can be generalized to handle communicating, message-passing parallel programs.

In this project we present a technique for dynamic slicing of distributed programs which computes accurate slices. We introduce the notion of Distributed Dynamic Dependence Graph (DDDG) which represents control, data and communication dependences in a distributed program. This graph is built at run-time and used to compute dynamic slices of the program.

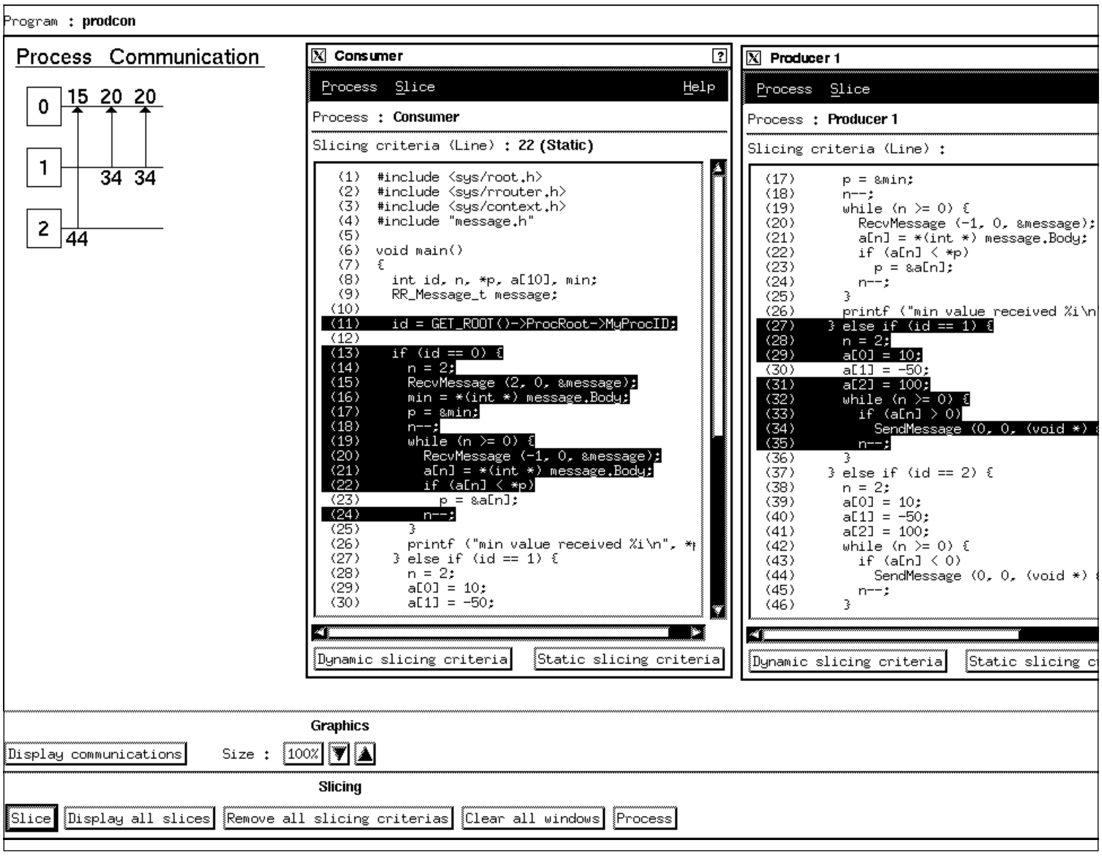


Figure 22-5. *The Parallel Distributed Dynamic Slicer (PDDS) system in action. Line 22 in the consumer (process 0) is marked as the slicing criterion. Statements which are part of the slice in the different processes are marked as shown. At the left process communication is displayed, e.g. the send at line 34 in process 1 is received at line 20 in process 0.*

A prototype dynamic slicer for parallel distributed programs is being developed as a master thesis project by Patrik Krajina. The system is shown in the Figure 22-5, slicing a small producer-consumer application. The target language is currently a subset of ANSI C, extended with message passing, and here executed on a MIMD Parsytec GC/PowerPlus 128 processor parallel computer.

22.4.3 Reliable Software by Automated Testing

Mariam Kamkar

The focus of this project is to study and develop methods for automated testing and to find the relationship between automated testing, standard software development, and software quality.

We cooperate with Ericsson Radio in Linköping in order to define/discuss industry-relevant problems and to be able to start research projects based on these problem areas. To achieve this, we have given a graduate course (August 94) whose participants were graduate students and employees from the industry. We have defined a number of master thesis projects in the area of testing. These projects are mainly based on the requirements from ERA, and will be done under joint supervision from ERA and PELAB.

One planned activity in this project is based on the interprocedural dynamic slicing technique, described in Mariam Kamkar's Ph.D. thesis, where procedures/functions are analyzed at an abstract level. A survey of software maintenance indicates that maintainers often have a functional view of the system being maintained, understanding the behavior of particular procedures better than the individual statements and data items implementing a procedure's function. Software maintainers often perform testing based upon this view of the system, choosing tests to exercise particular functions or requirements associated with procedures of the system. One goal of this project is to develop the interprocedural dynamic slicing technique further for practical testing applications such as interprocedural data flow testing, that require slicing at the procedural abstraction level.

22.4.4 A Multiparadigm Parallel Language Combining Object-Oriented and Rule-Based Programming

Mariam Kamkar, Eva Toller

Object-oriented programming has gained much attention both as a practical tool and a research subject. However, for complex applications, the object-oriented view must often be augmented with a rule/action-based view. For this purpose, object-oriented languages have been extended with rule-based features from production systems. We suggest a multiparadigm parallel language based on object-orientation, features from production systems, and joint actions. The joint action model is a rule-like language for executable specifications.

Much emphasis is also put on user-friendly parallel programming, ‘since in the future, parallel computers may be the prevailing way of gaining execution speed (but also for structuring distributed parallel applications).

Our main objectives are to investigate if the joint action concept can be adapted to programming languages, and to show if an object-oriented language can be gracefully extended with ideas from production systems and joint actions. Experiments with a prototype (written in the functional language Erlang) indicate that our design makes it easy to schedule concurrent execution, and also hints at likely application areas (experimental programming and modelling of non-deterministic events).

22.4.5 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 which run very slowly – usually a factor of 100 to 1000 times slower than with commercial compilers. Recent work by Peter Lee 1987 has demonstrated that it is indeed possible to generate realistic compilers from denotational semantics definitions. However, his system, called MESS, is monolithic and does not interface well with standard parser generators and code generators. Also the code generator in his system need to be handwritten in Prolog. In comparison to the MESS system, the DML system presented here is going 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.

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. Currently, most work is focused on a version of the system (DML-P) for data-parallel languages, see Section 22.4.7.

22.4.6 Generation of Efficient Compilers and Interpreters from Natural Semantics Specifications

Peter Fritzson, Mikael Pettersson

Compiler generation from Denotational Semantics was mentioned in the previous section. However, there are still some problems associated with Denotational Semantics, e.g. concerning modularity and when specifying concurrent languages. For these and other reasons, we have now focused on the more recent Natural Semantics formalism, and developed a first version of a system called RML (Relational Meta Language and system) for generating efficient implementations from Natural Semantics specifications. The following sections give more details of this work.

Background

Since the early eighties, a formalism known as Natural Semantics has become increasingly popular among programming language researchers. Natural Semantics is often used to specify type systems for the static semantics of programming languages, or the dynamic semantics, or both, and it has even been used to specify translations from abstract syntax to intermediate code. Lately, there has been a trend to use augmented type systems and translations, all specified in Natural Semantics, to do static analysis and code-improving transformations.

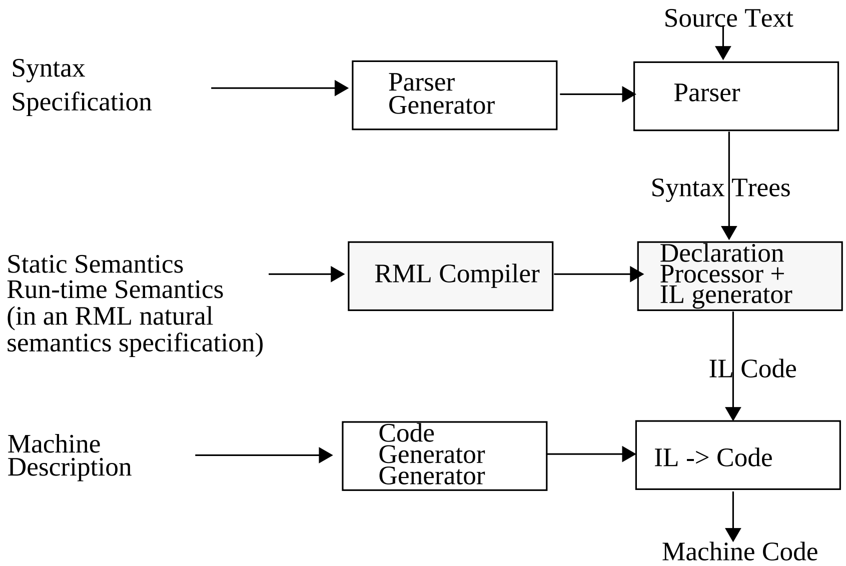


Figure 22-6. *The semantics analysis part of compilers is generated by the RML system from natural semantics specifications.*

Natural Semantics is based on Plotkin’s Structural Operational Semantics (SOS) and further developed at INRIA by Kahn. Specifications consist of data type declarations (abstract syntax, environments, run-time values, types, etc.) and sets of inference rules. The inference rules specify relations between objects, in a style akin to Gentzen’s Sequent Calculus for Natural Deduction. (Hence the name ‘Natural’ Semantics.) In a rule like

$$\frac{H_1 \vdash T_1 : R_1 \ \ldots \ H_n \vdash T_n : R_n}{H \vdash T : R} \quad \text{if } \langle\text{cond}\rangle$$

the H_i are hypotheses (typically environments containing bindings of source-level names to semantic objects), the T_i are terms (typically pieces of abstract syntax), and the R_i are results (typically types, run-time values, or augmented environments). An instance $H_j \vdash T_j : R_j$ is called a sequent. The sequents above the line are the premises, and the sequent below the line is the conclusion. The rule may be interpreted as follows: in order to prove a sequent $H \vdash T : R$, one must first prove the sequents $H_1 \vdash T_1 : R_1 \ldots H_n \vdash T_n : R_n$. The side condition, if present, must also be satisfied.

Natural Semantics offers several advantages over classical Denotational Semantics:

- All objects are finite terms, which means that the complicated domain theory of Denotational Semantics is not needed.
- More than one inference rule may be applicable at any given time, which means that some non-deterministic features are easy to model. For instance, the evaluation order of binary expressions in an imperative language can be left unspecified.
- Modern type systems involving polymorphic type inference are much easier to specify in Natural Semantics. A specification in Denotational Semantics would tend to resemble a type inference algorithm expressed as a functional program.

The Centaur programming environment, developed within the Esprit GIPE and GIPE-II projects, contains a meta-language for Natural Semantics called TYPOL. Until recently, this was the only available implementation of a language intended specifically for Natural Semantics. The default implementation uses a simple translation from TYPOL to the Centaur mu-Prolog sub-language for execution. It has also been shown that a restricted class of Natural Semantics specifications is equivalent to a certain kind of attribute grammars that can be executed by a functional evaluator.

We see several problems with the current state of affairs:

- The TYPOL implementation is very inefficient. We also feel that the Centaur system does not lend itself to the use of Natural Semantics in stand-alone applications.
- Coding Natural Semantics specifications in Prolog is not attractive, due to the lack of a decent type system in Prolog. We also believe that a compiler for a special-purpose Natural Semantics language can generate much better code than a Prolog compiler can for Natural Semantics specifications translated to Prolog.
- Some prefer to use the higher-order λProlog language. We feel that this language is too complicated, both for users and implementors alike.

Objectives

In the long run, we want to see Natural Semantics being as useful in programming language research and implementation, as are Context-Free Grammars and parser generators today.

Results

In the short run, we have defined a meta-language for Natural Semantics, and studied its implementability and practical usefulness. We have identified statically determinable properties of Natural Semantics specifications that allow (or disallow) interesting optimizations to be applied to the implementation of Natural Semantics specifications. In particular, the following results have recently been obtained:

- The Relational Meta-Language (RML) has been defined. It is strongly typed with a type system very much like that in Standard ML, has type-safe separate compilation and modules, and supports Natural Semantics-style inference rules. It has fewer non-declarative constructs than Prolog. The SML-like data types directly support structural-induction style specifications, which are central to Natural Semantics.
- The operational properties of RML were investigated and used to derive the initial implementation. A key component is the use of a Continuation-Passing Style (CPS) intermediate representation. CPS is easy to optimize, due to its declarative nature, but is also easily translated to low-level code, due to its simple operational semantics.
- Further observations lead to a refinement, whereby RML specifications are first translated to a First-Order Logic. High-level equivalences are used to rewrite this representation in order to reduce the amount of unnecessary non-determinism. This phase has proven to be essential for the practicality of the generated code.
- A compiler generating portable ANSI-C code has been implemented. The code runs unchanged on several different 32 and 64-bit architectures. Performance measurements indicate that this code runs several times faster than that generated by commercial Prolog compilers, and several orders of magnitude faster than TYPOL.
- Recent work has concentrated on the mapping of the control flow aspects of high-level languages to C. Results indicate that significant performance improvements can be made.

Performance Figures

We have a standard benchmark consisting of a NS for the dynamic semantics of a call-by-name functional language ‘Mini-Freja’. From this, we generate a compiled interpreter for the same language. Finally, we invoke the interpreter on a Mini-Freja program computing prime numbers.

Comparing the performance of TYPOL (T) and RML2C (R) for this specification on a Sun 10/41, gives the following results:

#primes	T	R	T/R
3	13s	0.0026s	5000
4	72s	0.0037s	19459
5	1130s	0.0063s	179365

The Mini-Freja specification was rewritten in Prolog to allow comparisons to be made with commercial native-code Prolog compilers: SICStus Prolog (S) and Quintus Prolog (Q). On a Sun 4/470, we have the following results:

#primes	S	Q	R	S/R	Q/R
18	5.0s	4.5s	0.45s	11.1	10.0

Further work

The pragmatic aspects of the generated code need to be improved, especially for interoperability with 'foreign' code. This is mostly a matter of design and engineering. There is much room for improvement in the compiler. Static analysis should be used to reduce the inefficiencies introduced by the language itself (e.g. unnecessary dereferencing), and those pertinent to certain classes of Natural Semantics specifications. For instance, dynamic semantics involving states are likely to benefit from an Natural Semantics analogy of the single-threadedness analysis of denotational semantics and lazy purely functional programming languages.

22.4.7 Architecture-Independent Parallel Programming and Compiler-Generation for Data-Parallel Hardware

Peter Fritzson, 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 data-parallel languages, which also includes selecting suitable data-parallel language constructs and intermediate forms, and experimenting with a small data-parallel language called Predula Noveau. 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 Noveau are 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). A special “parallel” version of the DML system called DML-P has been developed, including a set of low-level data-parallel primitives which are used as the target intermediate form.

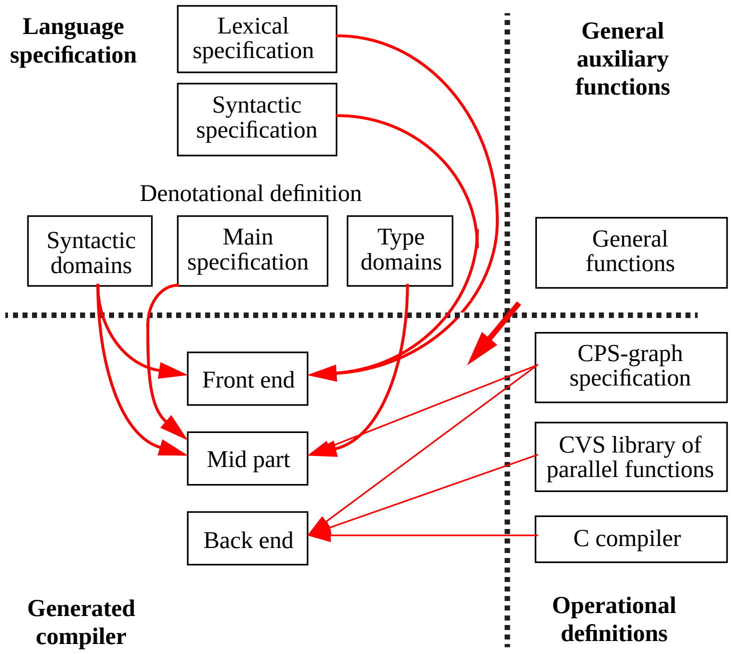


Figure 22-7. *Structure of the DML-P compiler generation system.*

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 as 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 from the VCode implementation, which implements the operations executed by the VCode interpreter. Future work will eliminate the use of the CVL library, and directly generate code for at least two SIMD target machines: the MASPAR and the special purpose VIP processor developed by the Department of Electrical Engineering and Ericsson Radar Electronics Inc.

22.4.8 Parallel Erlang

Peter Fritzson, Kent Engström

Many multiprocessing real-time applications have an increasing need for computational power, and may soon provide a larger market for massively parallel high performance computers than the rather narrow area of parallel scientific computing. Examples of such applications are telephone switching systems, image processing and recognition, real time databases, car driving assistant systems, etc. However, programming such real-time applications is quite complex and error-prone. To improve this situation a new non-lazy functional programming language has been developed by Ellemtel. This language, called Erlang, is designed especially for building large-scale industrial real-time fault tolerant applications where concurrency is explicit. It has already been used to build substantial real-time software products in the telecommunications area.

So far, the implementations of Erlang from Ellemtel use time-shared light-weight processes on single-processor workstations, addressing concurrency but not providing high performance. Therefore, a new Erlang implementation has been developed by us for high performance MIMD parallel computers, making it one of the very few real-time functional languages in industrial use that are available on such platforms. Our current parallel implementation runs on a Parsytec GC/Powerplus with 128 PowerPC processors.

Three "sequential" Erlang implementations have been developed by Ellemtel: a byte-code interpretive version, a threaded code interpretive version which is faster, and a compiled version which generates C code with some special tricks. The compiled version produces the fastest executing code, which however takes much more space than the interpretive code. All three versions use roughly the same run-time system for message-passing, I/O, error handling, etc.

Most of our work so far in developing the parallel version has been to re-implement the run-time system for the parallel platform (the first parallel version developed by Beshar Zuhdy as his master project). The message passing interface, handling of light-weight processes, etc. are all part of the run-time system, which has a reasonably well-defined external interface. Also, communication between processor nodes on the parallel machine and the frontend workstation has to be handled.

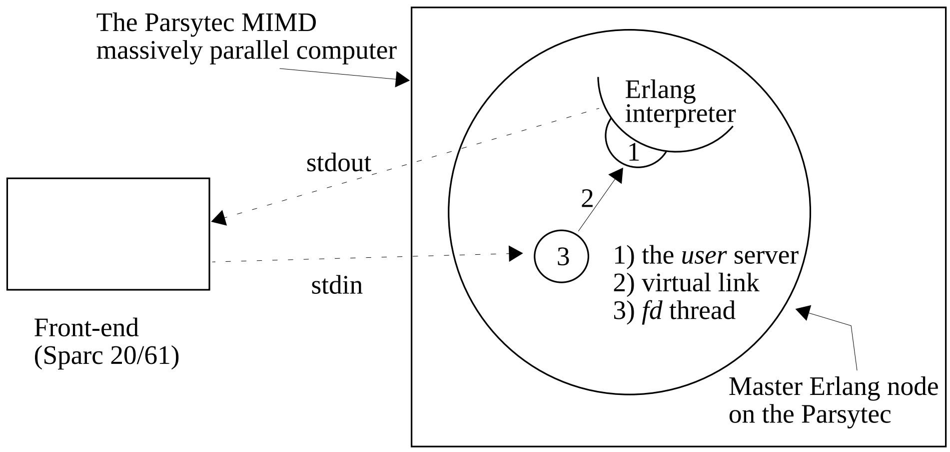


Figure 22-8 *Structure of the Parsytec Erlang implementation of standard I/O.*

We are also cooperating with industry to evaluate Parallel Erlang on real applications. The application program selected for evaluation is a simulator for an intelligent network service control system, which previously has been implemented in Erlang by Ericsson Telecommunicatie B.V. in the Netherlands. An example of a service provided by such an intelligent network service control is telephone voting. Suddenly, thousands of calls must be serviced almost at the same time. This is a typical example where the scalable processing power provided by MIMD multiprocessor systems is needed. For this application, the simulator for the intelligent network service control runs together with real switches to test the system, e.g. for situations which create a massive influx of phone calls.

Further work is planned to extend the Erlang system to handle load balancing, process migration etc., instead of forcing the application programmer to handle these issues, as is currently the case when programming in Distributed Erlang.

22.4.9 Object-Oriented Design

Johan Fagerström

In this project we study the introduction of object-oriented philosophy in an industrial setting, where the developers previous experience is limited to database design using COBOL and 4GLs. In particular, an important question is how the database design background influences analysis and design of systems. Current OO methodologies lack a way of integrating traditional database designs within a object-oriented framework. This work is done in the context of an industrial cooperation with TELUB and OKG.

During the second phase of this project, system architectures of large distributed object oriented systems will be studied, in particular the influence of large scale components on the process of analysis and design. This research will focus on integrating CORBA based systems with new large applications. The CORBA standard defines a system architecture with distributed objects that have a client-server relationship. This standard will also influence strategies for reuse.

22.4.10 Evaluative Research in Software Engineering

Bengt Lennartsson, Ulf Cederling

All too often, new methods, tools, programming languages, etc. are developed without subsequent investigation on how these concepts work in large-scale industrial use. Thus, there is a need for more *evaluative research* in software engineering, to evaluate tools etc. and systematically collect knowledge and experience on their use and on problems in developing and maintaining large software systems. This knowledge is very important as feedback to direct further research in relevant directions.

As one example, this project will collect information on experience from Ericsson's usage of the new real-time functional language Erlang for telephone switching applications. Substantial Erlang applications already exist, in the order of 250 000 lines of code.

An other activity is a case study on the relation between software architecture and software reuse at CelsiusTech Systems. This study is funded by NUTEK within the their Complex Systems program.

22.4.11 High-level Programming Environment Support for Equational Modeling in Scientific Computing

Peter Fritzson, Lars Viklund, Johan Herber, Niclas Andersson, Richard Westman, Vadim Engelson, Patrik Nordling, Patrik Hägglund
In cooperation with Dag Fritzson, SKF.

This project is developing a high-level programming environment for scientific computing (called ObjectMath) that supports mathematical modeling and semi-automatic transformation of mathematical models to efficient code, instead of manual low-level procedural programming. The high-level equational representation also gives better chances to utilize the inherent parallelism of a problem by 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.

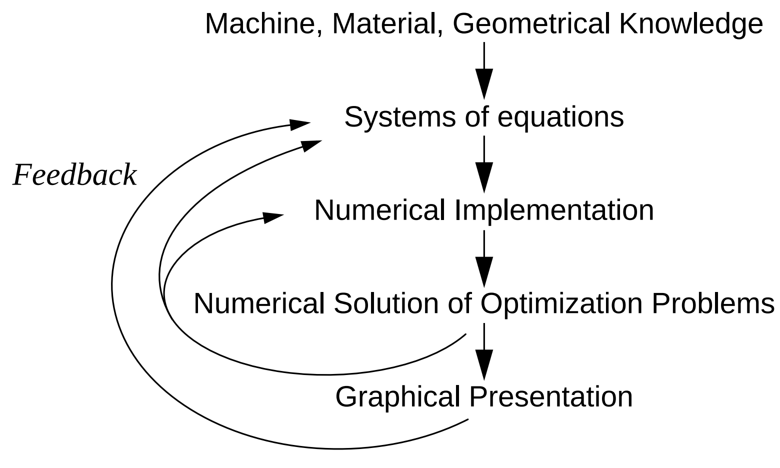


Figure 22-9 *The iterative process of modeling in traditional mechanical analysis. Often 50% of the time is spent on numerical implementation, usually FORTRAN programming.*

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 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 is usually too complex.

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

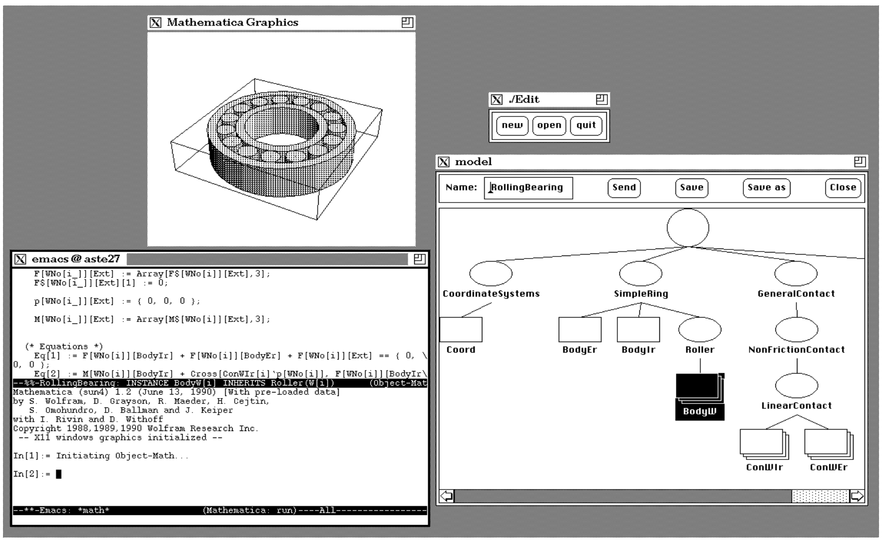


Figure 22-10. *The ObjectMath high-level modeling environment in use on a 200-equation rolling bearing example. The three-dimensional view of the bearing has been automatically generated from equations expressed in ObjectMath.*

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 and usage of part-of modeling 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.

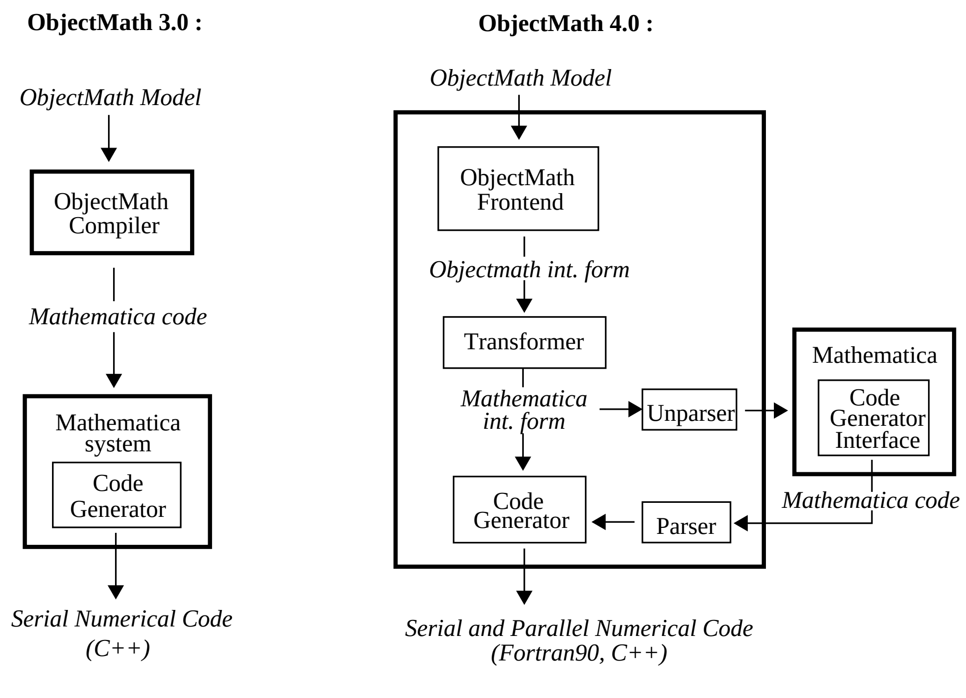


Figure 22-11 *The architecture of two generations of the translator part of the ObjectMath system (not including the browser). The latest version keeps a symbol table which is used by both the transformer and the code generator.*

The first ObjectMath prototype from 1991 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 released version of ObjectMath, from fall 1993, has been successfully used on several other applications problems at SKF. This version introduced multiple inheritance and the part-of relation into the modeling language, in addition to a comprehensive code generation facility for generating efficient C++ code from full models, the largest of size around 0.5 Mbyte source code. This is now routinely used to generate large portions of SKF's new bearing simulation programs.

A new version of the system (version 4), will be released spring 1995. It is a complete re-implementation of large parts of the system using the Cocktail compiler-writing toolbox developed in Karlsruhe. This gives new functionality and a stable foundation for further development. For example, an ObjectMath intermediate form has been defined, and a type system added to aid code generation to several target languages. The new code-generation facility can generate both Fortran and C++ code, in addition to code for parallel computers.

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 of the inherent parallelism from the application.

In the context of industrial applications in mechanical analysis, we have so far primarily explored generation of parallel code for solving systems of ordinary differential equations (ODEs), in addition to preliminary work on generating code for solving partial differential equations. Two approaches to extracting parallelism have been implemented and evaluated: extracting parallelism at the equation system level and at the single equation level, respectively.

To find parallelism at the equation system level, dependency analysis is performed on a graph describing dependencies between equations, using a standard algorithm for finding strongly connected components. The obtained strongly connected components are systems of equations to be solved. For many application models this works well, but for rolling bearing models we found that the corresponding systems of equations do not partition well into subsystems. This means that the equation system level approach is of restricted general applicability. Thus, we have recently focused on the equation-level approach which yielded significant parallelism for ODE systems solution. For bearing simulation applications the achieved speed-up is however critically dependent on low communication latency of the parallel computer.

22.5 List of publications

The following are publications by PELAB members during the years 1993-94 (together with a few from late 1992.) For the full list of publications, please refer to Appendix E.

PhD Theses

Kamkar, M. *Interprocedural Dynamic Slicing with Applications to Debugging and Testing*. Ph. D. Thesis No. 297, April 1993.

Licentiate Theses

Ringström, J. *Compiler Generation for Parallel Languages from Denotational Specifications*, Lic. May 1993. (Lic. No. 380),

Viklund, L. *Contributions to a High-level Programming Environment for Scientific Computing*, Jun. 1994. (Lic. No 437).

Nilsson, H. *A Declarative Approach to Debugging for Lazy Functional Languages,* Sep. 1994. (Lic. No. 450).

External publications

Andersson, N., Fritzson, P. Generating Parallel Code from Object Oriented Mathematical models. Accepted to the *5:th ACM Symposium on Principles and Practices of Parallel Programming (PPoPP'95),* Santa Barbara, California, Jul. 19 - 21, 1995.

Andersson, N., Fritzson, P. Overview and Industrial Application of Code Generator Generators. Accepted 1994 to the *Journal of Systems and Software*, Feb. issue, Elsevier Science Publ. Company Inc.

Engelson, V., Auguston, M. The programming language RIGAL, a tool for compiler writing. *In Proc. of NWPER'94, Nordic Workshop on Programming Environment Research*, pp. 167-176, LU-CS-TR-94-127 Dept. of Computer Science, Lund Institute of Technology, Jun. 1-3, 1994.

Fritzson, D., Fritzson, P., Viklund, L., Herber, J. Object-Oriented Mathematical Modelling - Applied to Rolling Bearings. In *Computers & Structures*, Vol. 51, No. 3, pp. 241-253. 1994. (Pergamon Press).

Fritzson, P., Fritzson, D. 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.

Fritzson, P., Andersson, N. Generating Parallel Code from Equations in the ObjectMath Programming Environment. In *Proc of the 2nd Int. Conf. of the Austrian Center for Parallel Computation*, Oct. 1993. LNCS 734, Springer-Verlag.

Fritzson, P., Auguston, M., Shahmehri, N. Using Assertions in Declarative and Operational Models for Automated Debugging. *Journal of Systems and Software*, 25:223-239, Jun. 1994.

Fritzson, P., Engelson, V., Viklund, L. Variant Handling, Inheritance and Composition in the ObjectMath Computer Algebra Environment. In *Proc. of DISCO'93 - Conf. on Design and Implementation of Symbolic Computation Systems*, LNCS 722, Sep. 1993.

Fritzson, P., Viklund, L., Fritzson, D., Herber, J. ObjectMath - An Environment for High Level Mathematical Modeling and Programming in Scientific Computing. Accepted Sep. 1994 to *IEEE Software*.

Kamkar, M. A Comparative Classification of Program Slicing Techniques. Accepted 1993 to the *Journal of Systems and Software*.

Kamkar, M., Fritzson, P., Shahmehri, N. Interprocedural Dynamic Slicing with Applications to Interprocedural Data Flow Testing. In *Proc of the IEEE Conf. on Software Maintenance, CSM-93*, San Diego, Sept. 1993.

Kamkar, M., Shahmehri, N., Fritzson, P. Three Approaches to Interprocedural Dynamic Slicing. In *Proc. of EUROMICRO'93 (Journal of Microprocessing and Microprogramming),* Vol. 38, No. 1-5, Step. 1993. Published by North-Holland.

Lennartsson, B., Shahmehri, N., Bonnier, S. On exploiting declarative programming and parallel execution in computer based real-time systems. In *Proc. of the IEEE Workshop on Systems Engineering of Computer Based Systems. ECBS'94*, IEEE, 1994.

Malmén, J., Fritzson, P. Algorithmic Debugging for Parallel Message-Passing Procedural Languages. In *Proc. of the Swedish National Real-Time System Symp.*, KTH, Stockholm, Aug. 25-26, 1993.

Nilsson, H., Fritzson, P. Declarative Algorithmic Debugging for Lazy Functional Languages. *The Journal of Functional Languages*, 4(3), 337-370, Jul. 1994.

Nilsson, H., Fritzson, P. Lazy Algorithmic Debugging: Ideas for Practical Implementation. *AADEBUG'93 -1st Int. Workshop on Automated and Algorithmic Debugging*, May 1993. In LNCS 749, Nov., 1993

Nordling, P., Fritzson, P. Solving Ordinary Differential Equations on Parallel Computers - applied to Dynamic Rolling Bearing Simulation, In Dongarra, Hansen and Wasniewski, eds., *PARA'94 Workshop on Parallel Scientific Computing*, UNI-C Danmarks EDB-Center for Forskning och Uddannelse, Copenhagen, Jun. 1994., LNCS 879, Springer-Verlag.

Pettersson, M. RML – A new language and implementation for natural semantics. In *Proc. of PLILP'94, Programming Language Implementation and Logic Programming*, LNCS 844 (pp. 117-131), Springer-Verlag, Aug. 1994. Also as report LiTH-IDA-R-94-25.

Pettersson, M. The RML2C compiler – present and future. In Magnusson, Hedin, Minör eds., *Proc. of NWPER'94, Nordic Workshop on Programming Environment Research*, pp. 303-304, LU-CS-TR-94-127, Dept. of Computer Science, Lund Institute of Technology, Jun. 1-3, 1994.

Ringström, J., Fritzson, P., Pettersson, M. Generating an Efficient Compiler for a Data Parallel Language from Denotational Specifications, In *Proc. of CC'94 – Int. Conf. of Compiler Construction*, Edinburgh, Apr. 1994, Proc. in LNCS 786, 1994.

Shahmehri, N., Kamkar, M., Fritzson, P. Usability Criteria for Automated Debugging Systems. *AADEBUG'93 -1st Int. Workshop on Automated and Algorithmic Debugging*, May 1993. In LNCS 749, Nov., 1993. Also accepted Mar. 1994 to the *Journal of Systems and Software*.

Toller, E. Action-oriented programming with joint actions. In Magnusson, Hedin, Minör (eds.), *Proc. of NWPER'94, Nordic Workshop on Programming Environment Research*, pp. 303-304, LU-CS-TR-94-127 Dept. of Computer Science, Lund Institute of Technology, June 1-3, 1994.

Westman, R., Fritzson, P. Graphical User Interfaces for Algorithmic Debugging. *AADEBUG'93 -1st Int. Workshop on Automated and Algorithmic Debugging*, May, 1993. In LNCS 749, Nov., 1993.

Research reports, submitted papers, etc.

Auguston, M., Fritzson, P. PARFORMAN - An Assertion Language for Specifying Behavior when Debugging Parallel Applications. Submitted 1992 to *IEEE Transactions on Software Engineering*., Revised 1994.

Hoffner, T., Fritzson, P., Kamkar, M. Evaluation of Program Slicing Tools. Submitted Nov. 1994 to the *2:nd Int. Workshop on Automated and Algorithmic Debugging (AADEBUG'95).*

Hoffner, T. Evaluation and Comparison of Program Slicing Tools. Research report, LiTH-IDA-R-95-01, Jan. 1995.

Pettersson, M. Main-Memory Linear Hashing - some Enhancements of Larsson's Algorithm. LiTH-IDA-R-93-04. Submitted to Software – Practice and Experience.

Pettersson, M. The Definition of RML. Forthcoming technical report, spring 1995.

Pettersson, M. Portable tailcalls in C. Forthcoming research report, Dept. of Computer and Information Science, Linköping University, spring 1995.

Viklund, L., Fritzson, P. ObjectMath – An Object-Oriented Language for Symbolic and Numeric Processing in Scientific Computing. Under review for *Scientific Programming*, Second round of review, Sep., 1994.

Zuhdy, B., Fritzson, P. Implementation of the real-time functional language Erlang on a massively parallel platform. Submitted Nov. 1994 to *the Conference on High Performance Functional Computing*, Denver, USA. Also as research report, Dept. of Computer and Information Science, Linköping University, 1995.

Books and proceedings

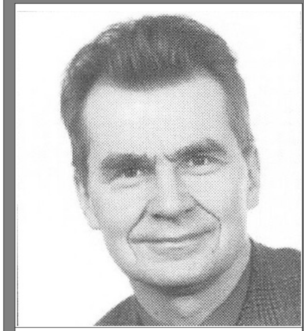
Fritzson, P., ed. *Proc. of AADEBUG'93 - the 1:st Int. Workshop on Automated and Algorithmic Debugging*, Linköping, May 3-5, 1993, LNCS 749, Springer-Verlag.

Fritzson, P., ed. *Proc. of CC'94 - the 5:th Int. Conf. on Compiler Construction*, Edinburgh Apr., 7-9, 1994., LNCS 786, Springer-Verlag.

Fagerström, J. *Objektorienterad systemutveckling - en introduktion*, Studentlitteratur, 1993.

RKLLAB

23
The Laboratory for Representation of Knowledge in Logic



Erik Sandewall
director

Artificial intelligence
Real-time systems
Non-monotonic logic
Planning
Autonomous agents
Temporal reasoning

Some of the members of the Laboratory for Representation of Knowledge in Logic.
From the back: Ulf Söderman, Thomas Drakengren, Choong-Ho Yi, Leif Finmo
Lise-Lotte Svensson, Christer Bäckström, Dimiter Driankov, Peter Jonsson, Erik Sandewall
Simin Nadjm-Tehrani, Lin Man, Sergio Brandani, Jacek Malec.
Front: Silvia Coradeschi.

23.1 Laboratory members

The following researchers have participated in RKLLAB's activities during the years 1993-1994.

Laboratory leadership, administrative and technical staff:

Erik Sandewall, Ph. D., professor
Lise-Lott Svensson, administrator
Leif Finmo, research engineer

Laboratory members having or completing a Ph. D. degree during the period:

Douglas Busch
Christer Bäckström
Patrick Doherty
Dimiter Driankov
Jacek Malec
Simin Nadjm-Tehrani

Visitors during extended periods:

Witold Łukaszewicz, Professor at Univ. of Warsaw, Poland
Jana Koehler, Ph. D., DFKI, Saarbrücken, Germany
Pavlos Peppas, Professor at Macquarie Univ., Australia
Lin Man, Tsinghua Univ., Beijing, China (scholarship for preparing for Ph. D. study)
Silvia Coradeschi, Univ. of Pisa, Italy (Erasmus student doing her M. Sc. project)

Laboratory members having or completing a licentiate degree during the period:

Lars Degerstedt
Magnus Morin
Hua Shu, Soft Center, Ronneby
Per Österling

Other graduate students:

Thomas Drakengren, B. Sc.
George Fodor, ABB Marine, Västerås, M. Sc.
Peter Jonsson, M. Sc.
Lars Karlsson, M. Sc.
Ezra Kim, M. Sc.
Tommy Persson, M. Sc.
Ulf Söderman, M. Sc.
Choong-Ho Yi, M. Sc.

23.2 Topic of Research: Reasoning about Actions and Change

The main research topic of RKLLAB is *computational methods for reasoning about actions and change*. Its purpose is to make possible the following types of automated reasoning about the behavior of a system over time:

- *Prediction* of the future behavior of a given object system (for example, a mobile robot or an industrial plant) on the discrete level of description;
- *Planning* of a sequence of actions which, with sufficient certainty, will take the object system to a given goal state;
- *Chronicle supervision*, where an incoming flow of basic sensory data are compared with an 'agenda' of expected changes (possibly the agenda is a plan that is being executed), so that unexpected events or other deviations from the agenda can be recognized;
- *Plan execution*, involving supervision of the executing plan and *plan revision* when problems arise;
- *Diagnosis*, involving an analysis of which events (which are not directly observable) must have occurred in a given object system, when certain observations of the current and recent states of the system are given.

Typically, the object system about whose behavior one is reasoning is a physical one. Possibly, but not necessarily, it occurs in an industrial environment. The case where the object system is another piece of software is not formally excluded, but it is not the main consideration.

The reasoning tasks of prediction and planning are the ones that we have concretely addressed so far. The other three represent additional, potential applications for the same theory and methodology.

This problem area is presently a topic of active international research, spanning over several disciplines. We approach it from a background of computer science and artificial intelligence, but in active cooperation with other disciplines and, in particular, with the control theory group in our university (headed by Professor Lennart Ljung).

Current challenges in this area of research include *modelling* (how is an aspect of the physical world best described on the level of discrete change?), *logics* (which systems of formal logic are capable of obtaining exactly those conclusions that are warranted in the system being described?), *algorithms* for planning and prediction, and *implementation techniques* including the choice of software architectures, inference 'engines' based, for example, on logic programming or on tableau methods, and compilation and other preprocessing techniques.

We originally entered this research topic because of a number of industrial applications, namely in industrial robotics, autonomous vehicles, and co-driver systems. The prototypical application for us is an autonomous intelligent agent which needs to continuously predict the immediate future and to choose its actions in order to avoid problems and to achieve goals. However, the same technology is used elsewhere for problems of human-machine interaction, common-sense reasoning, natural language understanding, etc.

In our research we address the following more specific research topics:

1. Semantics-based assessment of nonmonotonic entailment methods for actions and change.
2. Implementation techniques for nonmonotonic entailment methods.
3. Algorithmic methods for planning and temporal reasoning.
4. Hybrid modelling of physical systems.
5. Design methodology for intelligent autonomous robotic systems.
6. ‘Fuzzy’-related design principles for intelligent autonomous robotic systems.
7. Several industrial application projects.

Internally, we view these topics as closely related, and there is a high degree of interaction between our researchers in these topics. From an external point of view, however, these topics have their own communities, literatures, and research styles. For this reason, and also because each of our researchers is primarily affiliated with one of the topics, we choose to present them in separate sections of this chapter. This year, the first three topics will be presented in more detail than the latter ones; we have a tradition of varying the emphasis between the years.

23.3 Semantics-based assessments for nonmonotonic entailment methods for actions and change

The group for reasoning about actions and change has the following members at present: professor Erik Sandewall (group leader), professor Witold Łukaszewicz (Warsaw, visiting periodically, shared with the group of Dr. Patrick Doherty), Ph. D. Pavlos Peppas (Sydney, visiting researcher for parts of 1994–95), graduate students Thomas Drakengren and Choong-Ho Yi, and masters students Silvia Coradeschi and Sergio Brandano (Erasmus students).

23.3.1 Topic of research and state of the art

Our research topic is to investigate nonmonotonic logics for actions and change with a systematic methodology. Earlier work in this field was usually done by *ad hoc* methods: new logics were proposed and defended by their intuitive plausibility and by a small number of examples where they (were claimed to) give the intended results. Instead, we address the question “for which class of reasoning tasks does a given logic give exactly the intended set of conclusions?”, based on a precise definition of intended conclusions. A result of this kind is an *assessment*. A special case of assessment is *validation*, where the definition of intended conclusions is given and one asks whether a given logic always gives the intended set of conclusions (only restricted by the limitations on the definition of intended conclusions).

23.3.2 Research results

Our work so far has addressed the relatively limited case of scenario descriptions with strict inertia (no qualification, ramification, concurrency, or surprises), but allowing nondeterministic actions and actions with extended duration in time. They have been reported in conference, journal, and handbook articles [31, 33, 34], and in a monograph that has appeared very recently [32].

The development of the coherent framework for the assessment-oriented approach has been a major part of that work. It included:

- The definition of the underlying semantics. Our choice of underlying semantics is based on the use of an *ego-world game*, that is, a formalized version of the interaction between the intelligent system and the world in which it operates. In the case of strict inertia, the "ego" has a very simplistic definition, and the "world" is defined in terms of *trajectories*. We define a trajectory as a finite sequence of partial states of the world. From a semantic point of view, each action denotes a set of trajectories, representing the various ways that the action can be executed.
- The definition of an ontological (and epistemological) *taxonomy* of scenario descriptions. This taxonomy is based on the underlying semantics, and it is used as the coordinate system for expressing the range of applicability of the various entailment methods.
- The definition of a coherent perspective on the various temporal reasoning operations that may be needed in applications: prediction, explanation, various varieties of planning and replanning, etc.

With this framework in place, we have assessed the range of applicability of about one dozen nonmonotonic entailment methods (that is, variants of the logic which only differ with respect to the definition and the use of the preference relation on models). The following difficulties are allowed for in the results:

- Forward-branching as well as linear, integer time: definition and assessment of entailment methods for both alternatives.
- Actions with extended duration in time.
- Composite actions (composed using sequencing, conditionals, and loops): definition and assessment of entailment methods which apply for composite actions.
- Both lower bounds and upper bounds on the range of applicability have been identified, and are sometimes equal (so that the range of applicability has been precisely determined).
- Range of soundness as well as range of completeness has been identified.

A number of more general results can be obtained within the already existing framework, or with relatively minor extensions of it. Thus, *Choong-Ho Yi* is just completing a licentiate thesis where the range of applicability is analyzed for prototypical chronological minimization, PCM (with and without filtering), for a restricted case of concurrent actions, namely concurrent actions without interaction [37]. It seems a priori likely that PCM will apply to some cases of concurrent actions with interaction, and that the modified method also using occlusion will cover a significantly broader set of cases. We propose to continue his research in these directions.

Our visiting researcher *Pavlos Peppas* addresses the problem of ramification. He has his background in a group that has traditionally used the postulate-based approach, and contributes those impulses to our group. The problem of ramification is also addressed in some present work by *Erik Sandewall*, where he defines intended models for the ramification case using the variant of the approach of Kartha and Lifschitz, and investigates range of applicability for other current methods on that basis [forthcoming article].

Silvia Coradeschi investigates the problem of scenarios with unreliable observations. We view this as a first attempt at a difficult problem which is of great practical importance.

Thomas Drakengren has extended and corrected earlier results on the use of composite actions.

The main effect of these and other similar projects is to obtain the full benefit from the existing framework, in particular the trajectory semantics that has been used in our work so far.

23.4 Implementation techniques for nonmonotonic entailment methods

The nonmonotonic reasoning group currently has the following members: assistant professor Patrick Doherty (group leader), professors Witold Lukaszewicz and Andzrej Szalas (Warsaw, visit 1-3 months per year), and graduate student Lars Karlsson (completing his licentiate degree in February, 1995). In addition we have cooperation with professor John-Jules Meyer (Utrecht) and his group.

23.4.1 Topic of research and state of the art

The research problem for our group is to find efficient implementations for (nonmonotonic) logics for actions and change. We are particularly interested in those logics using explicit, metric time, since this appears to be essential for robotics-type applications where time-stamped sensory data are involved. We also focus on those logics for which there exists an assessment of their range of applicability, since in this way one can rely on the logic from the point of view of the application. In particular, we have investigated many of the different nonmonotonic temporal logics that are defined and assessed in Sandewall's book "Features and Fluents" [32]. As assessments are obtained for logics for broader ontological classes (compare section 23.3), we will address their implementation problems as well.

Much of the current research in this field has its roots in methods for general theorem-proving (where we include the work on nonmonotonic logic programming). Such methods may be specialized to the particular conditions of reasoning about time, actions, and change. One can distinguish the following basic approaches to implementation:

- Semantic-based approach
- Syntactic-based approach
- Operational-based approach

(The discussion will be restricted to logics with explicit time, omitting work based on modal-temporal logics for reasons of space). The semantics-based approach is related to model-checking, but modified towards dealing with minimal models. The syntactic-based approach appeals to more conventional theorem-proving techniques. The operational-based approach appeals to the strong formal relation between the underlying semantics and the models associated with the temporal non-monotonic logics.

There are also other implementation methods which are more remote from the logical basis, such as truth maintenance systems and planning systems. The principled development of practical software systems for planning is a very active research area. It is possible to view a planning system abstractly as a very specialized system for temporal reasoning, but practical planning systems are usually described, instead, in more pragmatic and algorithmic terms.

23.4.2 Research results

So far we have addressed the following approaches:

- Syntactic-based general-purpose approach
- Operational-based general-purpose approach
- Implemented planning systems

23.4.2.1 The syntactic-based approach

In Doherty and Lukaszewicz [19], we provided syntactic characterizations of each of the logics of action and change in terms of classical FOL and circumscription axioms. The intent of the exercise was to take advantage of a number of reduction results which exist in the literature on circumscription. It turned out that one of the logics in Features and Fluents, PMON, had a circumscriptive definition that was found to be reducible to the first-order case. Not only does PMON cover the broadest class of action scenarios considered so far in the Features and fluents framework, but it subsumes the other logics described in the book. This work is reported in [17]. By restricting syntax for action types in PMON, it can even be shown that certain scenarios can be compiled into logic programs [16].

We are currently investigating the generic case of reducing 2nd-order circumscription axioms to equivalent 1st-order formulas. Initial results in this direction are promising. In Doherty et. al. [18], we present implementable algorithm which takes a 2nd-order formula as input and either returns an equivalent 1st-order formulas as output or terminates. Of course, termination does not mean that there doesn't exist an equivalent 1st-order formula, just that the algorithm can not generate one. More surprisingly, this algorithm provably subsumes existing reduction results in the literature. We propose to continue work in this direction in the following manner:

- Integrate special purpose temporal constraint algorithms with the circumscription reduction algorithm.
- Specialize the circumscription reduction algorithm to take advantage of the various syntactic and semantic constraints associated with the various ontological classes.
- Generalize the algorithm to deal with, as yet non-reducible classes of 2nd-order circumscription axioms. In particular we expect a reduction to fixpoint logics for certain types of temporal nonmonotonic entailment. Results in this direction will also be useful for a number of situation calculus based approaches.
- Study how closure axioms in the context of the explanation closure technique can be automatically generated for theories via the reduction of circumscription axioms.

23.4.2.2 The operational-based approach

An obvious similarity between actions and commands of programming languages, and between action scenarios and programs, suggests that it may be fruitful to formalize reasoning about action and change using methods that have been introduced to reason about programs. Dijkstra's approach provides one such possibility.

In [24], we have applied Dijkstra's methodology to formalize reasoning about deterministic actions. In [25] we have generalized the approach to actions where non-deterministic and indirect effects are allowed.

23.4.2.3 Reconstruction of known systems

A number of practical software designs have been developed, in a more or less *ad hoc* manner, for action planning and to some extent for other reasoning operations. In principle, these systems can be seen as realizations of reasoning operations, but in practice they have not been conceived and built that way. However, they do embody a considerable amount of practical know-how, which one would like to extract and transfer to the context of systematic logic-based methods.

Our graduate student, Lars Karlsson, has just completed a licentiate thesis that takes a step in that direction. Starting from the general "Features and fluents" framework, he has reconstructed a number of classical planners in a systematic way: STRIPS, TWEAK, SNLP, and the sequential restriction of NONLIN. This has given an insight into how one can go from the logical basis to a practically working system.

23.5 Algorithmic Methods for Planning and Temporal Reasoning

The group for complexity of planning presently consists of asst. prof. Christer Bäckström and graduate student, M. Sc. Peter Jonsson. The group cooperates with Prof. Bernhard Nebel of the Univ. of Ulm (Germany), and with Dr. Inger Klein in the Division of Automatic Control of our University. In 1994 we have had a three-month visit by Dr. Jana Koehler, DFKI, Saarbrücken, Germany.

23.5.1 Topic of Research and State of the Art

23.5.1.1 Action Planning

The *action planning problem*[1] can be briefly described as follows: Given an initial (current) state, a goal (desired) state, and a collection of operators, the planning problem is to find a sequence of operators that, when applied in the initial state, changes the state of the world to the goal state. Although the research field of action planning has reached the age of some 25 years, many unsolved problems remain and the attempts at a theoretical understanding of the field has evolved only during the past five to ten years.

One of the most important problems faced by the planning research community is how to plan efficiently. Planning is known to be very hard in general; even for very simple formalisms, deciding whether a solution exists for a given problem instance is undecidable in the first-order case and PSPACE-complete in the propositional case. However, one must keep in mind that these complexity figures apply to the hardest problems that can be encoded in each formalism and they say nothing about the inherent complexity of application problems.

Therefore, there are likely to be many planning applications that do not need the full expressive power of even the standard formalisms (although their modelling may still benefit from adding certain new constructs to these formalisms). Similar observations have motivated research into identifying restrictions that make hard problems tractable in other areas than planning, e.g. *concept languages*, and *abduction*. Tractable planning problems have been identified, especially by ourselves (see Section 23.5.2.1) but also by others.

23.5.1.2 Applications in Automatic Control

Although it is of general, theoretical nature, our research has to a large extent been motivated by the application area of *sequential control* within the subarea *discrete event dynamic systems* of *automatic control* where many problems can be viewed as planning problems. Researchers in this area see the lack of mathematically well-founded methods with provable properties as one of the major problems with AI planning. Furthermore, a recent study of how computer software is used in large-scale industrial control applications has shown that the industry demands better mathematical tools for modelling dynamical systems of combinatorial nature, since heuristics and expert systems are not considered safe. We have found this a strong reason for insisting in doing research into provably correct planners and complexity analyses of restricted planning problems.

One of our target test applications is the LEGO Car Factory, which is a miniature, automated assembly line for LEGO cars, used for undergraduate laboratory assignments in sequential control at the department of EE at Linköping Univ. This application is interesting because it contains many of the problems and properties found in

1. We use action planning to denote the traditional AI planning field within the nowadays large and fuzzy area of AI planning.

real industrial processes, while being small enough to be used as a test application and to analyse theoretically.

23.5.2 Research Results

Our main line of research during the past five years has been to study the computational complexity of various restricted planning problems, especially identifying tractable ones, to some extent motivated by applications in automatic control. This research has many connections into other areas, which has forced us, at times, to also carry out research into related areas within planning and temporal reasoning.

23.5.2.1 Planning Complexity

The anticipated need for provably correct and efficient planners has motivated us to follow the restricted-problems line of research, trying to identify tractable planning problems.

In 1990 we presented the, to our knowledge, first ever tractable planning problem, the SAS-PUBS problem [8, 10]. In contrast to other, more recent publications into this topic, we did not use the formalism *per se* as a guide to which restrictions to use. Instead we identified four inherent restrictions in a toy problem in the sequential control domain, the tunnel problem [1]. These are essentially syntactic restrictions on the sets of operators and state variables and they can be tested in polynomial time [1].

Continuing with a bottom-up approach, we repeatedly removed restrictions while retaining tractability. This resulted in the SAS-PUS [9], SAS^+-PUS [1, 2] and SAS^+-US [12, 13] planning problems and we developed sound and complete, polynomial-time algorithms for each of these problems. This first phase of our research was concluded by an exhaustive analysis of the complexity resulting from all combinations of the restrictions [12, 13]. In this analysis we studied both the existence problem (finding out whether a solution exists or not) and the generation problem (generating a solution or answer that doesn't exist) and we considered both bounded and unbounded solutions. The result was that under the restrictions considered, SAS^+-PUS is the maximal tractable problem for bounded planning and SAS^+-US is the maximal tractable problem for unbounded planning.

Realizing that syntactic restrictions would not suffice much further, we initiated the next research phase of studying structural restrictions on the state-transition graph. Arbitrary such restrictions can be costly to test, since this graph is of exponential size. We have identified three structural restrictions which can be tested in polynomial time and which together define a planning problem, the SAS^+-IAO problem [22], which is tractable for optimal planning and properly extends the SAS^+-PUS problem, thus generalizing the previous results. We have developed a sound and complete algorithm for this problem and made an exhaustive analysis of the complexity for all combinations of the four syntactic restrictions and the new structural ones [21].

The SAS^+-IAO problem allows only certain, limited types of cycles in the state transition graph, and is not quite sufficient for modelling our main target application – the LEGO car factory. However, by attempting such a modelling and using this as a feedback for the theory, we have developed a provably correct, polynomial-time planning method based on the SAS$^+$-IAO algorithm which handles the LEGO car factory [23].

The method builds on automatically partitioning the problem instance into two parts. The first of these is solved using the SAS$^+$-IAO algorithm, resulting in a plan skeleton which can be filled in by applying the SAS$^+$-IAO planner to the second part, somewhat similar to hierarchical abstraction.

In all research above, we have used the SAS$^+$ formalism, which is a slightly restricted version of the action structures formalism [30] and can be viewed as a variation on propositional STRIPS. We have proven [1, 2, 5] the SAS$^+$ formalism to be expressively equivalent, under polynomial reduction, to two variants of the propositional STRIPS language and to the ground TWEAK language. This result shows that neither negative pre-conditions, negative goals, partial initial states, nor multi-valued state variables add to the expressiveness of propositional STRIPS. Despite this equivalence, it was crucial that we used the SAS$^+$ formalism, since some of the restrictions we have studied are not very useful for binary state variables. Recent results [6] also show that, despite the equivalence, the formalisms behave differently with respect to actual planning algorithms—it may be exponentially more costly to reencode a SAS$^+$ problem in STRIPS and then solve it using a standard partial-order planner, SNLP, while there is no efficiency loss in the other direction.

Except for continuing the search for tractable planning problems, our ongoing research includes studying the relationship between complexity and the length of optimal solutions and comparing our specialized polynomial-time planning algorithms with general-purpose planners equipped with heuristics.

23.5.2.2 Other Related Research

Dean and Boddy analyzed the problem of *temporal projection*, *i.e.* deciding the effects of a set of partially ordered events, for a certain variant of propositional STRIPS. They found the problem to be NP-hard, even under severe restrictions, and conjectured these complexity results to carry over to the problems of *planning*, *plan validation* and *story understanding*. We showed [11, 28, 29] that for all three of these problems, there are restrictions making them tractable, while the corresponding temporal projection problems are still NP-hard under the same restrictions, thus disproving Dean and Boddy's conjecture. We further showed that plan validation is, in fact, tractable for the important class of all unconditional plans [28, 29].

Our research into planning complexity has also forced us to reconsider some results from the literature on converting total-order plans into partial-order plans, subject to certain constraints. We found that two algorithms in the literature for converting total-order plans into partial-order plans that are *least constrained* or have a *shortest parallel execution* respectively, seem not to satisfy these optimality claims. In the absence of definitions from the authors of the algorithms, we suggest [3] some reasonable definitions for least-constrainment and optimal parallel executions. Most of these criteria are proven NP-hard and even impossible to approximate within a constant. Further, one least-constrainment criterion is tractable to achieve, but the algorithm in the literature fails to satisfy even this kind of (local) optimality. Continuing this line of research we have further found [4] that while a sequential plan will necessarily execute slower when actions are added to it, a parallel plan can in some cases be made to execute faster by adding actions to it.

Some of our ongoing, unpublished research has resulted in some spin-off results on hierarchical abstraction in planning, leading to deepened insight into this area. While it is well-known that hierarchical abstraction can speed up planning exponentially under ideal circumstances, we have shown that in other cases abstraction hierarchies can make planning exponentially slower—even forcing the generation of a plan which is exponentially longer than the optimal one. Furthermore, certain algorithms from the literature (ALPINE and HIGHPOINT) for automatically generating abstraction hierarchies are not able to avoid this problem, and theoretical results support that this may not even be possible.

23.6 Modelling and analysis of hybrid systems

The group for modelling and analysis of hybrid systems consists of Dr. Simin Nadjm-Tehrani, graduate student Ulf Söderman (finishing his Ph. D. in May, 1995), in cooperation with Dr. Jan-Erik Strömberg in the Division of Automatic Control in our university.

By a hybrid model we mean a description of a system which includes both continuous and discrete elements. Earlier work in our group has used hybrid models in two different ways:

(1) as a means for describing the interacting behaviour of an embedded real-time computer system and its physical environment;

(2) as a means of modelling the behaviour of physical systems which exhibit discrete structural changes, typically due to existence of a switching device such as a gear-box or a relay.

The first line of research is intended to lead to formal analysis methods arriving at a proof of expected or desired properties of the embedded system. Interdependence between discrete and continuous system descriptions in terms of state transition systems has been studied [26]. This work has among others considered the case when one is satisfied with a description of physical quantities in terms of distinct ranges, such as high/ medium/ low or increasing/ constant/ decreasing. The question addressed was the loss of information when the continuous description is replaced by a discrete one [27]. An alternative to the systematic derivation of discrete models is the use of *Hybrid Transition Systems* for representation of the embedded system. This formalism incorporates differential and algebraic equations for description of continuous change in each (discrete) mode of the physical environment. Modular development of models, and parallel composition is supported by the formalism.

In the second line of research we have developed an extension of the bond graph language by the introduction of a `switch` concept [35]. Bond graphs are a classical method in control engineering for describing the abstract behavior of a system, but traditional bond graphs can only describe continuous change. Our work has resulted in a methodology for the systematic modelling of physical systems which allows for discontinuous approximations of switching devices. In the final stage of this methodology a mathematical representation of a physical system is obtained in a variant of hybrid transition systems.

Hybrid transition systems have been applied for modelling several standard as well as some non-trivial examples. The current work in the group includes application of the modelling techniques in a realistic setting (see the description under the COHSY project). We are now in a position to study possible approaches to the problem of analysis. Automatic verification of hybrid systems is undecidable even for very restricted classes of systems. The aim of our work is to analyze a hybrid model using a combination of logical reasoning and traditional analysis techniques for continuous systems. Identifying interesting properties used as a requirements specification for hybrid systems is also an area for research.

23.7 Design methodology for intelligent autonomous systems

The group for design methodology for intelligent autonomous systems consists of Dr. Jacek Malec and graduate student Magnus Morin.

Intelligent autonomous systems, realized for example as robotic vehicles, provide an avant-garde application domain for the techniques that were described in previous sections. The synthesis of such systems involve a number of design issues, in particular:

- The choice of a hardware/software architecture which can serve as a backbone for the various contributing modules. In this context the use of layered architectures is often advocated.
- The choice of strategy guiding system implementation. The traditional model-based top-down approach, often adopted in AI and control engineering, is confronted with the bottom-up, behaviour-based strategy proposed by Brooks. Recently, many compound (i.e. mixing elements of both approaches) policies have been proposed.
- The choice of language for specifying the expected reactive (stimulus-response) behaviour of the designed system. Assuming that a discrete description is sufficient, there exist a number of well-known languages to choose from: automata, statecharts, Petri nets, etc.

Rather than proposing and implementing additional approaches, we have focused on assessing properties of the existing ones. This research is using a generic three-layered software architecture as a frame of reference. We have participated in the systematic comparison of various architectures and modelling languages in the context of specific classes of applications. For this purpose we have also investigated computational models and real-time properties of the software tools associated with the layered architecture. The results we have obtained so far justify our choice and prove its usefulness in several application domains.

23.8 'Fuzzy'-related design principles for continuous and discrete control

The research is done by Dr. Dimiter Driankov in a close cooperation with Siemens AG, Corporate Research and Development, Munich, Germany and ABB Industrial Systems, Dept. LA, Västerås, Sweden.

Fuzzy control of continuous nonlinear processes has recently attracted widespread attention both within the process control industry and the conventional control community. Our group has been involved in research in the area of fuzzy control and fuzzy logic since 1988 which is manifested in a recent book on Fuzzy Control, as well as in a long list of research publications. Recent work has been concerned with design techniques for fuzzy control based on deep models of the process under control and utilizing the power of existing modern nonlinear and linear conventional methods, e.g., fuzzy sliding mode control. This is in contrast with current work on fuzzy control which tends to use shallow models reflecting heuristic knowledge about the process behaviour and/or the related human operator control actions.

The present and future research focus is on the utilization of Lyapunov linearization for the systematic design of a particular class of fuzzy control rules (Takagi-Sugeno fuzzy controllers) that approximate in a smooth piecewise manner a nonlinear process and are capable of stabilizing a nonlinear process around an arbitrary set point. This approach makes possible the use of conventional linear control techniques for the purpose of stability analysis, optimal control design, and robustness analysis. We are also concerned with utilizing the principles of sliding mode fuzzy control for the simplification of the stability analysis of Takagi-Sugeno fuzzy controllers as well as the use of this type of fuzzy controllers within a gain scheduling framework. The research results will be published in 1996 by Springer Verlag in a book on model-based fuzzy control.

The second research topic of interest concerns general purpose PLC's (Programmable Logic Controllers) and the work is done in cooperation with ABB Industrial Systems, Dept. LA. A PLC can be abstracted as a discrete control algorithm executing different sequences of <condition, control action> pairs, where the 'control action' is executed whenever a predefined 'condition' holds. It is also expected that the prescribed effect of the control action (the 'condition'-part of a consecutive <condition, control action> pair) will take place if and only if the 'modelling assumptions' used in the construction of the <condition, control action> pairs are valid and not violated during the execution of the control algorithm. Of particular interest are those modelling assumptions which are not represented in the control algorithm, the so called 'ontological assumptions'. Since the controller itself is "unaware" of the ontological assumptions, possible violations of these cannot be detected by it. A control paradigm which can be used to provide a proof that the ontological assumptions are wrong or violated and thus can decide whether these assumptions underlying a control algorithm are adequate for achieving prespecified control goals is called *ontological control*. The major goal of our work is to develop the formal means and implementation techniques for the design of an *ontological controller.* Such a controller is an autonomous device intended to supervise a conventional PLC in order to:

- detect dynamically when the PLC acts in a situation where the ontological assumptions underlying the control algorithm are wrong or violated, and
- when possible, to move the control system into a state from which it can regain its control over the environment and still achieve a prespecified goal despite previous violations of ontological assumptions.

23.9 Areas of applied research in our group

The following are brief descriptions of those applied research projects where we are presently engaged.

23.9.1 Driver Assistance and Local Traffic Management (DALTM) (Malec)

We are participating in a project within the Driver Assistance and Local Traffic Management area of the Swedish RTI programme, with Volvo Cars, SAAB Automobile, the Swedish National Road Administration (Vägverket) and NUTEK as the industrial partners. The project is aimed at designing and implementing a *driver information unit* intelligently managing the information and warnings passed to the driver by several active modules mounted in a test car (autonomous intelligent cruise control, friction estimation, communication with roadside, etc.). To this end we use the software architecture and design methodology developed in our group (and mentioned earlier in section 23.7). The prototype system has been successfully tested in real motorway traffic in late 1994.

23.9.2 Hybrid Modelling and Verification of COmplex Heterogeneous SYstems (COHSY) (Nadjm-Tehrani)

The work in this project aims to integrate modelling, synthesis and analysis techniques rooted in different disciplines; i.e. various engineering domains (hydraulic, mechanical, electrical, automatic control), and computer science (software development methods, formal techniques, database technology). The industrial participants are SAAB military aircraft, Volvo Aerospace and VOAC Hydraulics.

The common aspect of the different applications is the complexity arising due to the interaction of embedded computer systems with the physical environment and the human operator. The distributed software modules perform tasks of control, chronicle supervision and diagnosis.

In our part in the project we have so far studied decompositions of the complex system in accordance with the software architecture referred to in section 23.7, and application of formal modelling and verification techniques (including those described in section 23.6) in order to prove safety and timeliness properties of some reactive control modules. The novelty of the approach has been to incorporate the model of the physical environment as derived by bond graph techniques, for the purpose of analysing the properties of the embedded software.

23.9.3 Marine Diesel Engines (Driankov)

Mr. George Fodor from ABB Industrial Systems AB, Dept. LA is completing a Ph. D. thesis under the supervision of Dr. Driankov. The thesis defines the basic principles, techniques, and implementation issues underlying the design of a general purpose ‘ontological controller’. The general purpose ontological controller will be applied to the new DEGO III control unit for marine diesel engines.

23.10 Additional information

This summary has been brief due to the obvious space limitations. An extended version of this chapter gives a more detailed picture, additional references in particular to the work in sections 23.6-23.9, and references to related work outside our group. It is available by remote ftp on ftp.ida.liu.se as /pub/labs/rkllab/general/annrep95.ps.gz, or by WWW via http://www.ida.liu.se/labs/rkllab/index2.html.

23.11 References

Bäckström, C. [1] *Computational Complexity of Reasoning about Plans*. Doctoral dissertation, Linköping Univ., Linköping, Sweden, Jun., 1992.

Bäckström, C. [2] Equivalence and tractability results for SAS^+ planning. In Bill Swartout and Bernhard Nebel, eds., *Proc. of the 3rd Int. Conf. on Principles on Knowledge Representation and Reasoning (KR-92)*, pp. 126–137, Cambridge, MA, USA, Oct., 1992. Morgan Kaufmann.

Bäckström, C. [3] Finding least constrained plans and optimal parallel executions is harder than we thought. In Christer Bäckström and Erik Sandewall, eds., *Current Trends in AI Planning: EWSP’93—2nd European Workshop on Planning*, pp. 46–59, Vadstena, Sweden, Dec., 1993. IOS Press.

Bäckström, C. [4] Executing parallel plans faster by adding actions. In Anthony G. Cohn, ed., *Proc. of the 11th European Conf. on Artificial Intelligence (ECAI-94)*, pp. 615–619, Amsterdam, The Netherlands, Aug., 1994. Wiley.

Bäckström, C. [5] Expressive equivalence of planning formalisms. *Artificial Intelligence, Special Issue on Planning and Scheduling*. To appear.

Bäckström, C. [6] Planning using transformation between equivalent formalisms: A case study of efficiency. In David Wilkins, ed., *Comparative Analysis of AI Planning Systems*, 1994. Held in conjunction with AAAI-94 [40].

Bäckström, C., Jonsson, P. [7]. Planning with abstraction hierarchies can be exponentially less efficient. Accepted to *the 14th Int. Joint Conf. on Artificial Intelligence (IJCAI’95)*, Montreal, Canada, Aug., 1995.

Bäckström, C., Klein, I. [8]. Planning in polynomial time. In G Gottlob and W Nejdl, eds., *Expert Systems in Engineering: Principles and Applications. Int. Workshop*. In LNAI, Vol. 462, pp. 103–118. Springer Verlag, 1990.

Bäckström, C., Klein, I. [9]. Parallel non-binary planning in polynomial time. In Reiter and Mylopoulos [38], pp. 268–273.

Bäckström, C., Klein, I. [10]. Planning in polynomial time: The SAS-PUBS class. *Computational Intelligence*, 7(3):181–197, Aug., 1991.

Bäckström, C., Nebel, B. [11]. On the computational complexity of planning and story understanding. In Bernd Neumann, ed., *Proc. of the 10th European Conf. on Artificial Intelligence (ECAI-92)*, pp. 349–353, Vienna, Austria, Aug., 1992. Wiley.

Bäckström, C., Nebel, B. [12] Complexity results for SAS^+ planning. In Ruzena Bajcsy, ed., *Proc. of the 13th Int. Joint Conf. on Artificial Intelligence (IJCAI-93)*, Chambéry, France, Aug.–Sept., 1993, Morgan Kaufmann, pp. 1430-1435.

Bäckström, C., Nebel, B. [13]. Complexity results for SAS^+ planning. *Computational Intelligence*. To appear.

Doherty., P. [14] A constraint-based approach to proof procedures for multi-valued logics. In *Proc. 1st World Conf. on the Fundamentals of Artificial Intelligence*, Paris, 1-5 July, 1991. WOCFAI-91.

Doherty., P. [15]. *NML3 - A Non-Monotonic Formalism with Explicit Defaults*. Ph. D. thesis, Linköping Univ., Sweden, 1991.

Doherty., P. [16]. Notes on PMON circumscription. Technical Report LiTH-IDA-94-43, Department of Computer and Information Science, Linköping Univ., Linköping, Sweden, Dec., 1994.

Doherty., P. [17] Reasoning about action and change using occlusion. In *Proc. of the 11th European Conf. on Artificial Intelligence,* Aug. 8-12, Amsterdam, pp. 401–405, 1994.

Doherty., P., Lukaszewicz, W., Szalas, A. [18] Computing circumscription revisited: A reduction algorithm. Technical Report LiTH-IDA-94-42, Department of Computer and Information Science, Linköping Univ., Linköping, Sweden, Dec., 1994.

Doherty., P., Lukaszewicz, W., Szalas, A. [19] Circumscribing features and fluents. In D. Gabbay and H. J. Ohlbach, eds., *Proc. of the 1st Int. Conf. on Temporal Logic*, Vol. 827 of LNAI, pp. 82–100, Springer, 1994.

Jonsson, P., Bäckström, C. [20] Tractable plan existence does not imply tractable plan generation. Submitted to IJCAI'95.

Jonsson, P., Bäckström, C. [21] Complexity results for state-variable planning under mixed syntactical and structural restrictions. In Philippe Jorrand, ed., *Proc. of the 6th Int. Conf. on Artificial Intelligence: Methodology, Systems, Applications (AIMSA-94)*, Sofia, Bulgaria, Sept. 1994. World Scientific Publishing.

Jonsson, P., Bäckström, C. [22] Tractable planning with state variables by exploiting structural restrictions. In *AAAI-94*, pp. 998-1003, [40].

Klein, I., Jonsson, P., Bäckström, C. [23] Tractable planning for an assembly line. Research Report LiTH-ISY-R-1717, Department of Electrical Engineering, Linköping Univ., Linköping, Sweden, Jan. 1995. Submitted to *IJCAI'95*.

Lukaszewicz, W., Madalinska-Bugaj, E. [24] Program verification techniques as a tool for reasoning about actions and change. In *Proc. of the 18th German Conf. on Artificial Intelligence,* Vol. 861 of LNAI, pp. 226–236, Springer-Verlag, 1994.

Lukaszewicz, W., Madalinska-Bugaj, E. [25] Reasoning about actions and change using Dijkstra's semantics for programming languages. Submitted to *IJCAI-95*, 1995.

Nadjm-Tehrani, S. [26] *Reactive Systems in Physical Environments: Compositional Modelling and Framework for Verification*. Dissertation No. 338, Dept. of Computer and Information Science, Linköping Univ., 1994. ISBN 91-7871-237-8.

Nadjm-Tehrani, S., Strömberg, J-E. [27] From Physical modelling to Compositional models of Hybrid Systems. *Proc. of the 3rd. Int. Conf. on Formal Techniques in Real-time and Fault-tolerant Systems*. LNCS, No. 863, pp. 583–604, Springer-Verlag, 1994.

Nebel, B., Bäckström, C. [28] On the computational complexity of temporal projection and plan validation. In *AAAI-92* [39], pp. 748–753.

Nebel, B., Bäckström, C. [29] On the computational complexity of temporal projection, planning and plan validation. *Artificial Intelligence*, 66(1):125–160, 1994. ARTINT 1063.

Sandewall, E., Rönnquist, R. [30] A representation of action structures. In *Proc. of the 5th (US) National Conf. on Artificial Intelligence (AAAI-86)*, pp. 89–97, Philadelphia, PA, USA, Aug., 1986. American Association for Artificial Intelligence, Morgan Kaufmann.

Sandewall, E. [31] The range of applicability of nonmonotonic logics for the inertia problem. In *Int. Joint Conf. on Artificial Intelligence*, 1993.

Sandewall, E. [32] *Features and Fluents. The Representation of Knowledge about Dynamical Systems. Volume I*. Oxford Univ. Press, 1994.

Sandewall, E. [33] The range of applicability of some nonmonotonic logics for strict inertia. *Journal of Logic and Computation*, 4(5), Oct., 1994.

Sandewall, E. Shoham, Y. [34] Nonmonotonic temporal reasoning. In Dov Gabbay, ed., *Handbook of Artificial Intelligence and Logic Programming*, Oxford Univ. Press, 1994.

Söderman, U. [35] *On Conceptual Modelling of Mode Switching Systems*. Licentiate thesis, Linköping Univ., 1995.

Szalas, A. [36] On an automated translation of modal proof rules into formulas of the classical logic. *Journal of Applied Non-classical Logics*, 4:119–128, 1994.

Yi, C.-H. [37] Towards the assessment of logics for concurrent actions. In *AAAI 1995 Spring Symposium*, 1995.

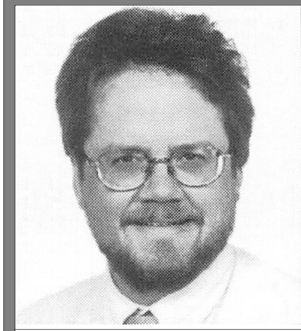
Reiter, R., Mylopoulos, J. [38] eds. *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI-91)*, Sydney, Australia, Aug., 1991, Morgan Kaufmann.

[39] American Association for Artificial Intelligence. *Proc. of the 10th (US) National Conf. on Artificial Intelligence (AAAI-92)*, San Jose, CA, USA, Jul., 1992,

[40] American Association for Artificial Intelligence. *Proc. of the 12th (US) National Conf. on Artificial Intelligence (AAAI-94)*, Seattle, WA, USA, Jul. – Aug., 1994.

RTSLAB

24 The Laboratory for Real Time Systems



Anders Törne
director

Tools, Method and Architecture for RT System Design

Applications in Automation and Embedded Systems

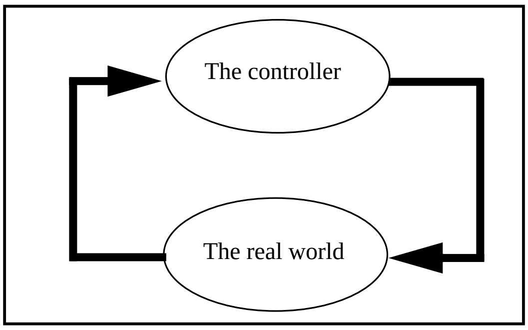
Some of the members of the Laboratory for Real Time Systems.
Esa Falkenroth, Peter Loborg, Anne Eskilsson, Erik Carlsson, Staffan Bonnier, Anders Törne, Stefan Kalmelid.

24.1 Summary

The laboratory for Real Time Systems (RTSLAB) was formally started in December 1994 by dividing EDSLAB into two different groups. The research within the laboratory is focused on aspects of the embedded software component of real-time systems. As this topic cannot be dealt with in isolation there is also interdisciplinary work in the laboratory between computer science and engineering, i.e. mechanical, electrotechnical, or chemical process engineering. The general aim here is to apply techniques in computer science to the engineering design process of real-time systems, and to use the obtained experience for initiating and inspiring new research within computer science.

24.1.1 General

As is generally acknowledged, a real-time system may be considered divided into two parts: A *real world* process (also called a plant) and a *controller*, that contains embedded software. The two parts are interconnected via sensor feedback and actuating devices. The purpose of the controller is to ensure that the real world process exhibits a desired behaviour. This is accomplished by sensing the process, and by carrying out appropriate manipulative actions. The transformation of sensory information to correct actions is today heavily dependent on software, and the software is more and more becoming the means for enabling advanced functionality in the total system.



Many of the traditional engineering sciences, like automatic control and control theory, contribute with methods and theories for how to sense and manipulate the physical environment (i.e. the real world) for it to behave appropriately. However, it must be realized that the high complexity of the software in interaction with other parts of the total system creates new demands on the engineering of such systems - in this respect computer science research has a significant role to establish and to migrate methods into an industrial reality. The laboratory is aimed at providing such methods and tools and to test their adequacy.

The general aim of the research in RTSLAB is twofold:

- To develop appropriate architectures, representations and methods to support engineers in the development, design and verification of the controller.
- To develop tools and methods to supervise and monitor executing real time system software.

24.1.2 Method

The laboratory work is based on computer science methods and fetches inspiration both from formal approaches and from implementation. The general method of research within RTSLAB may be considered divided into two levels: The *application* level and the *generalisation* level.

The application level comprises applicative projects in cooperation with industrial partners, where developed methods and tools are tested in case studies. The experience from these application projects is fed back into the generalization level activities, where general models and theories are developed.

24.1.3 Laboratory members

Laboratory leadership and technical/administrative staff:

Anders Törne, Ph. D., associate professor, laboratory director
Anne Eskilsson, secretary

Research staff:

Staffan Bonnier, Ph. D.
Esa Falkenroth, M. Sc.
Peter Loborg, M. Sc., lic.
Stefan Kalmelid, M. Sc.

Associated researchers and supervisors:

Zebo Peng, Ph. D., CADLAB, IDA, associated researcher
Tore Risch, Prof., Engineering Databases, EDSLAB, IDA

24.2 Projects

This section presents 4 main projects, in which RTSLAB is participating. Some of them are cooperation projects with industry and other departments. The general goal of each project is presented.

24.2.1 CENIIT

The laboratory research is supported by the technical faculty and by the local faculty organization CENIIT (Centre for Industrial Information Technology). This provides basic funding for the graduate students and the research supervision within the laboratory. The CENIIT project therefore works as an umbrella project for the laboratory research.

24.2.2 Embedded Real Time Systems

This project is funded by NUTEK and is entitled *Synthesis and Realization of Extendable Embedded Real-time Systems*. The project is aimed at combining the research on task level programming (see below) and executive, centred around an object relational database, AMOS, with formal description methods for synthesis and design of real-time control systems. The AMOS database has been developed at EDSLAB, IDA.

The work is based on earlier work on the instruction and control of automation equipment - the ARAMIS project. In ARAMIS, a primary memory database has been used to store the process model data and the control information. As a database, the earlier system is incomplete and the AMOS framework is perfectly suited for the purpose.

Presently the work is divided into two subprojects - *RT-Database for Distributed, Concurrent Software Control* and *Models and Tools for Synthesis of Embedded RT-Systems*. In the first subproject the necessary techniques will be developed, in cooperation with the AMOS-project at EDSLAB, and the resulting real time database will be integrated into the ARAMIS system and/or tested in another suitable application. In the second subproject formal techniques and a principle for software architecture for task level control will be developed.

24.2.3 Complex Systems

This is a collaboration project involving the following partners:

from LiTH, Linköping University - Dept. of Mechanical Engineering (IKP), Dept. of Electrical Engineering (ISY) and Dept. of Computer and Information Science (IDA);

from industry - Saab Scania AB/Saab Military Aircraft, Volvo Aero Corporation and VOAC Hydraulics AB.

The total project is funded by NUTEK within the programme “Complex Systems”.

The work within this project is concerned with tools and methods for the development process of complex, heterogeneous systems. It targets modelling for simulation/verification, distributed simulation, formal representations of discrete and/or continuous models, database technology and computer based tools for specification, synthesis and (semi-)automatic verification. The basic idea is to develop and get experience from tools applied to realistic industrial examples of complex heterogeneous systems. From this an adequate methodology will be developed which integrates formal analysis and simulation in the early design process, thus enhancing the quality/reliability of the final designs before implementation/manufacturing.

By a heterogeneous system we mean a system composed of, e.g., mechanical, electrical, and /or hydraulic components, in interaction with a software-based control system. We are considering systems like vehicles, aircrafts, large machinery etc. The software often performs controlling operations, even if a large part is purely supervisory in character.

The focus is on the system development phase. In this phase the early, main design decisions are taken. In our framework, these decisions are based on the analysis/simulation of abstract models of the design, depending on the function and situation of system usage. The design of the software is an important part, but is viewed in integration with the design of the hardware and the other subsystems with which it interacts.

24.2.4 Sensors and Control for Autonomous Assembly

This project is a cooperation project with two other departments at LiTH – the Dept. for Mechanical Engineering (IKP) and the Dept. for Physics and Measurement Technology (IFM). The aim of this project is to deepen the knowledge within sensor controlled robots in automatic assembly and to improve dynamic control and supervision functionality by developing general sensor and control models. One of the major tasks is to integrate sensor systems from IFM with TROCS (transputer based robot control system) at IKP and with the ARAMIS 'task control system' at IDA. The project has mainly been concerned with two aspects where the participation from RTSLAB is within the second:

- The integration of sensors into TROCS (IFM/IKP). So far, a basic communication protocol for an ultrasonic sensor has been developed. The basic principle is to distribute a TROCS process to the Sun workstation connected to the sensor. The protocol is general and can with different parameters be adapted to different kinds of sensors within this project.
- Interfacing of ARAMIS with the research platform at IFM. So far, a prototype implementation of the control level of ARAMIS has been developed at IFM, and equipment at IFM can now be manoeuvred through ARAMIS. Currently, sensor models are under development to represent the vision sensor at IFM.

Besides the inter-department cooperation, we are negotiating with one of the development groups at ABB Robotics in Västerås. Possible cooperation is the integration of our sensors and interfaces to the ABB robots under the S4 control system.

24.3 Current activity review

Here we make a short presentation of the current status of different activities related to individual graduate students and researchers. An activity is a coherent part of the laboratory research work, and is coordinated/supervised by Dr. Anders Törne.

24.3.1 Tools and Methods for Real Time System Design

Staffan Bonnier

This activity was started in December 1994. The aim is to develop analysis methods and tools to aid in the design and modelling of safety-critical real time systems. The general goal for the methods is for them to be viable in practical applications while remaining formally well-founded, thus aiding the designer in spotting inadequacies of a design or verifying the absence thereof. There are two main directions of research within the activity:

- Safety analysis
- Type checking

Safety analysis deals with the static detection of unsafe states of the designed system. This may be done by checking all states for unsafeness which are reachable from the initial state. However, such a method is often not adequate due to the large number of reachable states in industrially sized systems. In this activity we investigate alternative approaches to safety analysis along two different lines: (1) User assisted backward search; Starting from an unsafe state one attempts to find and eliminate the “minimal” source of the unsafeness (2) Abstract interpretation; The designed system may be “approximated” by a smaller system in which certain information has been ignored. Reachability analysis may then be applied to the smaller system, resulting in a sufficient condition for safeness.

By “type checking” we mean the detection of inconsistencies between user declared (redundant) types and the actual system design. This part of the activity aims at identifying appropriate notions of types in real time system design, and to develop type checking algorithms to support the designer in detecting shallow but severe flaws such as “missing cases” and “inappropriate connections”.

Modified Timed Petri Nets (MTPN) are used as a general target for methods developed in this activity. The work is financed from two of the projects; *Embedded Real Time Systems* and *Complex Systems*. CENIIT and other faculty funding for this activity is also acknowledged.

24.3.2 Control and Error Recovery in a Multi-machine Environment

Peter Loborg

The goal for this activity is to develop a programming and run-time environment for multi-machine manufacturing environments, including cooperating manufacturing cells. A forward-chaining rule based approach has been taken for the description of manufacturing tasks. The language combines *procedural abstraction* and *rules*. 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 data model - the world model.

Each object in the world model is represented as an *augmented finite state machine* where control algorithms are associated with each transition and state. When a state transition occurs in the Aramis world model (a primitive request in the task level language), the control algorithm is executed in the real-time control system and success or failure is acknowledged to the ARAMIS run-time system. Algorithms can also be associated with object states when they must be actively maintained (non-stable states).

The explicitly represented model of each object enables additional functionality in the overall system. As an example, a proposal for how to assist the operator at system restart (after failure and subsequent repair in, e.g. an FMS cell) is presented in a licentiate thesis by Peter Loborg. Currently, a simulator is under development to generate control and state data on a scale not available in our research platforms. The purpose is to develop the algorithms needed to achieve restart from an almost arbitrary state and to evaluate the proposal.

This activity is financed by NUTEK via the project *Sensors and Control for Autonomous Assembly*. CENIIT and other faculty funding for this activity is also acknowledged.

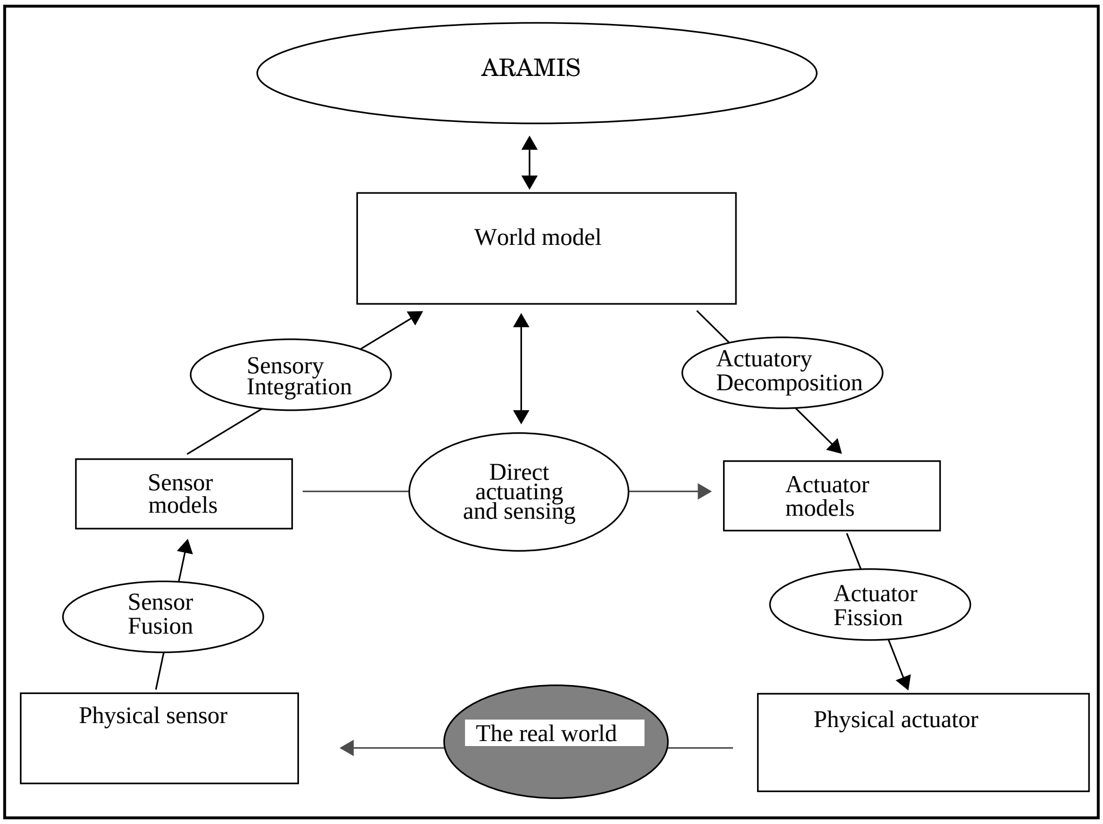


Figure 24-1 *A model of the dataflow and the different subprocesses when connecting ARAMIS to an executing environment*

24.3.3 Databases in Control and Simulation

Esa Falkenroth

This activity studies the use of database technology for control and simulation modelling. Currently, ad hoc solutions are often used in process control for handling concurrency control and complex operations on large sets of data. The resulting systems tend to be inflexible and data management problems arise when they grow in size and complexity. The work is based on an active object-relational database, AMOS, developed at EDSLAB, IDA.

More specifically, the integration of control applications with active object-relational databases and activity modelling languages aims at a general approach to modelling, data management, and communication issues, using a hierarchical architecture with a database in the middle.

Furthermore, the integration of a simulator in the architecture will enable partial on-line simulation of the controlled system, where some parts are simulated. It will also provide support for a mixture of discrete and continuous simulation. The basic idea has been developed in the ARAMIS project. Given this base it is natural to continue with connecting simulation and discrete model development tools to the database.

The usage of a database for model storage and run-time support for mode changes will have the following benefits:

- it will be possible to pose ad hoc queries about model state during the simulation, for example when the simulation is in real time and continues for long time.
- it provides a common data model, i.e. analysis tools and model generators can easily be changed
- it will naturally be integrated with distributed simulation of different subsystems and provide the interface between these

This activity is financed via the subproject *RT-Database for Distributed, Concurrent Software Control* in the project *Embedded Real Time Systems*, but is also funded via the project *Complex Systems*. CENIIT and other faculty funding for this activity is also acknowledged.

24.3.4 Petri Net Modelling

Stefan Kalmelid, Zebo Peng

The approach within this activity is to use Modified Timed Petri Nets (MTPN) to describe real-time processes, in particular manufacturing processes. The general idea is to partition the system into one controlling (embedded) system and one controlled system. This is similar to practice in automatic control and discrete event systems. The controlling system is described by a Petri net with timed places and guarded transitions, where the guards represent states of the controlled system. The controlled system is described by a set of finite state automata with timed transitions, where each transition is guarded by a place of the controlling system. Non-determinism of the automata enables modelling of a non-deterministic environment.

The MTPN approach supports hierachical decomposition by allowing places of the controlling system and transitions of the controlled system to be further subdivided. Furthermore, global timing analysis as well as pre-runtime conditional scheduling is supported.

The focus of the work is on developing formally based modelling tools for an abstract specification of system behaviour and to perform case studies involving real scale industrial application.

This activity is financed via the subproject “*Models and Tools for Synthesis of Embedded RT-Systems”* in the project “Embedded Real Time Systems” and the applications are to be made within the “Complex Systems” project. CENIIT and other faculty funding for this activity is also acknowledged.

24.3.5 Tool Integration in Complex Systems Design

vacant

Complex designs necessarily means large amounts of information. In this activity we limit this to design information for heterogeneous systems, in particular specification/ modelling tool output, formal models for analysis and simulation models, functional and non-functional specifications, test case specifications, results (possibly intermediary) from analysis and simulation. We envision a common database as a design information repository.

An object oriented database will, e.g., support multiple access, hierarchical decomposition, object oriented system modelling, ad hoc querying over the design and a common data model for design information representation. Local user databases will hold data in update. The common data model is instantiated for a specific set of tools used.

The base technology for this is presently developed at EDSLAB in a basic research project, AMOS. In the AMOS architecture the mediator concept provides a way to interface a plethora of different application tools - the so called domain mediators. The main issues to be addressed in this project are the common data model – generation and representation, and the tool integration (the mediators) to the common data model. The mediators may be generated from a description of the representation used by the tool.

This activity is financed within the *Complex Systems* project.

24.4 Other activities

A graduate course on “Real Time Systems” was carried out during spring 1994. The course is intended as a master course (i.e., a basic course for the graduate students) within the graduate school at IDA. Staffan Bonnier participated in the planning of a graduate course - “Formal Methods in Programming Languages”.

Peter Loborg took his licentiate exam in june 1994. The thesis was titled “Error Recovery Support in Manufacturing Control Systems”.

The laboratory is participating in the formation of a competence centre ISIS - “Integrated Control and Information Systems”, lead by Prof. Lennart Ljung, Automatic Control, ISY, Linköping University. The focus of the competence centre can be exemplified by the keywords: Control Theory, Logic and Discrete Mathematics, Real-time Databases, Real-time Programming and Control Systems. The centre is connected to several of the largest swedish industries within this area.

The laboratory has a close network of contacts with other departments at the university - several of the projects are inter-disciplinary and involve participation from Physics and Measurement Technology, Electrical Engineering and Mechanical Engineering.

There are also frequent contacts with industries in the aerospace, process control and robotics area.

24.5 Relevant publications

Peng, Z., Törne, A. Timing Analysis and Conditional Scheduling in a Real-Time System Design Environment, in *Proc. of the 6th Euromicro Workshop on Real Time Systems (IEEE)*, Västerås, Sweden, Jun., 1994.

Loborg, P. *Error Recovery Support in Manufacturing Systems*, Linköping Studies in Science and Technology, Lic. Thesis No. 440, ISBN 91-7871-376-5.

Ekman, A., Strömberg, D., Törne, A. Exploration of Polygon Environments using Range Data. Accepted for publication in *IEEE Transactions on Systems, Man and Cybernetics.*

Ekman, A., Törne, A., Strömberg, D. A Robot Control Formalism Based on an Information Quality Concept, in *Proc. of AIAA/NASA Conf. on Intelligent Robots for Factory, Field, Service, and Space*, Houston, Texas, Mar., 1994.

Loborg, P., Törne, A. (1994). Manufacturing control system principles supporting error recovery. In *Working notes of the AAAI 94 Spring Symp.* Series: Detecting and Resolving Errors in Manufacturing Systems, Stanford, CA, USA.

Loborg, P. (1993). Error recovery - an overview. In "*LiTH Robotikdagar, Automatiserad tillverkning från högteknologi till tillämpning*", Linköping, Sweden. Later version presented at *AAAI 94 Spring Symp. on Detecting and Resolving Errors in Manufacturing Systems*, Stanford, CA, USA, Spring 1994.

Loborg, P., Risch, T., Sköld, M., Törne. A. Active OO Databases in Control Applications, in Microprocessing and Microprogramming, In *Proc. of the 19th Euromicro Conf.*, Vol. 38, No. 1-5, pp. 255-264, Barcelona, Sep., 1993.

Loborg, P., Holmbom, P., Sköld, M., Törne, A. Programming Flexible Manufacturing Systems. In *Proc. of Workshop on Adaptive Manufacturing Equipments*, Nutek 1992:53, Oct., 1992, CAELAB Memo 92-02.

Loborg, P., Sköld, M., Törne, A., Holmbom, P. A Model for the Execution of Task Level Specifications for Intelligent and Flexible Manufacturing Systems. In *Proc. of the Vth Int. Symp. on Artificial Intelligence, ISAI92*, Cancun, Mexico, Dec., 1992. LiTH-IDA-R-92-22.

Loborg, A., Törne, A. A Hybrid Language for the Control of Multimachine Environments. In *Proc. of EIA/AIE-91*, Hawaii, Jun., 1991. LiTH-IDA-R-91-09.

Loborg, P., Sköld, M., Törne, A. A Hierarchical Software Architecture for Control of Industrial Robots and Manufacturing Equipment. In *Proc. of 1st National Symp. on Real-Time Systems*, in Tech.Rep. No. 30, Dept. of Computer Systems, Uppsala Univ., Jun., 1991. CAELAB Memo 92-01.

Peng, Z. Törne, A. A Petri Net Based Modelling and Synthesis Technique for Real-Time Systems. In *Proc. of the 5th Euromicro Workshop on Real Time Systems*, Oulo, Finland, Jun., 1993.

Törne, A. The Instruction and Control of Multi-Machine Environments. In *Applications of Artificial Intelligence in Engineering V, Vol. 2, Proc. of the 5th Int. Conf.*, Boston Jul. 1990, Springer-Verlag and presented at the *AAAI 90 workshop on intelligent diagnostic and control systems for manufacturing*, Boston, Jul., 1990. LiTH-IDA-90-07.

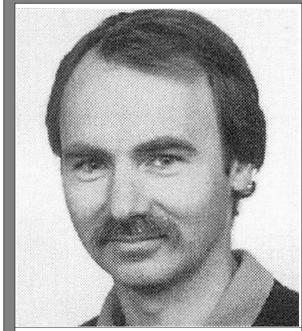
Törne, A. Computer Assistance in Automation, Report IDA/CENIIT, LiTH-IDA-R-89-51, Linköping Univ., 1989.

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

VITS

25 Information Systems and Work Contexts

Systems development as organizational change analysis
Business process redesign information requirements analysis
Information systems architecture
Evaluation of information systems
CASE tools and metamodelling of ISD methods



Göran Goldkuhl
director

Some of the members of the group for Informations Systems and Work Context.
Back row: Göran Goldkuhl, Fredrik Öberg, Stefan Cronholm, Klas Gäre, Lena Lindeberg, Marie-Therese Lundmark.
Front row: Mikael Lind, Annie Röstlinger, Karin Axelsson, Malin Bergvall, Torbjörn Näslund.

25.1 Introduction

VITS was formed as a research group within the Dept. of Computer & Information Science, Linköping University in 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. We are studying strategies, methods and tools for information systems development (ISD).

Some important aspects of our research orientation:

- collaboration with practice
- methods development
- empirical studies, mainly qualitative methods used

25.2 Overview of current research

The research is performed in the following projects:

- Change analysis
- Business Process Redesign through Change Analysis
- Contextual and activity based evaluation of information systems
- Strategies for information systems architecture
- Inter-organizational information systems
- CASE tools and method support - situational analysis and development
- CASE supported systems development and user participation

25.2.1 Change analysis

The first phase of ISD is often called feasibility study, pre study, survey or something like that. This phase involves an investigation into whether the proposed system is considered feasible. Both in method and in practice it seems that IS development is often taken for granted when performing feasibility studies.

When an analysis activity (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: The development of an information system must be seen as an organizational change. As such it is one 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 decisions 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 SIMM. The method consists of different method components, such as:

- activity analysis
- problem analysis
- strength analysis
- goal analysis
- determination of change measures

The research is now directed towards problem and strength analysis; particularly investigating empirical experiences from practical use of the method and its theoretical grounding. We are also developing the method towards an explicit business process approach; this is described in the next section.

This research is partly funded by the Faculty of Arts and Sciences, Linköping University.

Researchers: Annie Röstlinger, Göran Goldkuhl

25.2.2 Business Process Redesign through Change Analysis

The topic for this research project is business development with business processes and increased time efficiency being the focus. Through the concept Business Process Redesign (BPR), the interest is in restructuring organizations keeping business processes in focus:

- see the customer all the way through the business, where one looks into the organization horizontally
- gain cost, time, volume efficiency
- increase the quality of the products/services

The organization is divided into business processes, where a process (a coherent series of activities) is defined by relation to the customer in combination with internal activities, which the organization has to execute to fulfill its commitments.

Change analysis/SIMM focuses on activities and flows. In this research project, CA/SIMM is further developed towards an expanded methodological concept: Business Process oriented Change Analysis. At present this methodological concept is being tested in an action research project at Structo AB (a medium sized Swedish steel corporation).

Researchers: Mikael Lind, Göran Goldkuhl

25.2.3 Contextual evaluation of information systems

Evaluation of information systems can be performed in different ways. One can study pure technical aspects of information systems. One can also study issues of information quality and information content. 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 develop a methodology for contextual evaluation of information systems. We have limited 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 of information systems

The development of methodology is based on our earlier methods for change analysis and activity analysis. We aim at simple and powerful analysis and description techniques to be used by non-experts.

The development of the evaluation methodology has been performed in close relation to action research projects in different municipalities. In these projects VITS researchers participated together with IS users evaluating information systems in a contextual way.

This research is funded by The Swedish Association of Local Authorities.

Researcher: Annie Röstlinger

25.2.4 Activity based evaluation of information systems

This is a research project, just starting up, which is closely related to the earlier mentioned one.

Today there is an intensive debate concerning the economic effects of investments in information systems. The reason for this debate is that despite the big investments that companies and organizations have made in information systems, there is very little evidence that these investments have been profitable. A possible way to evaluate the economic effects of information technology in organizations is to combine methods and theories from economic science and information system science. The research idea is to combine the theories and methods of information systems (”Activity based systems development”) and the economic methods of ABC/ABM (Activity Based Costing/Management). The purpose is to develop a method to be used in evaluating the economic effects of information systems in organizations.

Financial support from NUTEK, The Swedish National Board for Industrial and Technical Development (ITYP Program).

Researcher: Owen Eriksson

25.2.5 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 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 not dependent on organizational structure. The structure of information determines the structure of IS. Another approach is functional structuring of 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 have been studying and comparing these two strategies (IRM and VBS) for information systems architecture. We have performed six empirical case studies of organizations using one of these strategies (three of each). We have studied IS functionality, IS change, IS transparency and IS responsibility. After these case studies we are now embarking on methods development for IS structuring; i.e. the design and redesign of information systems architecture. We have just started an action research project at the National Maritime Administration (Sjöfartsverket).

This research is funded by The Swedish Council for research in the Humanities and Social Sciences and The Swedish Work Environment Fund.

Researchers: Karin Axelsson, Göran Goldkuhl

25.2.6 Inter-organizational information systems

A project we are cooperating with the Swedish National Audit Bureau concerning governmental inter-organizational information systems. Earlier studies have shown that there are severe problems in this area, like deficiencies in information quality, unclear responsibility and roles between different governmental authorities, both in development, use and maintenance. This part of the project is aiming at development of a better conceptual and descriptional model for governmental inter-organizational information systems.

We have also an interest in methods for developing inter-organizational information systems. We are just starting a collaboration with a Swedish consultancy firm (FRONTEC AB). Together we are going to develop an IOIS development method based on their and our earlier ISD methods.

Researchers: Bengt Andersson, Lena Lindeberg

25.2.7 CASE tools and method support - situational analysis and development (CATMANDO)

This project studies ISD methods, CASE tools and the method support embedded in such tools. We are developing and applying meta modelling methods in order to describe, analyse and (re)design methods and tools. This metamethod is called method analysis/SIMM (MA/SIMM). This metamethod consists of method components for 1) situational analysis of method/tool usage and 2) ideal-typical method modelling, as analysis of 2a) method perspective and objectives, 2b) work procedure, 2c) concepts and 2d) notations of the method. Combined with this we are developing a component-based framework for ISD methods.

We have performed qualitative empirical studies on method and tool usages in different Swedish organizations.

One important part of this research is a collaboration project with ABB Infosystems. Together we are developing a new ISD method architecture with method components. This cooperation project is called “From business processes to software components”. The new ISD methods will have a clear business process orientation. Developing information systems should be part of a business process redesign. One shall also apply a software component approach; i.e. IS development with and for reuse of software components.

The research is supported by The Swedish National Board for Industrial and Technical Development (NUTEK) and ABB Infosystems.

Researchers: Tommy Wedlund, Fredrik Öberg, Göran Goldkuhl, Dan Fristedt, Kenneth Åhlgren

25.2.8 CASE supported systems development and user participation (CASA)

This is a sister-project to CATMANDO. We have in these two projects performed an integrated study of CASE tool usage and ISD methods in several Swedish companies. It has been a qualitative empirical investigation.

We are interested in improving the active utilization of CASE tools in ISD projects. Different tool properties are investigated and we are developing design objectives for an ideal tool concept called “flexible method tool”. Based on this tool concept we are developing a tool for our method SIMM. At the moment this tool consists of modules for activity analysis, problem analysis, strength, analysis and goal analysis.

This research is supported by The Swedish Work Environment Fund.

Researchers: Stefan Cronholm, Fredrik Öberg.

25.3 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 in interesting several persons in other organizations to cooperate in research and participate in graduate studies related to our research.

Group leadership and administrative and technical staff:

Göran Goldkuhl, Ph. D., associate professor
Lena Wigh, secretary
Christer Fahlgren, programmer
Jörgen Malmsten, technical writer

Employed graduate students:

Karin Axelsson, Lic.
Stefan Cronholm, Lic.
Lena Lindeberg, B. Sc.
Annie Röstlinger, B. Sc.
Tommy Wedlund, B. Sc.
Fredrik Öberg, B. Sc.

External graduate students participating in the group´s research:

Bengt Andersson, Swedish National Audit Bureau
Malin Bergvall, University College Östersund
Owen Eriksson, University College Falun/Borlänge
Dan Fristedt, ABB Corporate Research
Mikael Lind, University College Borås
Kenneth Åhlgren, University College Örebro

Other external graduate students in Information Systems Development:

Anders Avdic, University College Örebro
Anna Brolin, Gaia System AB
Benneth Christiansson, University College Borås
Klas Gäre, SYSteam AB
Patrik Hedberg, University College Borås
Boris Karlsson, University College Borås
Roger Lindquist, University College Växjö
Marie-Therese Lundmark, University College Borås
Carita Åbom, Jönköping Interrnational Business School

25.4 Publications

Book:

Goldkuhl, G. (1993). *Verksamhetsutveckla datasystem*. Intention, Linköping.

Licentiate Theses:

Cronholm, S. (1994). *Varför CASE-verktyg i systemutveckling? En motiv- och konsekvensstudie avseende arbetssätt och arbetsformer.*

Eriksson, O. (1994). *Informationssystem med verksamhetskvalitet - utvärdering baserat på verksamhetsinriktat och samskapande synsätt.*

Axelsson, K. (1994). *Informationssystemstrukturering, ansvarsfördelning och användarinflytande - en komparativ studie med utgångspunkt i två informationssystemstrategier.*

Papers to international research conferences:

Cronholm, S., Fristedt, D. (1993). CASE shells - flexible and open environments for customizing CASE tools. In *Proc. of 16th Information Systems Research Seminar in Scandinavia.* Copenhagen, Denmark.

Cronholm, S., Goldkuhl, G. (1994). Meanings and motives of method customizations in CASE environments - observations and categorizations from an empirical study, in Theodoulides B (Ed, 1994) *Proc. of the 5th workshop on the next generation of CASE tools.* Universiteit Twente, Twente, The Netherlands.

Goldkuhl, G., Röstlinger, A. (1993). Joint elicitation of problems: An important aspect of change analysis, in Avison D et al. (1993, Eds) *Human, organizational and social dimensions of Information systems development*. North-Holland/IFIP w.g. 8.2.

Pettersson, K. (1993). Consequences of an IRM-based system development, in *Proc. of 16th Information Systems Research Seminar in Scandinavia*. Copenhagen University, Copenhagen, Denmark.

Papers to national research conferences:

Eriksson, O. (1993). Att beteckna och benämna i samband med utformning av informationssystem, *VITS Höstseminarium.* Linköpings universitet, 1993.

Goldkuhl, G. (1993). Välgrundad metodutveckling, på *VITS Höstseminarium.* Linköpings universitet, 1993.

Goldkuhl, G., Fristedt, D. (1994). Metodanalys - en beskrivning av metametoden SIMM, *VITS Höstseminarium.* Linköpings universitet, 1994.

Pettersson, K., Goldkuhl, G. (1993). IRM på byggföretag - en fallstudie om strukturering av informationssystem, *VITS Höstseminarium*. Linköpings universitet, 1993.

Röstlinger, A. (1993). Styrkeanalys. Ett arbetssätt för att tillvarata positiva aspekter i verksamheten, *VITS Höstseminarium.* Linköpings universitet, 1993.

Röstlinger, A., Brolin, A. (1993). Verksamhetsinriktad utvärdering av datasystem på ett Miljö- & hälsoskyddskontor - uppläggning, genomförande och erfarenheter från en tillämpning, *VITS Höstseminarium.* Linköpings universitet, 1993.

Papers to practice conferences:

Axelsson, K. (1994). Informationssystemstrategiers betydelse vid förändring av informationssystem. *Sundsvall42*. Sundsvall, Sweden.

Eriksson, O. (1994). Aktivitetsbaserad utvärdering av informationssystem och verksamhet, *Sundsvall42*. Sundsvall, Sweden.

Goldkuhl, G. (1994). Några problem vid datadriven strukturering av informationssystem, *LISS 94*. Linköpings universitet. Linköping, Sweden.

Goldkuhl, G., Cronholm, S. (1993). Customizable CASE environmnets: A framework for design and evaluation, *COPE' IT' 93/NordDATA*. Copenhagen, Denmark.

Axelsson, K. (1994). Några problem vid verksamhetsbaserad strukturering av informationssystem, *LISS 94*. Linköpings universitet., Linköping, Sweden.

Pettersson, K., Goldkuhl, G. (1993). A comparison between two strategies for information systems architectures, *Sundsvall42*. Sundsvall, Sweden.

Röstlinger, A. (1993). Verksamhetsinriktad värdering av datasystem inför förändring - metod och erfarenhet, *Sundsvall42*. Sundsvall, Sweden.

Röstlinger, A. (1994). Att ta tillvara styrkan i gjorda IT-investeringar, *Sundsvall42*. Sundsvall, Sweden.

Other papers:

Goldkuhl, G., Pettersson, K., Eriksson, O. (1993). Hur studera realisering och konsekvenser av strategibaserade informationssystemarkitekturer?

Pettersson, K. (1993). IRM på försäkringsbolag - en fallstudie om strukturering av informationssystem, IDA, Linköpings universitet. Linköping, Sweden.

Pettersson, K. (1993). VBS i industriföretag - en fallstudie om strukturering av informationssystem, IDA Linköpings universitet, Linköping, Sweden.

Pettersson, K., Goldkuhl, G. (1993). VBS på pappersbruk - en fallstudie om strukturering av informationssystem, IDA Linköpings universitet. Linköping, Sweden.

Axelsson, K., Eriksson, O. (1994). VBS på skogsbolag - en fallstudie om strukturering av informationssystem, IDA Linköpings universitet, Linköping, Sweden.

Axelsson, K., Röstlinger, A., Brolin, A. (1994). Datadriven utveckling i kommun - en fallstudie om strukturering av informationssystem.

Röstlinger, A., Pettersson, K. (1993). Utvärdering och förändring av datorstödda kommunala verksamheter - rapport från en förstudie bland kommuner i Östergötland.

Appendix A
Department Organization

A.1 Organization

The Department was formed in 1983, bringing together subjects previously in the Mathematics and the Electrical Engineering departments. In September 1990 another subject, 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 fourteen.

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 directors of undergraduate studies (studierektorer) are responsible for the courses and the teachers in its subject area. Most teachers are shared between the lab group and one (or sometimes more) teaching group.

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.

A.2 Economics

The department budget for the fiscal year 1994/95 balances at 82,7 MSEK. (One SEK is at present approximately 0.13 USD.) The resources for undergraduate education supplied by the university amount to 26,2 MSEK, and corresponding resources for research and graduate education are 19,9 MSEK. The research activities are heavily dependent on external sources, where *Swedish National Board for Industrial Technical Development*, NUTEK, is the main contributor (94/95 17 MSEK). Another main contributor is the *Swedish Research Council for Engineering Sciences*, TFR, (94/95 6,4 MSEK). Additional funds are provided by the *Swedish Council for Research in the Humanities and Social Sciences*, HSFR (94/95 1 MSEK). We also have external funds through co-operation with external accounting firms (94/95 2,0 MSEK). Totally our external funds amount to 35,5 MSEK.

	92/93	**93/94**	**94/95**
Total budget, MSEK	65	74	83
University	38	41	47
Undergraduate education	22,8	23,0	26,2
Research and graduate education	15,3	16,1	18,3
Graduate school	-	1,6	1,6
External funding	27	35	36
NUTEK (incl EG-projects)	11,9	16,1	17,0
TFR	4,6	6,0	6,4
HSFR	1,4	1,5	1,0
FRN	1,5	0,9	-
Accounting firms	2,6	2,4	2,0
Continued education	0,2	1,3	1,5
Industry, defence research institute, other Research councils etc.	4,8	6,8	8,1

Figure A-1. *IDA's budget for the last three years.*

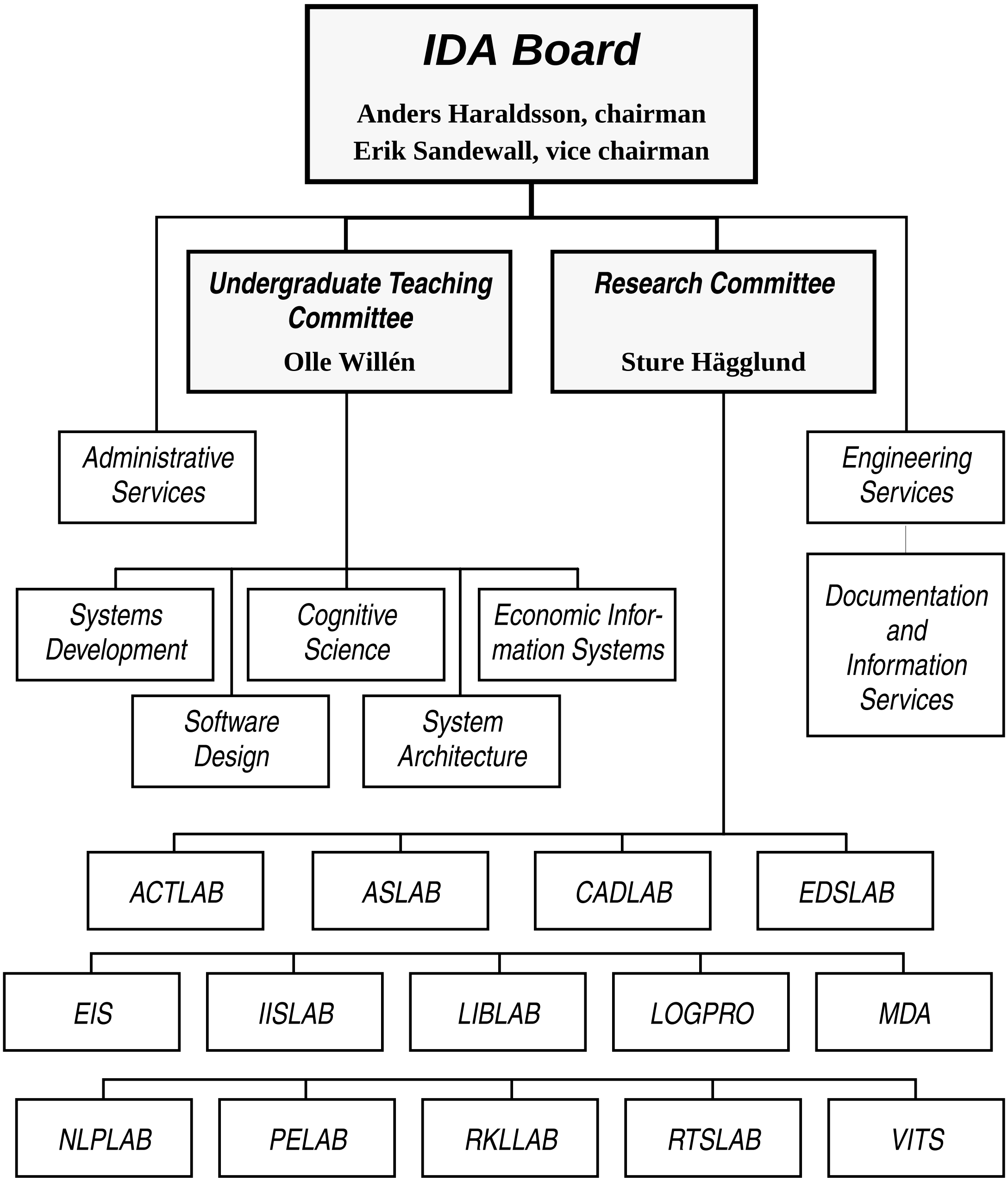
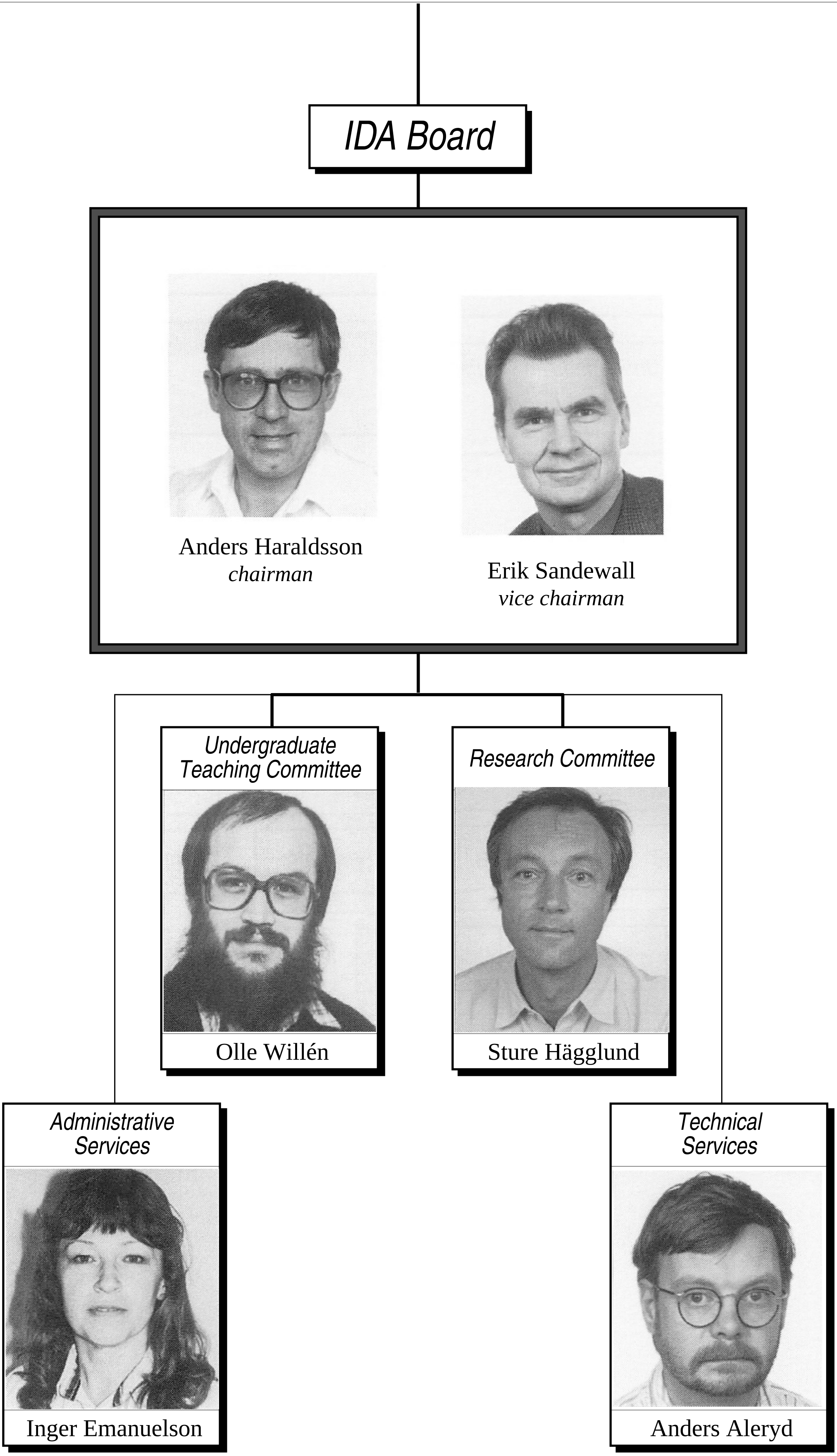


Figure A-2. *Organization of the department.*


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.3 The Department Board

The Department Board is chaired by Anders Haraldsson, with Anne Eskilsson 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.

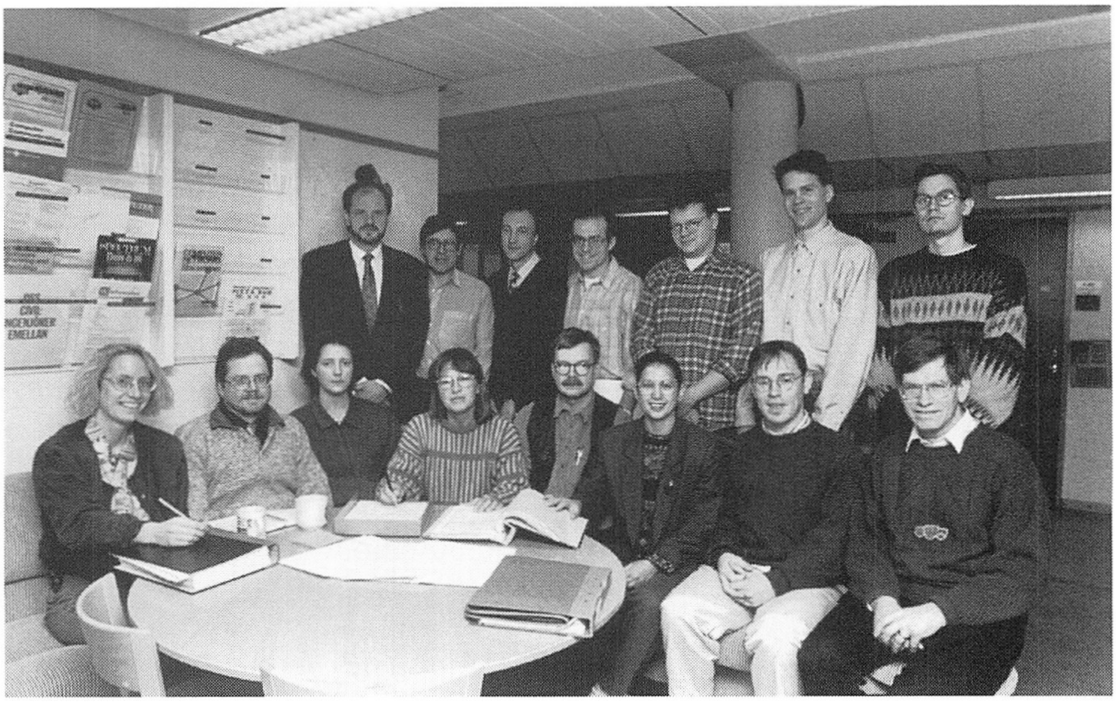
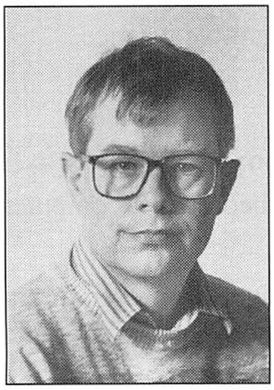


Figure A-3. *The IDA Board. .*

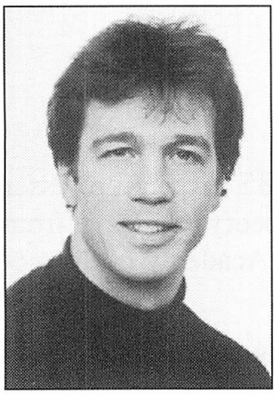
Appendix B
Graduate Study Programme

B.1 Faculty engaged in graduate study program



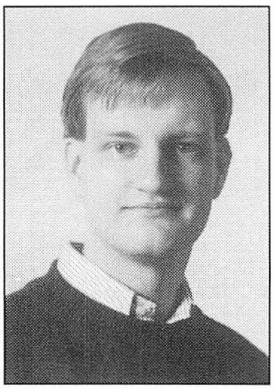
Lars Ahrenberg, Ph. D., Uppsala 1987. Professor of computational linguistics. Group leader, NLPLAB. On partial leave 1994-95.

Syntax, semantics and pragmatics of natural language; natural language understanding; natural language interfaces; machine-aided translation.



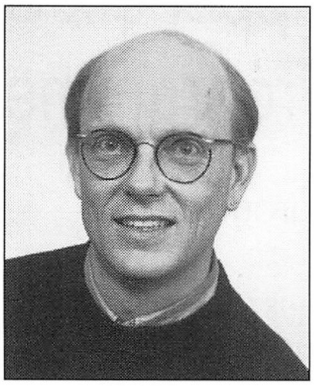
Staffan Bonnier, Ph. D., Linköping 1992. Research associate (*forskarassistent)*. Previous affiliation for Carlstedt Elektronik AB, Göteborg 1993-1994.

Formal methods in real-time system development, theory and practice of declarative programming.



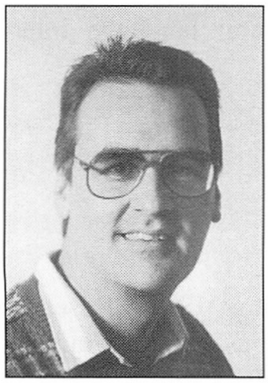
Christer Bäckström, Ph. D., Linköpig 1992. Assistant professor *(universitetslektor*).

Planning and temporal reasoning, algorithms and complexity for AI problems, representation and reasoning about knowledge.



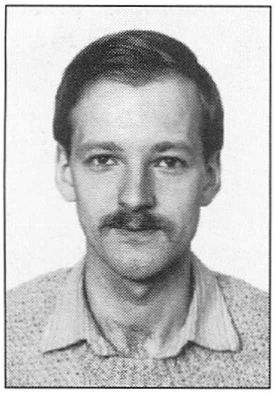
Nils Dahlbäck, Ph. D., Linköping 1992. Acting professor, (*tf professor*) in Computational Linguistics (94/95). Assistant professor in Cognitive Science. Director of undergraduate studies in cognitive sciences.

Natural language processing, especially empirically based computational models of discourse. Cognitive aspects of discourse coherence in man and machine. User modeling and intelligent interfaces. Philosophy of mind and its consequences for empirical theories in cognitive science.



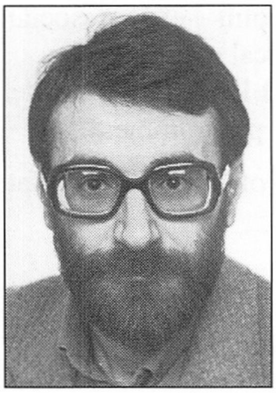
Patrick Doherty, Ph. D., Linköping 1991. Assistant professor (*universitetslektor*), logic and theoretical computer science.

Logical approaches to knowledge representation; reasoning with incomplete information, non-monotonic reasoning, reasoning about action and change.



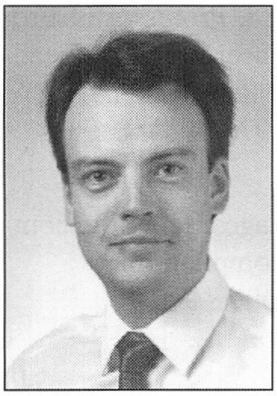
Wlodek Drabent, Ph. D., Warsaw 1985. Acting professor (*docent*, *tf professor*), programming theory. On leave from Institute of Computer Science, Polish Academy of Sciences.

Logic programming, programming languages semantics.



Dimiter Driankov, Ph. D., Linköping 1989. Assistant professor *(universitetslektor),* logic and AI.

Reasoning under uncertainty, many-valued logics, approximate reasoning, fuzzy control & systems, autonomous agents.

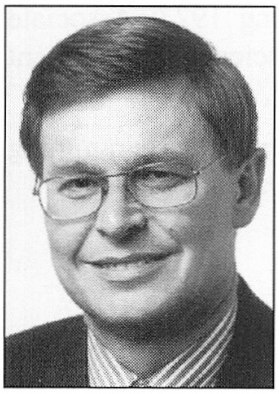


Henrik Eriksson, Ph. D., Linköping 1991. Assistant professor (docent, *vik. universitetslektor*), informatics. Previous affiliation Stanford University, Stanford, California 1991–94.

Knowledge-based systems, knowledge acquisition, software development environments, software reuse.

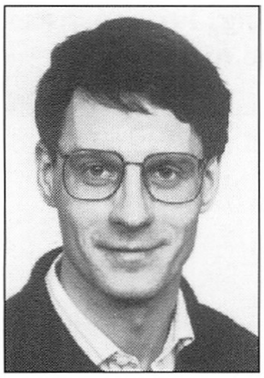
Johan Fagerström, Ph. D., Linköping 1988. Assistant professor (*universitetslektor*), computer science. Director of undergraduate studies in System Architecture.

Distributed systems, object-oriented programming, object-oriented analysis and design, operating systems.



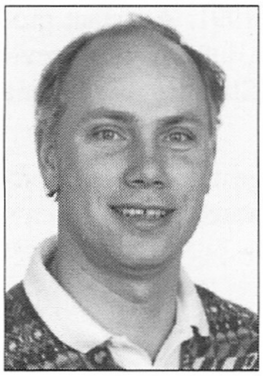
Thomas Falk, Econ. Dr., Stockholm 1976. Adjunct professor in economics of information technology. Chalmers University of Technology. Several previous affiliations.

Management of information technology.



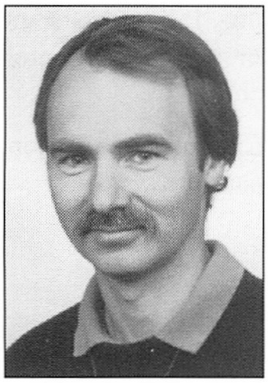
Per-Olof Fjällström, Ph. D., Stockholm 1985. Associate professor *(universitetslektor)*, theoretical computer science. Previous affiliation KTH and IBM. Group leader, ACTLAB. Director of graduate study programme.

Computational geometry, analysis of algorithms, data structures.



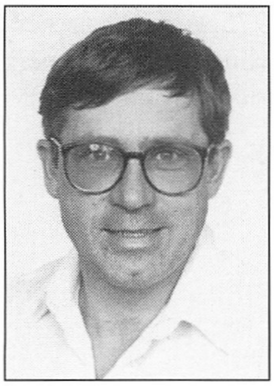
Peter Fritzson, Ph. D., Linköping 1984. Professor of Programming Systems and Software Engineering. Group leader, PELAB. Previous affiliations Sun MicroSystems, USA, 1985-86, Linköpings Univ. Hospital 1976-79, LiU Physics dep., 1975-76.

Programming environments and languages, scientific computing, debugging tools, incremental compilation technology, compiler generation, compilers and development tools for parallel hardware.



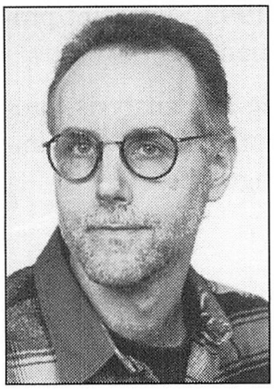
Göran Goldkuhl, Ph. D., Stockholm 1980. Associate professor (*docent, universitetslektor*), administrative data processing. Group leader, VITS. Research director at Centre for studies on Man, Technology and Organization. Previous affiliations Stockholm, Göteborg.

Theories/methods on problem formulation, business process and activity analysis, IS design and evaluation; ISD methods and customization of CASE tools; Humanistic science traditions and qualitative research methods.



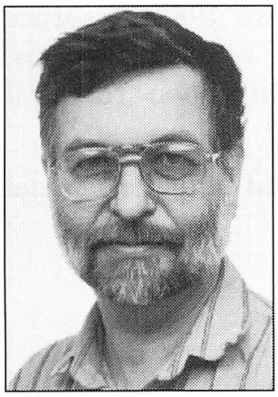
Anders Haraldsson, Ph. D., Linköping 1977. Associate professor (*bitr professor*), computer science. Department chairman. Previous affiliation Uppsala.

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



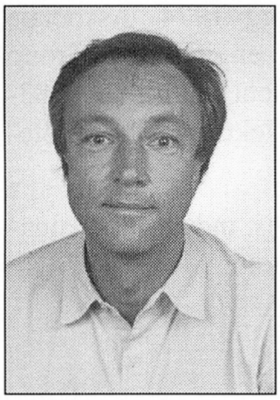
Richard Hirsch, Ph. D., Gothenburg 1989. Associate professor *(docent, universitetsadjunkt*) of general and computational linguistics. Thesis supervision in the laboratory for natural language processing. Previous affiliation Gothenburg.

Syntax, semantics, and pragmatics of natural languages; discourse analysis; argumentation theory; semiotics; philosophy of language.



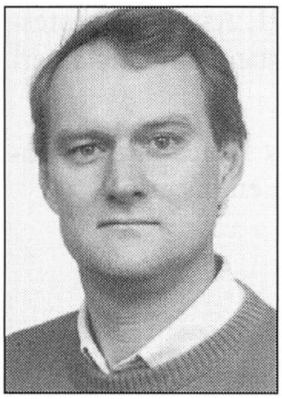
Roland Hjerppe, Universitetsadjunkt. Group leader, LIBLAB. Previous affiliation KTH, DFI and expert mission Tanzania. Visiting Distinguished Scholar at Office of Research, OCLC Inc. in Columbus, Ohio, 1988-89.

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.



Arne Jönsson, Ph. D., Linköping 1993. Assistant professor (*universitetslektor),* computer science. Director of undergraduate studies in cognitive sciences. On leave 1994-95 to Monash University, Clayton, Australia.

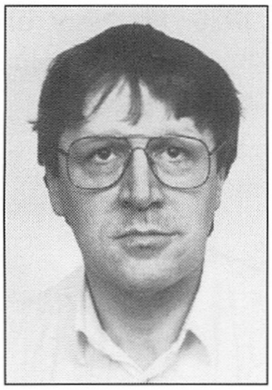
Artificial intelligence, Natural language processing, especially empirically based computational models of human-computer dialogues.

Mariam Kamkar, Ph. D., Linköping 1993. Assistant professor (*universitetslektor)*, computer science.

Software maintenance, software testing and analysis, program debugging, program analysis, optimization in compilers, multiparadigm programming languages.

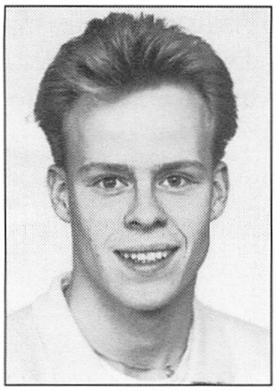
Krzysztof Kuchcinski, Ph. D., Gdansk 1984. Associate professor (*docent*, *universitetslektor*), computer systems. Group leader, CADLAB. Previous affiliation Technical Univ. of Gdansk, Poland.

Computer architecture, computer-aided design of digital systems, VLSI, design for testability.



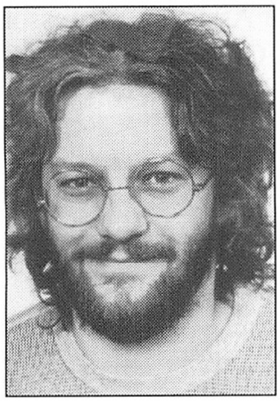
Bengt Lennartsson, Ph. D., Göteborg 1974. Associate professor (*universitetslektor*), software engineering. Group leader, PELAB 1981-88 and department chairman 1983-90. Previous affiliation for Carlstedt Elektronik AB, Göteborg 1992-94.

Software engineering, real-time systems, industrial evaluation of elements of new software technology.



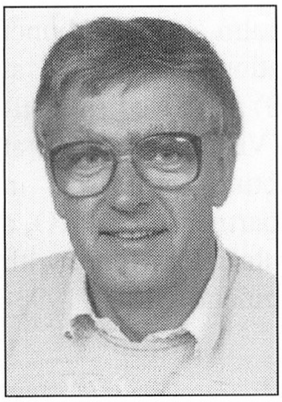
Jonas Löwgren, Ph. D., Linköping 1991. Associate professor *(docent*, *universitetslektor*), of human-computer interaction.

Human-Computer Interaction, usability issues in information systems development and software engineering, usability-oriented design support.



Jacek Malec, Ph. D., Wroclaw 1987. Research associate (*forskarassistent*). Previous affiliation Technical Univ. of Wroclaw, Poland.

Artificial intelligence, knowledge representation, reactive systems, autonomous systems, system theory.

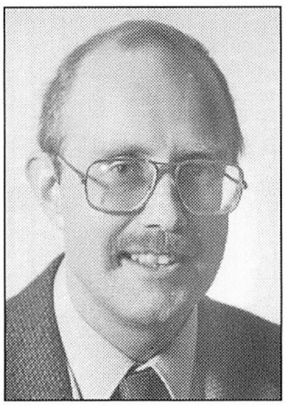


Jan Maluszynski, Ph. D., Warsaw 1973. Professor of programming theory. Several previous affiliations. Group leader, LOGPRO. On leave 1994-95 INRIA, France.

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

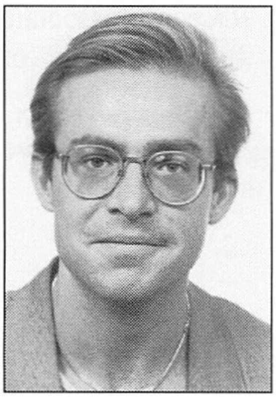
Simin Nadjm-Tehrani, Ph. D. Linköping 1994. Assistant professor *(vik. universitetslektor)* computer science.

Modelling and verification of embedded systems, temporal logic, real-time systems, logic programming.



Anders G. Nilsson, Econ. Dr., Stockholm 1991. Affiliated 94/95. Position at the Institute for Business Process Development (Institute V) and Stockholm School of Economics. Previous affiliations Royal Institute of Technology and Stockholm University.

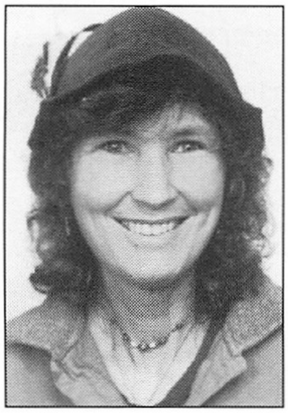
Business modelling, strategy planning, activity based development, information systems development, maintenance management, application packages, information management.



Ulf Nilsson, Ph. D., Linköping 1992. Assistant professor (*universitetslektor*), computer science. Group leader, LOGPRO. Previous affiliation State University of New York at StonyBrook, USA.

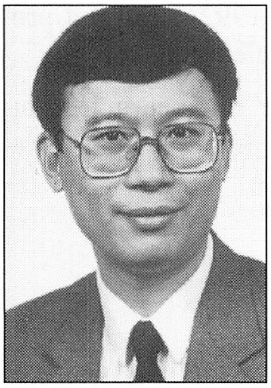
Logic programming and deductive databases; Evaluation strategies for query processing; program transformation and abstract interpretation.

Nils-Göran Olve, Econ. Dr., Stockholm 1977, Adjunct professor in economics of information technology at Stockholm School of Economics (HHS) and Swedish Institute of Management (IFL) from 1981. Visiting professor at EIASM in Brussels 1984-85. Acting professor in Linköping 1993. Since 1986 a senior partner in CEPRO, a management consultancy in Stockholm. Part-time with Swedish Institute of Systems Development (SISU) 1993-95.



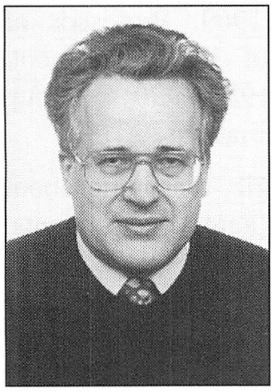
Lin Padgham, Ph. D., Linköping 1989. Associate professor (*docent, universitetslektor),* computer science. Previous affiliation Univ. of Oregon, USA, and Tektronix. Group leader, IISLAB. On leave 1993-95 to the University of Melbourne, Australia.

Inheritance, default reasoning, taxonomical reasoning, object-oriented systems.



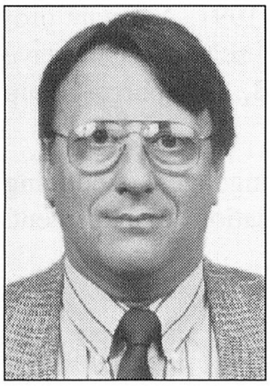
Zebo Peng, Ph. D., Linköping 1987. Associate professor *(docent, universitetslektor)*, computer systems.

Automated synthesis of digital systems, formal description of hardware, hardware/software co-design, design for testability, VLSI, computer architecture.



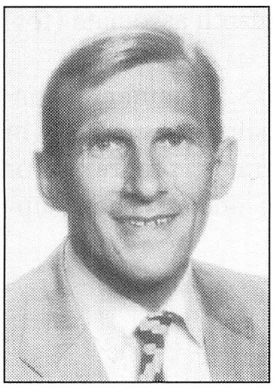
Birger Rapp, Econ. Dr., Stockholm 1974. Professor of economic information systems. Vice president at large for IFORS. Editorial (advisory) boards to EJOR, IJMSD, JORBEL and Omega. President of the Pronova Research and Development Board in Sweden.

Accounting, business control, agency theory, IT and organization, production, economics.

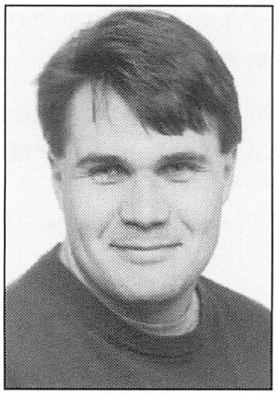


Tore Risch, Ph. D., Uppsala 1978. Professor of engineering databases. Group leader, EDSLAB. Previously at Uppsala University, IBM Almaden Research Lab. (San Jose, CA), Stanford Research Institute, Syntelligence Inc. (Sunnyvale, CA), HP Laboratories (Palo Alto, CA), and Stanford University.

Database support for engineering and scientific applications, e.g., object-oriented databases, heterogeneous data bases, active databases, and real-time databases.

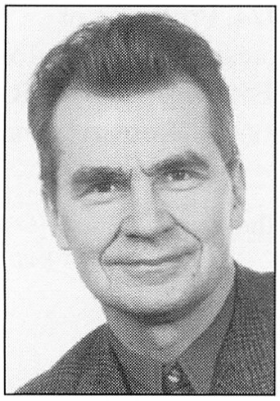


Rolf Rundfelt, Ph. D., Stockholm 1974. Adjunct professor in economic informations systems, especially in Swedish and International External Accounting. Docent in Business Administration, University of Stockholm. Since 1966 lecturer at the University of Stockholm.



Kristian Sandahl, Ph. D., Linköping 1992. Assistant professor (*universitetslektor*), computer science.

Knowledge Management, Knowledge Engineering, Industrial Software Engineering, Quality Improvement Paradigm, Experimental Research Methods.

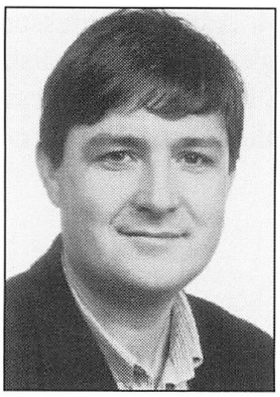


Erik Sandewall, Ph. D., Uppsala 1969. Professor of computer science. Vice-Chairman of the department. Research Committee Chairman (1983-1990). Group leader, RKLLAB. Several previous affiliations.

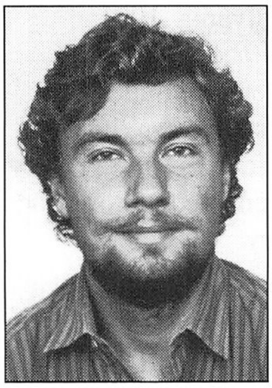
Representation of knowledge with logic, reasoning about actions and change, cognitive robotics, autonomous agents.

Nahid Shahmehri, Ph. D., Linköping 1991. Assistant professor (*universitetslektor*), computer science. On leave 1993-94 for Carlstedt Elektronik AB, Göteborg. Group leader, IISLAB.

Programming theory, programming languages, debugging tools, compiling technology, information management, business process modelling, CSCW.

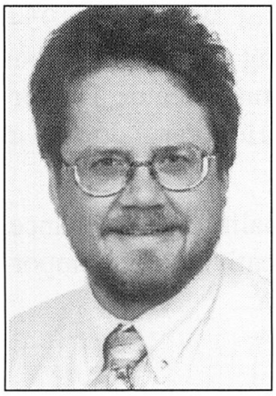


Åke Sivertun, Ph. D., Umeå 1993. Research associate (forskarassistent) at LIBLAB.
Geographical Information Systems - GIS. Communication of complex data and linking multi disciplinary models in GIS. Research in environmental programs, programs for medical geography, physical planning and decision support.



Toomas Timpka, MD., Stockholm 1983, Ph. D., Linköping 1989. Associate professor (*docent*, *universitetslektor*), computer and information science. Group leader, MDA. Acting Professor of Social Medicine and Associate Professor of Medical Informatics.

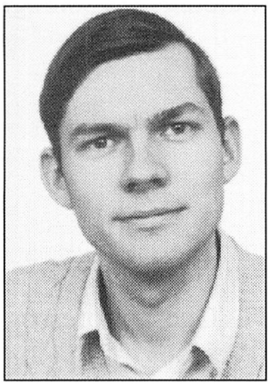
Hypermedia, computers and society, human-computer interaction, systems development.



Anders Törne Ph. D., Uppsala 1980. Associate professor (*universitetslektor)*, computer support in automation. Group leader, RTSLAB. Corporate Research, Västerås. CENIIT manager from 1992.

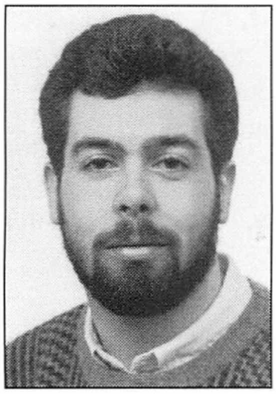
Tools, method and architecture for real-time system design. Applications in automation and embedded systems. Real time programming and specification languages. Robot programming.

B.2 Guest researchers engaged in graduate study program

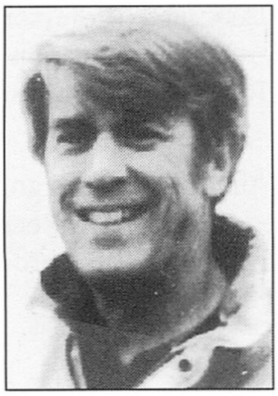


Roland Bol, Ph. D., Amsterdam 1991. Guest researcher. Previous affiliation CWI, Amsterdam.

Logic programming, non-monotonic reasoning, deductive databases, process algebra.

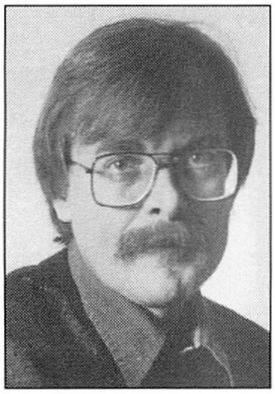


Brant Cheikes, Ph.D., Univ. of Pennsylvania, 1991.
Guest researcher.
Natural language processing and cooperative dialogue; architectures for response-planning systems, esp. intelligent help systems; simulator-based training systems. Applications of knowledge-based systems technology for operator training in the process industry.



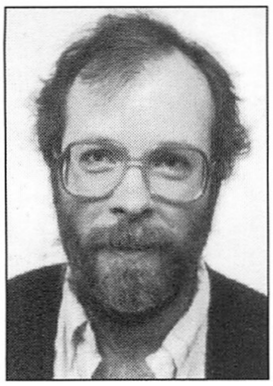
Robert L. Glass, M. Sc., University of Wisconsin 1954. Guest professor. President, Computing Trends. Previous affiliation Software Engineering Institute, Seattle University, The Boeing Company, USA. Honorary doctor at Linköping University 1995.

Software engineering: software quality, maintenance, design, technical communication, creativity. The importance of the application domain.



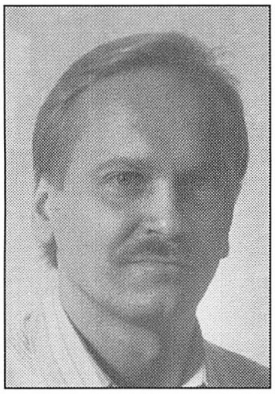
Witold Lukaszewicz, Ph. D., Warsaw University 1979. Guest professor. On leave from the Institute of Informatics, Warsaw University, Poland.

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.

Barbara Wasson, Ph.D. Saskatchewan 1990. Professor of pedagogical information systems (University of Bergen). Previous Affiliations Applied Cognitive Science (University of Toronto), Norwegian Telecom Research, NORUT Information Technology. Visiting professor 1994-1997.

Artificial intelligence in education and training, instructional planning, collaberative learning and intelligent learning environments.

B.3 Graduate Study Course Program 1993-94

Basic and Occasional Graduate Courses:

VLSI System Design (Krzysztof Kuchcinski, Zebo Peng)
Distributed Databases (Tore Risch)
Parallel Programming and Compilation Techniques (Peter Fritzson)
Generalized Knowledge Representation Systems (Lars Ahrenberg)
Non-Monotonic Reasoning (Witold Lukaszewicz)
Machine Learning (Nada Lavrac)
Introduction to Research Methodology in Computer Science (Sture Hägglund)
Documents: Subject Analysis and Description (Roland Hjerppe)
Cognitive Science (Nils Dahlbäck, Yvonne Waern)
Ekonomisk styrning (Nils-Göran Olve)
Kunskapsutvecklingteori, projektering, genomförande (Göran Goldkuhl)
Metodik för verksamhetsanalys & informationsanalys (Göran Goldkuhl)
Vetenskaplig dialog om informationssystem (Göran Goldkuhl)
Real-Time Systems (Anders Törne)
Principles of Modern Databases (Tore Risch)
Principles of Automated Action Planning (Christer Bäckström)
Applied Rewriting (Jan Maluszynski)
Natural Language Semantics (Richard Hirsch)
Architectures of Intelligent Autonomous Agents (Jacek Malec)
Compiler Construction – Advanced Course (Mariam Kamkar)
Formal Description Techniques for Distributed and Communicating Systems (Piotr Dembinski)
Information Needs and Uses (Roland Hjerppe)

B.4 Graduate Study Course Program 1994-95

Basic and Occasional Graduate Courses:

Aspects of Scientific Writing (Ulf Nilsson)
Formal Methods in Programming Languages (Anders Haraldsson)
Företagsteori (fortsättningskurs) (Birger Rapp)
Företagsvärdering (Birger Rapp)
Information och verksamhet (Göran Goldkuhl)
Introduction to Research Methodology in Computer Science (Sture Hägglund)
Klassisk informationssystemsteori (Anders G Nilsson)
Knowledge organization (Roland Hjerppe)
Kunskapsutveckling – teori, projektering, genomförande (Göran Goldkuhl)
Logic for Ida-ites (Patrick Doherty)
Real-time Databases (Tore Risch)
Rhetorical Theory and the Philosophy of Language (Richard Hirsch)
Systems Modelling (Jacek Malec)
Testing-Problems and Techniques (Mariam Kamkar)
Topics in Software Engineering (Sture Hägglund)
Utredningsmetodik och kvantitativa metoder (Birger Rapp)
Approximate reasoning and fuzzy control (Dimiter Driankov)
Business Process Redesign (Göran Goldkuhl)
Documents: Formal Description and Architectures (Roland Hjerppe)
Functional Programming (Peter Fritzson)
Heuristic Algorithms for Combinatorial Optimization Problems (Zebo Peng)
Human-Computer Interaction: Traditions and Trends (Jonas Löwgren)
Object-oriented system development (Johan Fagerström)
Parallel Algorithms for Regular Architectures (Per-Olof Fjällström)
Participatory Design (Toomas Timpka)
Programming Languages (Wlodek Drabent)
Situated Robotics (Jacek Malec)
Temporal Constraint Reasoning (Christer Bäckström)

B.5 Seminars 1993-1994

Seminars later half of study year 1992-93

Feb.	9	Yvonne Waern, Tema K, Linköping Univ. – *Humanistisk informationsteknologi*
	25	Magnus Sterky, EuroSim AB, Nyköping, Sweden – *Experiences from operator training by means of simulators*
March	4	Conrad Wolfram – *Mathematica – Ett språk och programpaket*
	17	Kai Koskimies, Univ. of Helsinki, Finland – *An Object-Oriented Language Implementation Model*
	22	Erland Jungert, FOA – *Spatial Information Systems, seminarieserie*
	23	Björn Nilsson, SISU – *Presentation av forskning vid SISU – Svenska Institutet för Systemutveckling*
	26	Nimal Jayaratna, Heriot-Watt Univ., Scotland – *Do Methodologies Work in Practice?*
April	5	Miguel Nussbaum, Catholic Univ. of Chile – *Strategic Approximation Methods for Optimization Problems*
	15	Dan Shapiro, Lancaster Univ., UK – Using ethnography to inform system design for CSCW
	22	Bengt Andersson, FOA 1 – Användning av datormodeller inom krigsförloppsstudier i försvarets studie- och planeringsverksamhet
	27	Peter Struss, Univ. of Technology, Munich – Model-based Diagnosis – (Why) Does it Work?

28 Tingting Zhang, Linköping Univ., disputation – A Study in Diagnosis Using Classification and Defaults
29 Bogdan Korel, Wayne State Univ., Detroit -Automated Generation of Program Test Data
30 Mariam Kamkar, Linköping Univ., disputation – Interprocedural Dynamic Slicing with Applications to Debugging and Testing

May 4 Rolf Ohlsson, Gunnar Wadensten, Rationell Mikro Teknik AB – *Systemutveckling i praktiken*
5 Steven Reiss, Brown Univ., Providence, RI – Program Visualization: Where We Go From Here
5 Lars-Gunnar Larsson, Flygtekniska Försöksanstalten, Bromma – Artificiella förare i obemannad och bemannad simulering av luftstrid
5 Preston Ginsburg, Unisys Corp., Philadelphia, USA – Introducing HCI Design to Conventional Design Organizations Languages from Denotational Specifications
26 Andrew Gordon, Chalmers Univ. – An Operational Semantics for I/0 in a Lazy Functional Language
26 Mehran Noghabai, Linköping Univ., Lic. seminar – Värdering av strategiska datorinvesteringar
26 Mats Larsson, Linköping Univ., Lic. seminar – A Transformational Approach to Formal Digital System Design
27 Johan Ringström, Linköping Univ., Lic. seminar – Compiler Generation for Parallel Languages from Denotational Specifications

June 1 Jan Komorowski, The Norwegian Institute of Technology, Trondheim, Norway – *Towards A Refinement Calculus of Logic Programs*
3 Witold Litwin, Univ. Paris 9 – *Scalable Distributed Data Structures*
4 Viktor P Havin – *Mathematics as a Source of Certainty and Uncertainty*
4 Bjarne Däcker, Ellmentel Utvecklings AB – *Management of Technology Applied to Software*
7 Jonni Harrius, Linköping Univ., Lic. seminar – *An Architecture and a Knowledge Representation Model for Expert Critiquing System*
8 Göran Collste, Linköping Univ. – *Ethics in the Computer Society*
8 Michael Jansson, Linköping Univ., Lic. seminar – Propagation of Change in an Intelligent Information System
10 Oliviero Stock, Institute for scientific and technical research at Trento – *Error Detection* and Bidirectional Chart-parsing
10 Per Österling, Linköping Univ., lic.semianrium – Symbolic Modelling of the Dynamic Environments of Autonomous Agents
11 Arne Jönsson, Linköping Univ., disputation – Dialogue Management for Natural Language Interfaces – an Empirical Approach
14 Keith Hall, IBM European Networking Center – *Change Management for Cooperative Work*
23 Iris Vessey and Bob Glass, Univ. of Wisconsin – *Toward a Taxonomy of Software App*lication *Domains Contemprary Candidates Information System*

Seminars study year 1993-94

Sept. 6 Hideo Shimazu, Uppsala Univ. – *Multi-Modal Definite Clause Grammar*
7 Pascal van Hentenryck, Brown Univ. – *Constraint Logic Programming (talk 1)*
8 Pascal van Hentenryck, Brown Univ. – *A Tour of Abstract Inter*pretation Algorithms
9 Jim Sandkvist, Karl-Johan Raggl, SSPA Maritime Consulting AB – *Dator- och simulatorstöd i* övergripande lednings träning
20 Henri Muller, Gent Univ. – Logistics Games in Management
29 Börje Langefors, prof. emeritus i adm databehandling – *System och infologi*

Oct. 5 Gilberto File, Padova Univ. – *Basic Introduction and Theoretical Foundation of* Abstract Interpretation
5 Susan Chipman, U.S. Office of Naval Research – Cognitive Science in the U.S. – A View From ONR
8 Johan Boye, Linköping Univ., Lic. seminarium – Dependency-based Groundness Analysis of Functional Logic Programs
25 Anca-Juliana Stoica, Univ. of Bucharest – Högre seminarium i ekonomiska informations system: Some Problems of Industrial Systems Modelling and Control

Nov.	9	Nada Lavrac, the Jozef Stefan Institute in Ljubljana – *Inductive Concept Learning:* From Attribute-Value Learning to Inductive Logic Programming
	10	Berner Lindström, Göteborgs Univ. – Pedagogiska aspekter på utveckling och användning av simulering
	11	Jonas Barklund, Uppsala Univ. – Metaprogramming in Logic
	12	Lars Degerstedt, Linköping Univ., Lic. seminar – *Tabulated Resolution for Well-Founded* Semantics
	15	Peter Thomas, Brunel Univ., London – Understanding Computer Systems: what the user has to do
	15	Bogdan Hausman, Claes Wikström, Ellemtel – Det funktionella realtidsspråket Erlang
	22	Seif Haridi, SICS, Kista – DDM – Data Diffusion Machine
	22	Birgitta Hörberg, FOA 55, Karlstad – Simulering som utbildningsmetod
	29	Hans Zima, Univ. of Vienna – Compilation Techniques for Distributed Memory Machines
	29	Christine L. Borgman, Univ. of California – *Research and Development Issues in the* Design of Interfaces for Public Access Information Systems
	30	Erik Stolterman, Umeå Univ. – *Design Theory and Design Support Methods and Tools*
	30	Paul Over, Cray Research Inc., UK – *Bringing Supercomputer Technology to Massively* Parallel Processing
Dec.	2	Kurt Normark, Aalborg Univ., Denmark – *A Hypertext System for Program Development*
	6	Håkan Millroth, Uppsala Univ. – *Implicit vs. Explicit Control in Parallel Programming*
	7	Kurt Gestrelius, Lunds Univ. – *Simulering och utbildningsspel: upplevelsebaserad uppbygg*nad av erfarenhet
	13	Robert Schreiber, NASA/AMES Research Institute for Advanced Computer Science – Subway, a Communication Compiler for the Maspar MP-x Computers
	13	John R. Nickolls, MasPar Computer – Signal and Image Processing using Massively Parallel Computers
	16	Hubert Mille, Riksrevisionsverket – *Kvalitetsgranskning av ADB-utvecklingsprojekt*
	16	Anna Moberg, Linköping Univ., Lic. seminar – *En studie av kommunikationsmönster* vid arbete på distans

Jan.	7	Carl-Erik Måls of Teologic, Malmö – *the SDL Tool Set*
	11	John Williams, Kevin Amaratunga, MIT – *Research at the Intelligent Engineering Systems* Laboratory at MIT
	28	Takashi Gomi, M. Eng, P. Eng, Applied AI Systems, Inc., – Behavior Based AI and the New Robotics
	31	Peter Carlsson, Linköping Univ., Lic. seminar – Separation av företagsledning och finansiering – fallstudier av företagsledarutköp ur ett agentteoretiskt perspektiv
Feb.	25	Jukka Paaki, Univ. of Jyväskylä, – An Integrated Language For Protocol Engineering
	8	Camilla Sjöström, Linköping Univ., Lic. seminar – Revision och lagreglering – ett historiskt perspektiv
March	14	John Hughes, Chalmers Tekniska Högskola, Göteborg – *Sequential Algorithms and their* Applications
	22	Kenneth Zadeck, IBM Hawthorne research Center – *Optimizing Instruction Tags Using* Generated Transformations
April	7	Witold Litwin, Univ. Paris 9 – *A Family of Order Preserving Scalable Distributed Data* Structures
	15	Dag Folkesson, Saab Military Aircraft – *Principer för realtidsexekvering i svenska militära* avioniksystem
	20	Stig Nordqvist, KTH – Simulering som beslutsstöd i tidningsproduktion
	26	Bernard Lang, INRIA, France – Two Views of Tabulation in Parsing
	28	Anders Lundell, ABB Industrial Systems AB – ABB Master Distributed Control System
	28	Kent Nyström, ABB Industrial Systems AB – ABB MasterNet Communication
	28	Erik Gyllensvärd, ABB Industrial Systems AB – ABB Object Management Facility
May	3	Christian Balkenius – Mechanisms of Spatial Orientation: From Biology to AI
	5	Konrad Morgan, Univ. of Portsmouth – *Expanding Design Spaces and Diverse* User Populations in Human-Technology Interaction
	9	Anders Ravn, Technical Univ. of Denmark – *Correct Systems/Refinement of Real-Time* Specifications
	9	Simin Nadjm-Tehrani, Linköping Univ., disputation – Reactive Systems in Physical Environments: Compositional Modelling and Framework for Verification
	18	Elisabet Engdahl, Univ. of Edinburgh – A Framework for Computational Semantics
	19	Stellan Ohlsson, Pittsburgh Univ. – Datorstödd Patientutbildning
	25	Edwin Paalvast, Delft Univ. – Tools for Parallelization in Parallel Programming
	26	Philip Smets, Univ. of Bruxelles – *Belief functions for logical problems*
	27	Jerzy W. Rozenblit, Univ. of Arizona – *An Integrative Approach to Engineering* Design
	27	Cecilia Sjöberg, Linköping Univ., Lic. seminar – *Voices in Design: Argumentation in* Participatory Development
	31	Gert Smolka, DFKI, Germany – A Foundation for Concurrent Constraint Programming
June	1	Hans-Jürgen Eikmeyer, Univ. of Bielefeld – *Situated artificial communicator*
	9	Lars Viklund, Linköping Univ., Lic. seminar – *Contributions to a High-level Programming Environment for Scientific Computing*
	9	Lars Poignant, Linköping Univ., Lic. seminar – *Informationsteknologi och företags-etablering – effekter på produktivitet och region*
	10	Karin Pettersson, Linköping Univ., Lic. seminar – *Informationssystemstrukturering, ansvarsfördelning och användarinflytande*
	10	Owen Eriksson, Linköping Univ., Lic. seminar – *Informationssystem med verksamhets*-kvalitet – utvärdering baserat på ett verksamhetsinriktat och samskapande perspektiv
	13	Peter Loborg, Linköping Univ., Lic. seminar – Error Recovery Support in Manufacturing Control Systems
	14	Brant Cheikes, the MITRE Corp. Bedford, USA – *The Design of a Plan-Based Intelligent* Tutoring System for Analytic Problem Solving
	15	Edwin Paalvast, Delft Univ., The Nederlands – Tools for Parallelization in Parallel Programming
	15	Gustav Fahl, Linköping Univ., Lic. seminar – Object Views of Relational Data in Multi-database Systems
	22	Peter W. Eklund, Univ. of Adalaide, Australia – Applying Machine Learning to GIS Applications

Seminars first half of study year 1994-95

Aug.	17	Kevin Ryan, Univ. of Limerick, Ireland – Requirements Engineering
	30	Michael Schrefl, Univ. of Lienz, Austria – *Extending Object-Oriented Systems with Roles*
	31	Ernst A. Heinz, Univ. of Karlsruhe, Germany – Towards Improved Programmability of Parallel Computers by Integrated Development of Languages, Compilers, and Hardware Architecture
Sept.	1	Peter Gray, Univ. of Aberdeen, Scotland – *The Colan Constraint language and its* Implementation by Active Rules
	6	Claes Wohlin, Lunds Univ. – *Software Reliability from a Cleanroom Perspective*
	8	Martin Sköld, Linköping Univ., Lic. seminar – *Active Rules based on Object Relational* Queries – Efficient Change Monitoring Techniques
	9	Henrik Nilsson, Linköping Univ., Lic. seminar – A Declarative Approach to Debugging for Lazy Functional Languages
	29	Pär Carlshamre, Linköping Univ., Lic. seminar – *A Collaborative Approach to Usability* Engineering: Technical Communicators and System Developers in Usability-Oriented Systems Development
Oct.	3	Victor R. Basili, Univ. of Maryland, USA – *Software Engineering as a Laboratory Science*
	7	Henry M. Paynter, MIT – *Real Time Modelling and Control of Physical Experiments* or in an AI Terminology: An Autonomous agent for automated experimental research
	18	Daniel Gopher, Technion, Israel – Studies of Communication and Response Sharing in Team Work
	21	Stefan Cronholm, Linköping Univ., Lic. seminar – Varför CASE-verktyg i system-utveckling? En motiv- och konsekvensstudie avseende arbetssätt och arbetsformer
	19	Jana Koehler, DFKI Saarbrücken, Tyskland – *Correct Modification of Complex Plans*
Nov.	7	Richard Kaufmann, Digital Equipment Corp. – *Scalable Parallel Processor Architecture with "Reflected Memory"*
	15	Wayne W. Zachary, CHI Systems, Inc. – *A Workbench for Developing Intelligent Agents*
	24	Olav Hansen, Technical Univ. of Munchen – *Performance Analysis of* Parallel Programs
	24	Johan Krammer, Technical Univ., Munchen, Germany – comments on the projects done by – the Parallel Processing group at TU, Munich
	29	Christopher Ahlberg, Chalmers Tekniska Högskola, Göteborg – *Visual Information Seeking*
Dec.	6	Svetlana Linkova, Rjan Radio Engineering Institute, Russia – *What is it Like to Live in Russia*
	6	Björn Malmberg – *Digital Equipment och Digital Business Partners*
	6	Mikael Lindvall, Linköping Univ., Lic. seminar – *A Study of Traceability in Object-Oriented* Systems Development
	15	Cecile Montarnal, INRA, Grenoble – *Linearization Activity in Discourse: The Structure of* Spatial Configuration Description
	19	Hans Olsén, Linköping Univ., Lic. seminar – Collage Induction: Proving Properties of Logic Programs by Program Synthesis
	20	Jonas Lind, Linköping Univ., Lic. seminar – Creditor – Firm Relations: an Inter-disciplinary Analysis
	20	Fredrik Nilsson, Linköping Univ., Lic. seminar – Strategi och ekonomisk styrning – En studie av Sandviks förvärv av Bahco Verktyg

Appendix C
Courses and Teachers in the Undergraduate Education 1993/94

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 140 such undergraduate courses with a total of almost 5500 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 runs 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
Engineering Biology (TB), 30 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), 30 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 (SVP) and the program for Cognitive Science (KGP) ranges over three years and both have an annual intake of 30 students.

In the list below “ADB” means single subject courses in administrative data processing, and “Inf” means single subject courses in Informatics. “SL” means 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 tillämpning (C4, D4)	System Development	Christian Krysander
CASE (C4, D4)	CASE for Development and Maintenance	Lars Viklund/ Peter Fritzson
Objektorienterad systemutveckling (C3, D3, DI2)	Object-Oriented System Development	Johan Fagerström
Programutvecklingsmetodik och programmeringsprojekt (D4)	Methodology of Program Development and Project	Christian Krysander
Ledarskap (C4, D4)	Leadership	Christian Krysander
Informationssystem och förändringsarbete (SVP1)	Information Systems and the Proess of Changes in Enterprises	Ivan Nilsson
Systemutvecklingsprojekt (SVP1)	System Development	Mikael Johansson
Databaser och datamodellering (SVP2)	Data Bases and Data Modelling	Ivan Nilsson
Systemförvaltning (SVP2)	System Maintenance Management	Christian Krysander
Komparativ analys av systemutvecklingsmetoder (SVP2)	Comparative Study of Methodology for Systems Analysis	Mikael Johansson
Samhällsvetenskaplig metodik (SVL3)	Methods in Social Science	Göran Goldkuhl
Teorier och strategier för informationssystem (SVL3)	Theories and Strategies on Information Systems	Torbjörn Näslund /Göran Goldkuhl
CASE-verktyg för systemutveckling (SVL3)	CASE for System Development	Stefan Cronholm
Objektorienterad systemutveckling (SVL3)	Object-Oriented System Development	Johan Fagerström
Informationssystem och förändringsarbete (ADB1)	Information Systems and the Process of Change in Enterprises	vakant
Systemutvecklingsprojekt (ADB1)	System Development Project	Ivan Nilsson
Databaser och datamodellering (ADB1)	Data Bases and Data Modelling	Ivan Nilsson
Systemförvaltning (ADB1)	System Maintenance Management	Christian Krysander
Komparativ analys	Comparative Study	Mikael Johansson
Samhällsvetenskaplig metodik (SVL3)	Methods in Social Science	Göran Goldkuhl
Elektroniska media I (Inf)	Electronic Media I	Roland Hjerppe
Elektroniska media II (Inf)	Electronic Media II	Roland Hjerppe
Informationsresurser I (Inf)	Traditional Information Resources	Roland Hjerppe
Informationsresurser II (Inf)	Digital Information Resources	Roland Hjerppe
4G-språk (SL3)	Fourth Generation Languages	Stefan Cronholm
Praktik (SVL2, SVL3)	Practical Work	Lise-Lotte Raunio
Examensarbete (SVL3)	Degree Project	Hans Holmgren/ Mikael Johansson
C-uppsats		Hans Holmgren
D-uppsats		Torbjörn Näslund/ Karin Axelsson

Software Design: Olle Willén, director of undergraduate studies

Course name in Swedish	Course name in English	Examiner
Orientering, datateknik och datorutrustning (C1, DI1, EI1)	Introduction to Computer Science and Computer Equipment	Torbjörn Jonsson
Programmering i Ada (M4)	Programming in Ada	Olle Willén
Ada och programspråk (C3, D3, D4, DX3)	Ada and Programming Languages	Tommy Olsson
Datastrukturer (Y4, M4, M5, I3, I4)	Data Structures	Katarina Löfstrand
Konstruktion och analys av algoritmer (C4, D5)	Design and Analysis of Algorithms	Per-Olof Fjällström
Logikprogrammering (C3, D4)	Logic Programming	Ulf Nilsson
Programutvecklingsmetodik M (M3)	Methodology of Program Development	Olle Willén
Data- och programstrukturer (C2, D4)	Data and Program Structures	Erik Stoy/ Anders Haraldsson
Datorsystem och programmering (M1)	Computer Systems and Programming	Torbjörn Jonsson
Programvarukvalitet (C4, D5)	Software Quality	Christian Krysander
Orientering, datateknik och datorutrustning (D1)	Introduction to Computer Science and Computer Equipment	Simin Nadjm-Tehrani
Imperativa språk (C1)	Imperative Programming Languages	Ulf Nilsson
Programmering - tillämpning och datastrukturer (Y3)	Programming and Data Structures	Tommy Olsson
Programmering I, gk (I2, TB1)	Introduktion to Programming	Torbjörn Jonsson
Programmering - abstraktion och modellering (Y2)	Programming - Abstraction and Modelling	Jalal Maleki
Programmering - abstraktion och modellering (M2)	Programming - Abstraction and Modelling	Staffan Bonnier
Datastrukturer och algoritmer (C2, D2, DX3)	Data Structures and Algorithms	Christer Bäckström
Programmering i inkrementellt system (C1)	Programming in Incremental System	Åke Thurée
Programmering i inkrementellt system (D1)	Programming in Incremental System	Anders Haraldsson
Datorer och programmering (Kemi1)	Computers and Programming	Jonas Wallgren
Programmering i C (DI1, EI1)	Programming in C	Thomas Padron-McCarthy
Datastrukturer och algoritmer (DI1)	Data Structures and Algorithms	Tommy Olsson
Programmeringsprojekt (DI1)	Program Development Project	Thomas Padron-McCarthy
Introduktion till programmering och datateknik, Pascal (SVP1)	Introduction to Programming and Computer Architecture, Pascal	Torbjörn Jonsson
Programkonstruktion (SVP2)	Program Design	Mariam Kamkar
Programutvecklingsmetodik (SVL3, ADB3)	Methodology of Program Development	Olle Willén
Programmering i C och principer för programspråk (SVL3, ADB4)	Progamming in C and Programming Language Concepts	Katarina Löfstrand
Programmering (KVP1)	Programming	Anders Haraldsson
Introduktion till programmering och datateknik, Pascal (ADB1)	Introduction to Programming and Computer Architecture, Pascal	Katarina Löfstrand

Course name in Swedish	Course name in English	Examiner
Introduktion till programmering och datateknik, Pascal (ADB1)	Introduction to Programming and Computer Architecture, Pascal	Jacek Malek
Programkonstruktion (ADB2)	Program Design	Hans Holmgren
Programmering, gk	Introductory Programming	Katarina Löfstrand

Cognitive Sciences: Jalal Maleki, director of undergraduate studies

Course name in Swedish	Course name in English	Examiner
Databaser (I4, I3, M4, Y4)	Databases	Åke Thurée
Artificiell intelligens D (D3, DX3)	Artificial Intelligence D	Jalal Maleki
AI-programmering (C3, C4)	AI Programming	Henrik Eriksson
Logik, grundkurs (C1, D2, DX3)	Logic, introductory course	Erik Sandewall
AI kunskapsrepresentation (C4)	Representation of Knowledge in AI	Patrick Doherty
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
Artificiell intelligens C (C2)	Artificial Intelligence C	Ivan Rankin
Expertsystem – metodik och verktyg (C4, D4, I4, M4, Y4)	Expert Systems – Methods and Tools	Henrik Eriksson
Datalingvistik (C2)	Computational Linguistics	Lena Strömbäck
Lingvistik grundkurs (C1, D4, Y4)	Introduction to Linguistics	Ivan Rankin
Kognitionsvetenskapliga kommunikations-modeller (C3)	Cognitive Science Models of Communication	Ivan Rankin
Människa-datorinteraktion (C3, D4)	Human-Computer Interaction	Jonas Löwgren
Databasteknik (C3, D4)	Database Technology	Tore Risch
Diskret matematik och Logik (D1)	Discrete Mathematics and Logic	Dimiter Driankov
Tekniska databaser (DI2)	Technical Databases	Jalal Maleki
Människa-datorinteraktion (SVL3)	Human Computer Interaction	Hans Holmgren
Expertsystem och kunskapsteknik (SVL3)	Expert Systems and Knowledge Engineering	Sture Hägglund
Design för användbarhet (SVL3, ADB4)	Design for Usability	Jonas Löwgren
Introduktion till kognitionsvetenskap (KGP1)	Introduction fo Cognitive Science	Nils Dahlbäck
Logik och diskret matematik (KGP1)	Logic and Discrete Mathematics	Patrick Doherty

System Architecture: Johan Fagerström, director of undergraduate studies

Course name in Swedish	Course name in English	Examiner
Processprogrammering (C3, D3, Dx3)	Concurrent Programming	Rober Bilos
Kompilatorer och interpretatorer (D4, Y4)	Compilers and Interpreters	Nahid Shahmehri
Programmeringsteori II (C4)	Programming Theory II	Johan Boye/ Ulf Nilsson
Kompilatorkonstruktion (C3, D4)	Compiler Construction	Peter Fritzson
Programmeringsteori (C3)	Programming Theory	Nahid Shahmehri
Realtids- och processprogrammering (M4, Y4)	Real Time and Concurrent Programming	Anders Törne
Distribuerade system (C4, D4)	Distributed Systems	Rober Bilos

Course name in Swedish	Course name in English	Examiner
Formella språk och automatateori (C2, D4)	Formal Languages and Automata Theory	Johan Boye/ Ulf Nilsson
Systemprogramvara (Y4)	System Software	Lars Strömberg
Diskret simuleringsteknik (D4, Y4)	Discrete Simulation	Zebo Peng
Datornät (C4, D5, Y5)	Computer Networks	Lars Strömberg
Datorarkitektur (C4, D4, Y4)	Advanced Computer Architecture	Krzysztof Kuchcinski
Datorstödd elektronikkonstruktion (D4, Y4)	Computer Aided Design of Electronics	Mats Larsson/ Krzysztof Kuchcinski
Datorarkitektur, projektkurs (D5, Y5)	Advanced Computer Architecture Project	Lars Viklund/ Johan Fagerström
Operativsystem (DI2)	Operating Systems and System Programming	Rober Bilos
Datornät och distribuerade system (DI2)	Computer Networks and Distributed Systems	Lars Strömberg
Datorarkitektur (DI2)	Advanced Computer Architecture	Krzysztof Kuchcinski
Systeminstallation (DI2)	System Installation	Lars Strömberg
Realtids- och processprogrammering (DI2)	Real Time and Concurrent Programming	Bengt Lennartsson
Programmering, fk (EI2)	Programming, advanced course	Rober Bilos
Datorsystem och programmering (MI1)	Introduction to Computer Science and Programming	Katarina Löfstrand

Economic Information Systems: Jörgen Andersson, director of undergraduate studies

Course name in Swedish	Course name in English	Examiner
Industriell Ekonomi (M2)	Industrial Economics and Management	Lennart Ohlsén
Redovisning och budgetering (I2)	Accounting and Budgeting	Lennart Ohlsén
Informationssystem, gk (I3, M4)	Information Systems, basic course	Jörgen Andersson
Teknisk och ekonomisk utvärdering av datorsystem (M4, I4, D4)	Technical-Economic Evaluation of Information Systems	Lennart Ohlsén
Ekonomiska informationssystem, ak (I4)	Economic Information Systems	Bengt Savén
Ekonomiska informationssystem, fk (I4)	Economic Information Systems, advanced course	Björn Helander
Modellbyggnad (I3)	Modelling	Jaime Villegas
IT, Kommunikation och organisationsdesign (I4, M4)	IT, Communication and Organizational Design	Kristina Larsen

C.2 Teaching staff

The following persons from IDA have been responsible for one or more courses:

Jörgen Andersson, B. Ec
Thomas Bennet, B. Ec.
Rober Bilos, Ph. D.
Staffan Bonnier, Ph. D.
Johan Boye, Lic.
Christer Bäckström, Ph. D.
Stefan Cronholm, Lic.
Nils Dahlbäck, Ph. D.
Patrick Doherty, Ph. D.
Dimiter Driankov, Ph. D.
Henrik Eriksson, Ph. D.
Johan Fagerström, Ph. D.
Per-Olof Fjällström, Ph. D.
Peter Fritzson, Ph. D.
Göran Goldkuhl, Ph. D.
Anders Haraldsson, Ph. D.
Björn Helander, B. Ec.
Roland Hjerppe, M. Sc.
Hans Holmgren, M. Sc. E.
Sture Hägglund, Ph. D.
Mikael Johansson, M. Sc.
Torbjörn Jonsson, M. Sc.
Mariam Kamkar, Ph. D.
Krzysztof Kuchcinski, Ph. D.
Christian Krysander, M. Sc.
Kristina Larsen, B. Sc.
Mats Larsson, Lic.
Bengt Lennartsson, Ph. D.
Katarina Löfstrand, M. Sc.
Jonas Löwgren, Ph. D.
Simin Nadjm-Tehrani, Ph. D.
Jalal Maleki, Lic.
Ivan Nilsson, B. Sc.
Ulf Nilsson, Ph. D.
Torbjörn Näslund, Lic.
Thomas Padron-McCarthy, M. Sc.
Lennart Ohlsén, M. Sc.
Tommy Olsson, M. Sc.
Zebo Peng, Ph. D.
Ivan Rankin, Lic.
Lise-Lotte Raunio, M. Sc.
Tore Risch, Ph. D.
Erik Sandewall, Ph. D.
Bengt Savén, M.Sc.
Nahid Shahmehri, Ph. D.
Erik Stoy, M. Sc.
Lars Strömberg, Lic.
Lena Strömbäck, Lic.
Stefan Svenberg, M. Sc.
Eva-Chris Svensson, M. Sc.
Åke Thurée, M. Sc.
Anders Törne, Ph. D.
Jonas Wallgren, M. Sc.
Olle Willén, B. Sc.
Lars Viklund, M. Sc.
Jaime Villegas, B. Ec.

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 pace with the rapid development in the area, e.g. regarding the emergence of powerful workstations with high-resolution graphics and high-performance CPUs. Our orientation towards experimental computer science makes such a policy especially important and we believe that adequate computer equipment is essential for quality of research and education.

Table D–1. *Computer resources, January 1995.*

265	Sun workstations
28	Sun file servers
4	HP workstations
1	IBM AS/400 MiniMainframe
25	Macintoshes
80	IBM PC compatibles
2	VAX 3200
20	Laser printers
1	Supercomputer Parsytec GC

Our main computer resources are the Sun SPARC-10 workstations and fileservers.

In addition there are lots of smaller computers (MicroVax, Macintoshes and other PCs of various kinds) and around 20 laser printers.

There is also special purpose equipment for specific research projects.

The latest resource for research purpose is the parallel supercomputer Parsytec GC with 128 processors.

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.
No. 297 **Mariam Kamkar:** Interprocedural Dynamic Slicing with Applications to Debugging and Testing, 1993.
No. 302 **Tingting Zhang:** A Study in Diagnosis Using Classification and Defaults, 1993.
No. 312 **Arne Jönsson:** Dialogue Management for Natural Language Interfaces – An Empirical Approach, 1993.
No. 338 **Simin Nadjm-Tehrani:** Reactive Systems in Physical Environments: Compositional Modelling and Framework for Verification, 1994.
No. 371 **Bengt Savén:** Business Models for Decision Support and Learning. A Study of Discrete-Event Manufacturing Simulation at Asea/ABB 1968-1993, 1995.
No. 375 **Ulf Söderman:** Conceptual Modelling och Mode Switching Physical Systems, 1995.
No. 383 **Andreas Kågedal:** Exploiting Groundness in Logic Programs, 1995.

Dissertation by IDA member published elsewhere.

Lars Ahrenberg: Interrogative Structures of Swedish: Aspects of the Relation between Grammar and Speech Acts. (Reports from Uppsala Univ. Dept. of Linguistics No. 15, 1987).

E.2.1 PhD Dissertation Abstracts

No. 292

STUDIES IN INCREMENTAL NATURAL-LANGUAGE ANALYSIS

Mats Wirén

This thesis explores the problem of incremental analysis of natural-language text. Incrementality can be motivated on psychological grounds, but is becoming increasingly important from an engineering perspective as well. A major reason for this is the growing importance of highly interactive, "immediate" and real-time systems, in which sequences of small changes must be handled efficiently.

The main technical contribution of the thesis is an incremental parsing algorithm that analyses arbitrary changes (insertions, deletions and replacements) of a text. The algorithm is grounded in a general chart-parsing architecture, which allows different control strategies and grammar formalisms to be used. The basic idea is to analyse changes by keeping track of dependencies between partial analyses (chart edges) of the text. The algorithm has also been adapted to interactive processing under a text editor, thus providing a system that parses a text simultaneously as it is entered and edited. By adopting a compositional and dynamic model of semantics, the framework can be extended to incremental interpretation, both with respect to a discourse context (induced by a connected, *multisentential* text) and a non-linguistic context (induced by a model of the world).

The notion of keeping track of dependencies between partial analyses is similar to reason maintenance, in which dependencies are used as a basis for (incremental) handling of belief changes. The connections with this area and prospects for cross-fertilization are discussed. In particular, chart parsing with dependencies is closely related to assumption-based reason maintenance. Both of these frameworks allow competing analyses to be developed in parallel. It is argued that for the purpose of natural-language analysis, they are superior to previously proposed, justification-based approaches, in which only a single, consistent analysis can be handled at a time.

No. 297

INTERPROCEDURAL DYNAMIC SLICING WITH APPLICATIONS TO DEBUGGING AND TESTING

Mariam Kamkar

The need of maintenance and modification demand that large programs be decomposed into manageable parts. Program slicing is one method for such decomposition. A program slice with respect to a specified variable at some program point consists of those parts of the program that may directly or indirectly affect the value of that variable at the particular program point. This is useful for understanding dependencies within programs. A static program slice is computed using static data and control flow analysis and is valid for all possible executions of the program. Static slices are often imprecise, i.e., they contain unnecessarily large parts of the program. Dynamic slices however, are precise but are valid only for a single execution of the program. Interprocedural dynamic slices can be computed for programs with procedures, and these slices consist of all executed call statements which are relevant for the computation of the specified variable at the specified program point.

This thesis presents the first technique for *interprocedural dynamic slicing* which deals with procedures/ functions at the abstract level. This technique first generates summary information for each procedure call (or function application), then represents a program as a summary graph of dynamic dependencies. A slice on this graph consists of vertices for all procedure calls of the program that affect the value of a given variable at the specified program point. The amount of information saved by this method is considerably less than what is needed by previous methods for dynamic slicing, since it only depends on the size of the *program's execution tree*, i.e., the number of executed procedure calls, which is smaller than a trace of all executed statements.

The interprocedural dynamic slicing method is applicable in at least two areas, *program debugging* and *data flow testing*. Both of these applications can be made more effective when using dynamic dependence information collected during program execution. We conclude that the interprocedural dynamic slicing method is superior to other slicing methods when precise dependence information for a specific set of input data values at the procedural abstraction level is relevant.

No. 302

A STUDY IN DIAGNOSIS USING CLASSIFICATION AND DEFAULTS

Tingting Zhang

This dissertation reports on the development of a model and system for medical diagnosis based on the use of general purpose reasoning methods and a knowledge base which can be built almost entirely from existing medical texts. The resulting system is evaluated empirically by running 63 patient protocols collected from a community health centre on the system, and comparing the diagnoses with those given by medical experts.

It is often the case in Artificial Intelligence that general purpose reasoning methods (such as default reasoning, classification, planning, inductive learning) are developed at a theoretical level but are not used in real applications. One possible reason for this is that real applications typically need several reasoning strategies to solve a problem. Combining reasoning strategies, each of which uses a different representation of knowledge is non-trivial. This thesis addresses the issue of combining strategies in a real application. Each of the strategies used required some modification, either as a result of the representation chosen, or as a result of the application demands. These modifications can indicate fruitful directions for future research.

One well known problem in building A.I. systems is the building of the knowledge base. This study examines the use of a representation and method which allowed for the knowledge base to be built from standard medical texts with only minimal input from a medical expert.

The evaluation of the resulting system indicated that in cases where medical experts were in agreement, the system almost always reached the same diagnosis. In cases where medical doctors themselves disagreed the system behaved within the range of the medical doctors in the study.

No. 312

DIALOGUE MANAGEMENT FOR NATURAL LANGUAGE INTERFACES – AN EMPIRICAL APPROACH

Arne Jönsson

Natural language interfaces are computer programs that allow a person to communicate with a computer system in his own language. This thesis deals with management of coherent dialogue in natural language interfaces, which involves addressing the issues of focus structure and dialogue structure. Focus structure concerns the recording of entities mentioned in the discourse to allow a user to refer to them in the course of the interaction, dialogue structure involves handling the relationships between segments in the dialogue.

In a theoretical investigation two approaches to dialogue management are compared: one is based on recognizing the user's plan from his goals and intentions, and the other on modelling the possible actions of the user in a dialogue grammar. To establish a sound foundation for the design of the dialogue manager, empirical studies were carried out in the form of Wizard of Oz experiments. In such studies users interact with what they think is a natural language interface, but in fact there is a human intermediary. Conducting experiments of this kind requires careful design and a powerful simulation environment. Such an environment is presented together with guidelines for the design of Wizard of Oz experiments. The empirical investigations indicate that dialogue in natural language interfaces lack many of the complicated features characterizing human dialogue. Furthermore, the kind of language employed by the user is dependent to some extent on the application, resulting in different sublanguages.

The results from the empirical investigations have been subsequently used in the design of a dialogue manager for natural language interfaces which can be used in a variety of applications. The dialogue manager utilizes the restricted but more computationally feasible approach of modelling dialogue structure in a dialogue grammar. Focus structure is handled via dialogue objects modelled in a dialogue tree. The dialogue manager is designed to facilitate customization to the sublanguage utilized in various applications. In the thesis I discuss how the dialogue manager is customized to account for the dialogue behaviour in three applications. The results demonstrate the feasibility of the proposed approach to building application-specific dialogue managers for various applications.

No. 338

REACTIVE SYSTEMS INPHYSICAL ENVIRONMENTS: COMPOSITIONAL MODELLING AND FRAMEWORK FOR VERIFICATION

Simin Nadjm-Tehrani

This thesis addresses the question of correctness of reactive programs which are embedded in physical environments, and which perform a combination of symbolic and numeric computations. Such *hybrid* systems are of growing interest in the areas of artificial intelligence, control engineering, and software verification. The verification of hybrid systems requires the use of modular models and a combination of discrete and continuous modelling techniques. The thesis proposes new methods which serve these goals. The proposed methods are closely related to a layered software architecture hosting both synchronous and asynchronous computations. The architecture has been used for the development of prototype automobile co-driver systems.

We consider the adequacy of representational formalisms for hybrid systems. To this end, modular models for verification of an anti-collision device are studied at three levels of abstraction. First, *dynamic transition systems* (DTS) and their timed version TDTS are proposed for the discrete models of the physical environment and the controller. Using the detailed example, the derivation of discrete environment models from physical models is discussed – a point of emphasis being the association of discrete modes with regions in the continuous state space. Next, the models are compared with a *hybrid transition system* in which the continuous changes are represented explicitly.

We show that if strict modularity *wrt*. the sequence of control actions is required, a physically motivated timed (discrete) model for the environment can not be obtained by simply adding timing constraints to the *untimed* model. The iterative method used for the derivation of *untimed* models is then extended by inclusion of *memory* modes. In the hybrid model, this complete separation of the plant and the controller can be achieved with minimal effort.

The thesis presents formal definitions, operational semantics, and parallel composition operators for the three types of transition systems. Two novel features of the hybrid formalism enable a convenient interface to physical models of mode switching systems. First, the separation of state and input variables, and second, the use of algebraic equations for the description of change in some continuous variables. A variant of metric temporal logic is also presented for description of complex transformations of quantitative to qualitative values.

No. 371

BUSINESS MODELS FOR DECISION SUPPORT AND LEARNING. A STUDY OF DISCRETE-EVENT MANUFACTURING SIMULATION AT ASEA/ABB 1968-1993

Bengt Savén

This thesis describes the planning, execution and results of an embedded case study of discrete-event manufacturing simulation at Asea Brown Boveri's (ABB) operations in Sweden from 1968 through 1993. The main aim of the study has been to explore and learn more about the values created from manufacturing simulations. This is done by describing and analyzing the context of manufacturing simulation within one company group for a long period of time. The work has focused on four issues: the manufacturing systems, the simulation software tools, the application projects and the developments over the 26 years time horizon. The study is based on personal interviews, questionnaires and documents, such as project reports, meeting minutes etc.

Two in-house manufacturing simulators are described and compared with the two most frequently used standard software tools during the 26 year period. The most important differences between the tools were found in the ease of learning and use of the tools, the modeling flexibility and the model visibility. 61 projects in which this software has been applied are described and analyzed. The majority of the projects involve capacity planning and/or evaluation of control rules.

Three recent projects within one division are described and analyzed in detail. The values created are more diverse than expected. Generally the literature brings us the notion of simulation as a tool for evaluating alternatives in a decision process. However, the study shows that this is just one of twelve possible motives for using simulation. A model is suggested that distinguishes these possible motives along three dimensions: focus on process, focus on phase and focus on actors.

Different hypotheses as to why the use of simulation has changed over the 26 year period are discussed. One reason is found to be the level of investment and the software capabilities. However, management's interest in manufacturing in general and organizational learning through simulation in particular seem to be of greater importance. Trends in the manufacturing industry and their impact on the demand for simulation are also discussed in the text, as well as a comparison between discrete-event simulation and some alternatives for capacity planning.

No. 375

CONCEPTUAL MODELLING OF MODE SWITCHING PHYSICAL SYSTEMS

Ulf Söderman

This thesis deals with fundamental issues underlying the systematic construction of behaviour models of *mode switching engineering systems*, i.e. systems constructed by engineers involving continuous as well as discrete behavioural changes. The aim of this work is to advance the design and development of effective computer aided modelling systems providing high-level support for the difficult and intellectually demanding task of model construction. In particular, the thesis is about *conceptual modelling* of engineering systems, *i.e*. modelling characterized by the explicit use of well defined abstract physical concepts. A comprehensive review of conceptual modelling is presented, discussing modelling in its own and forming a reference for the development of computer aided modelling systems.

The main contribution of this work is the extension of the conceptual modelling framework by an abstract and generic concept referred to as the *ideal switch concept.* This concept enables a uniform and systematic treatment of mode switching engineering systems. In the discussion of the switch concept and its usage, the energy based *bond graph approach* is employed as a specific example of a conceptual modelling approach. The bond graph version of the switch concept is presented. This switch element complies with the classical bond graph modelling formalism and hence the extended formalism, here referred to as *switched bond graphs,* preserves all the essential properties of classical bond graphs. The systematic method for construction of bond graphs can be applied. Component models can remain context independent through acausal modelling and causal analysis can be performed automatically at the bond graph level.

Furthermore, for the representation of overall computational models of mode switching systems a mathematical structure related with state automata is introduced. This structure is referred to as *mode transitions systems.* For the mathematical characterization of individual switches a simplified version of this structure, referred to as *switch transition system,* is introduced. The systematic construction of computational models is discussed and a systematic method is proposed. For this purpose a transition system composition operator for *parallel composition* is presented.

No. 383

EXPLOITING GROUNDNESS IN LOGIC PROGRAMS

Andreas Kågedal

The logical variable is one of the distinguishing features of logic programming, but it has been noticed that its full potential is used quite restrictively. Often program predicates are used in a “directional” way, where argument positions are partitioned into *input* and *output* positions. At every call of a given predicate, input arguments are bound to ground terms and at success of the call the output arguments will also have been instantiated to ground terms. This thesis addresses two aspects related to this kind of directionality in logic programming.

The first part of the thesis is a continuation and implementation of the work of Bonnier and Maluszynski. They give a theoretical framework for how *external procedures* written in another programming language can be integrated into a logic programming framework without sacrificing a declarative reading. In many Prolog systems, one is allowed to call an external procedure as a directional predicate from a program clause. With nonground arguments this may cause unpredictable effects and often leads to a run-time error. Instead, Bonnier/Maluszynski view external procedures as functions which will not be evaluated until all arguments are ground. The thesis defines a language *GAPLog*, a superset of Prolog, using this kind of external procedures. Systematic development of its implementation by transformation techniques is one of the contributions of this thesis. The result is a compiler from GAPLog to (SICStus) Prolog.

The second part of the thesis is a continuation of Kluzniak’s work concerning data flow analysis of programs written in *Ground Prolog*. In Ground Prolog, argument positions of all predicates must be user-defined as either input or output positions. Input values are required to be ground at call time and output values—at success. This restriction enabled Kluzniak to develop a specialized method for data flow analysis which can be used for inferring *liveness* information. An interesting feature of this approach is that it provides a conceptual model for the analysis of data flow between individual program variables. However, it is presented in a rather informal way. This makes it difficult to understand the mechanisms of approximations and to ensure the correctness of the method. The main contribution of the second part is a theoretical framework designed for Kluzniak’s method and based on *abstract interpretation*. A concept of dependency graph between program variables is systematically derived from a formal semantics based on the notion of proof tree. The derivation steps clearly indicate the design decisions taken. This allows for a better understanding of the method and a more precise approximation of the program's data flow. Kluzniak's work on *liveness analysis* for Ground Prolog is also extended and improved.

E.3 Licentiate 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. (Swedish).
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.
No. 371 **Mehran Noghabai:** Evaluation of Strategic Investments in Information Technology, 1993.
No. 378 **Mats Larsson:** A Transformational Approach to Formal Digital System Design, 1993.
No. 380 **Johan Ringström:** Compiler Generation for Parallel Languages from Denotational Specifications, 1993.
No. 381 **Michael Jansson:** Propagation of Change in an Intelligent Information System, 1993.
No. 383 **Jonni Harrius:** An Architecture and a Knowledge Representation Model for Expert Critiquing Systems, 1993.
No. 386 **Per Österling:** Symbolic Modelling of the Dynamic Environments of Autonomous Agents, 1993.
No. 398 **Johan Boye:** Dependency-based Groudness Analysis of Functional Logic Programs, 1993.
No. 402 **Lars Degerstedt:** Tabulated Resolution for Well Founded Semantics, 1993.
No. 406 **Anna Moberg:** Satellitkontor – en studie av kommunikationsmönster vid arbete på distans, 1993. (Swedish).
No. 414 **Peter Carlsson:** Separation of Management and Financing, Case Studies of Management Buy-Outs in an Agency Theoretical Perspective, 1994.
No. 417 **Camilla Sjöström:** Revision och lagreglering – ett historiskt perspektiv, 1994. (Swedish).
No. 436 **Cecilia Sjöberg:** Voices in Design: Argumentation in Participatory Development, 1994.
No. 437 **Lars Viklund:** Contributions to a High-level Programming Environment for Scientific Computing, 1994.
No. 440 **Peter Loborg:** Error Recovery Support in Manufacturing Control Systems, 1944.
FHS 3 **Owen Eriksson:** Informationssystem med verksamhetskvalitet – utvärdering baserat på ett verksamhetsinriktat och samskapande perspektiv, 1994. (Swedish).
FHS 4 **Karin Pettersson:** Informationssystemstrukturering, ansvarsfördelning och användarinflytande – En komparativ studie med utgångspunkt i två informationssystemstrategier, 1994. (Swedish).
No. 441 **Lars Poignant:** Information Technology and Industrial Location – Effect on Productivity and RegionalDevelopment, 1994.
No. 446 **Gustav Fahl:** Object Views of Relational Data in Multidatabase Systems, 1994.
No. 450 **Henrik Nilsson:** A Declarative Approach to Debugging for Lazy Functional Languages, 1994.
No. 451 **Jonas Lind:** Crediter – Firm Relations: an Interdisciplinary Analysis, 1994.
No. 452 **Martin Sköld:** Active Rules based on Object Relational Queries – Efficient Change Monitoring Techniques, 1994.
No. 455 **Pär Carlshamre:** A Collaborative Approach to Usability Engineering: Technical Communicators and System Developers in Usability-Oriented Systems Development, 1994.
FHS 5 **Stefan Cronholm:** Varför CASE-verktyg i systemutveckling? – en motiv- och konsekvensstudie avseende arbetssätt och arbetsformer, 1994. (Swedish).

No. 462 **Mikael Lindvall:** A Study of Traceability in Object-Oriented Systems Development, 1994.
No. 463 **Fredrik Nilsson:** Strategi och ekonomisk styrning – En studie av Sandviks förvärv av Bahco Verktyg, 1994. (Swedish).
No. 464 **Hans Olsén:** Collage Induction: Proving Program Properties by Program Synthesis, 1994.
No. 469 **Lars Karlsson:** Specification and Synthesis of Plans Using the Features and Fluents Framework, 1995.
No. 473 **Ulf Söderman:** On Conceptual Modelling of Mode Switching Systems, 1995.
No. 475 **Choong-ho Yi:** Reasoning about Concurrent Actions in the Trajectory Semantics, 1995.
FHS 7 **Anders Avdic:** Arbetsintegrerad systemutveckling med kalkylprogram, 1995.
No. 476 **Bo Lagerström:** Successiv resultatavräkning av pågående arbeten. Fallstudier i tre byggföretag, 1995.
No. 478 **Peter Jonsson:** Complexity of State-Variable Planning under Structural Restrictions, 1995.
No. 482 **Eva L Ragnemalm:** Towards Student Modelling through Collaborative Dialogue with a Learning Companion, 1995.
No. 488 **Eva Toller:** Contributions to Parallel Multiparadigm Languages: Combining Object-Oriented and Rule-Based Programming, 1995.
No. 489 **Erik Stoy:** A Petri Net Based Unified Representation for Hardware/Software Co-Design, 1995.
No. 497 **Johan Herber:** Environment Support for Building Structured Mathematical Models, 1995.
No. 498 **Stefan Svenberg:** Structure Driven Derivation of Inter-Lingual Functor-Argument Trees for Multi-Lingual Generation, 1995.

E.4 Master of Engineering Theses

No. 1 **Patrick Lambrix:** Management of Historical Information of Composite Objects, 1992.
No. 2 **Ying Shu:** The Analysis and Implementation of a Terminological Reasoner, 1993.

E.5 External Publications since 1993

Refereed papers published in books, journals or international conference proceedings.

Ahrenberg, L. (1993). Topological frames in sign-based grammars. In *NODALIDA'93. Proc. from 9:e Nordiska datalinvistikdagarna* (Swedish), pp. 9-17, Stockholm, Sweden, June 3-5, 1993.

Ahrenberg, L., Dahlbäck, N., Jönsson, A. (1994). Coding Schemes for Studies of Natural Language Dialogue. To appear in *Working notes from AAAI Spring Symp.,* Stanford, Mar. 27-29, 1995.

Ahrenberg, L., Jönsson, A., Thurée, Å. (1993). Customizing interaction for natural language interfaces workshop on pragmatics in dialogue. In *Proc. of the XIV:th Scand. Conf. of Linguistics and the VIII:th Conf. of Nordic and General Linguistics*, Göteborg, Sweden.

Ahrenberg, L., Merkel, M. (1994a). Translation corpora and translation support tools. *Languages in Contrast. Text-based cross-linguistic studies*, Lund Univ. Press, Sweden.

Andersson, B., Nilsson, P. (1992). How to manage complexity in inter-organizational information systems – some preliminary conclusions. In *Proc. of IFIP 8.3/8.5 Conf. "Decision support in public administration"*, Noordwijkerhout, The Netherlands.

Andersson, N., Fritzson, P. (1994). Overview and industrial application of code generator generators. *Journal of Systems and Software*, Feb. issue, Elsevier Science Publ. Company Inc.

Andersson, N., Fritzson, P. (1995). Generating Parallel Code from Object Oriented Mathematical Models. Accepted to the *Fifth ACM SIGPLAN Symp. on Principles and Practice of Parallel Programming*, Santa Barbara, California, July 19-21, 1995.

Axelsson, J. (1995). Hardware/software partitioning of real-time systems. Accepted to the *IEE colloquium on partitioning in hardware/software codesigns*, London, U.K., Feb. 13, 1995.

Axelsson, J., Menth, S., Semmler, K. (1993). Genetic Algorithms in Industrial Design. In *Proc. of Fifth Int. Conf. on Tools with Artificial Intelligence,* Boston, USA, Nov. 8-11, 1993.

Bennet, T., Olhager, J., Rapp, B. (1994). Computerization of manufacturing planning and control systems in Sweden 1980-1993. In *IFIP Conf. on Evaluation of Production Management Methods*, pp. 355-362, Elsevier Science Publishers B.V., Amsterdam, The Netherlands.

Björklind, A. (1994). An architecture for organising dynamic information about space and time. In Albrechtsen, H. and Oernager, S., eds., *Knowledge Organization and Quality Management. Proc. of the Third Int. ISKO Conf.* Copenhagen, Denmark, (*Advances in Knowledge Organisation*, Vol. 4) pp. 358-365, Frankfurt/Main, Germany, Indeks.

Björklund, L. (1994). The potential of using knowledge organising tools in collaborative system development. In Albrechtsen, H. and Oernager, S., eds., *Knowledge Organization and Quality Management, Proc. of the Third Int. ISKO Conf.* Copenhagen, Denmark, (*Advances in Knowledge Organisation*, Vol. 4) pp. 264-271, Frankfurt/Main, Germany, Indeks.

Björklund, L., Björklind, A., Hjerppe, R. (1994). Highways and Backroads of Internet: Strategies and Tactics. In *Information Superhighway: The Role of Librarians, Information Scientists and Intermediaries, Proc. of the 17th Int. Essen Symp.* Essen, Germany, Oct. 24-27, 1994. In press.

Bol, R., Degerstedt, L. (1993). The Underlying Search for Magic Templates and Tabulation. In D.S. Warren, ed., *Proc. of 10th Int. Conf. on Logic Programming*, pp. 793-811, MIT Press, 1993, Budapest, Hungary, June, 1993.

Bol, R., Degerstedt, L. (1993). Tabulated Resolution for Well Founded Semantics. In D. Miller, ed., *Proc. of the Int. Logic Programming Symp., ILPS '93*, The MIT Press, Vancouver, Canada, Oct. 26-29, 1993.

Boye. J. (1993). Avoiding Dynamic Delays in Functional Logic Programs. In M. Bruynooghe and J. Penjam, eds., *Programming Language Implementation and Logic Programming 93*, LNCS 724, pp. 12-27, Springer-Verlag, 1993.

Boye, J., Maluszynski, J. (1995). Two aspects of directional types. To appear in *Proc. of Int. Logic Programming Conf.*, Tokyo, Japan, MIT Press, June 13-16, 1995.

Boye, J., Paakki, J., Maluszynski, J. (1993). Synthesis of Directionality Information for Functional Logic Programs. In P. Cousot, M. Falschi, G. File, and A. Rauzy, eds., *Proc. of the Workshop on Static Analysis, WSA'93*, LNCS 724, pp. 165-177, Springer-Verlag, 1993.

Busch, D. (1993). An Expressive Three-Valued Logic with Two Negations. *Seventh Int. Symp. on Methodologies for Intelligent Systems, ISMIS'93*, Trondheim, Norway, June 15-18, 1993. In Proc. by Springer-Verlag in the series LNAI.

Busch, D. (1993). Intuitionistic three-valued logic and SLDNF-resolution. In *Proc. of the Second Int. Workshop on Logic Programming and Non-Monotonic Reasoning*, Lisbon, Portugal, 1993.

Busch, D. (1993). A Sequent Axiomatization of Three-Valued Logic with Two Negations. In *Proc. of LPNMR'93. The Second Int. Logic Programming and Nonmonotonic Reasoning Workshop*, Lisbon, Portugal, June 28-30, 1993.

Bäckström, C. (1993a). Finding least constrained plans and optimal parallel executions is harder than we thought. *Current trends in AI planning: EWSP'93—2nd European Workshop on Planning*, Vadstena, Sweden. EWSP, Bäckström, C. and Sandewall, E., eds., IOS Press. Also in *Proc. of the Workshop on Algorithms, Complexity and Commonsense Reasoning at the 11th European Conf. on Artificial Intelligence, ECAI'94*, Amsterdam, The Netherlands, Aug. 8-12, 1994.

Bäckström, C. (1994a). Executing parallel plans faster by adding actions. In Cohn, A. G., ed., *Proc. of the 11th European Conf. on Artificial Intelligence, ECAI'94*, pp. 615–619, Amsterdam, The Netherlands, Wiley.

Bäckström, C. (1994). Finding Least Constrained Plans and Optimal Parallel Executions is Harder than We Thought. In *Proc. of Second European Workshop on Planning, EWSP'93*, Vadstena, Sweden, Dec. 9-11, 1993. In *Proc. of Workshop on Algorithms, Complexity and Commonsense Reasoning at the 11th European Conf. on Artificial Intelligence, ECAI'94*, Amsterdam, The Netherlands, Aug. 8-12, 1994.

Bäckström, C. (1994). Executing Plans Faster by Adding Actions. In *Proc. of the 11th European Conf. on Artificial Intelligence, ECAI'94*, Amsterdam, The Netherlands, Aug. 8-12, 1994.

Bäckström, C. (1995). Expressive Equivalence of Planning Formalisms. Accepted for publication in the journal *Artificial Intelligence, Special Issue on Planning and Scheduling*, 1995.

Bäckström, C., Jonsson, P. (1994). Planning with abstraction hierarchies can be exponentially less efficient. Accepted to *the 14th Int. Joint Conf. on Artificial Intelligence (IJCAI'95)*, Montreal, Canada, Aug., 1995.

Bäckström, C., Nebel, B. (1993). Complexity results for SAS^+ planning. In Ruzena Bajcsy, ed., *Proc. of the 13th Int. Joint Conf. on Artificial Intelligence (IJCAI-93)*, Chambery, France, Aug.–Sept., 1993, Morgan Kaufmann, pp. 1430-1435.

Bäckström, C., Nebel, B. (1994). Complexity results for SAS^+ planning. *Computational Intelligence*. To appear.

Bäckström, C., Sandewall, E. (eds.) (1994). *Current Trends in AI Planning: EWSP'93 – 2nd European Workshop on Planning*. Proc. from this conf. in Vadstena, Sweden, Dec. 9-11, 1993, printed by IOS Press, Amsterdam, The Netherlands, 1994.

Carlshamre, P. (1993). Looking back on my first year as a maintainer. *The Software Practitioner*, 3(6):10–13.

Carlshamre, P. (1994). Technical communicators and system developers collaborating in usability-oriented systems development: A case study. In *Proc. 1994 ACM 12th Annual Int. Conf. on Systems Documentation, (SIGDOC-94)*, pp. 200-207, New York, USA, ACM.

Carlshamre, P., Löwgren, J., Rantzer, M. (1994). Usability meets the real world: A case study of usability-oriented method development in industrial software production. In Bradley, G. E. and Hendrick, H. W., eds., *Proc. 4th Int. Symp. Human Factors in Organization Design and Management (ODAM'94)*, pp. 427–432, North-Holland, The Netherlands.

Cheikes, B. (1993). Methodological issues in the design of intelligent and cooperative information systems. In *Proc. of the Int. Conf. on Intelligent and Cooperative Information Systems (ICICIS-93)*, pp. 3–12, Rotterdam, The Netherlands, IEEE, Computer Society Press.

Cheikes, B., Ragnemalm, E. L. (1993). Simulator-based training-support tools for process-control operators. In *Proc. of the NATO Workshop on Natural Dialogue and Interactive Student Modeling*, Varenna, Italy. To appear. Also available as LiTH-IDA-R-93-29.

Cronholm, S., Goldkuhl, G. (1992). Meanings and motives of method customizations in CASE environments – observations and categorizations from an empirical study. In *Proc. of the 5th European Workshop on Next Generation of CASE tools*, Utrecht, The Netherlands.

Dahlbäck, N., Jönsson, A., Ahrenberg, L. (1993). Wizard of Oz Studies – Why and How. *Knowledge-Based Systems*, 6 (4):258–266.

Degerstedt, L., Nilsson, U. (1995). Magic Computation for Well-founded Semantics. To appear in *Nonmonotonic Extensions of Logic Programming*, LNAI, Springer-Verlag., 1995.

Deransart, P., Maluszynski, J. (1993). *A Grammatical View of Logic Programming*. MIT Press, 1993.

Doherty, P. (1994). Reasoning About Action and Change Using Occlusion. In *Proc. of the 11th European Conf. on Artificial Intelligence, ECAI'94*, Amsterdam, The Netherlands, Aug. 8-12, 1994.

Doherty, P., Driankov, D., Hellendoorn, H. (1993). Fuzzy If-Then-Unless Rules and Their Implementation. Published in *Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 1, No. 2. Dec., 1993.

Doherty, P., Lukaszewicz, W. (1994a). Circumscribing Features and Fluents. In *Proc. of the 1st Int. Conf. on Temporal Logic (ICTL94)*, Bonn, Germany, July 11-14, 1994.

Doherty, P., Lukaszewicz, W. (1994b). Circumscribing Features and Fluents: A Fluent Logic for Reasoning about Action and Change. In *Proc. of ISMIS'94,* Charlotte, N.C., USA, Oct. 16-19, 1994.

Drabent, W. (1993). Completeness of SLDNF-Resolution for Non-floundering Queries. In *Proc. Int. Logic Programming Symposium*, 1993. Poster abstract. (Extended version accepted in the *Journal of Logic Programming*.)

Drabent, W. (1993). SLS-resolution Without Floundering. In L. M. Pereira and A. Nerode, eds., *Proc. 2nd Int. Workshop on Logic Programming and Non-Monotonic Reasoning*, pp. 82-98, MIT Press, 1993.

Drabent, W. (1995). What is Failure? An Approach to Constructive Negation. *Acta Informatica*, 32(1), pp. 27-59, 1995.

Driankov, D., Eklund, P. W., Ralescu, A. L. (eds.) (1994a). *Fuzzy Logic and Fuzzy Control*. LNAI, Vol. 833, (book) Springer-Verlag, 1994.

Driankov, D., Fodor, G. (1993). Ontological Real-Time Control. In *Proc. of EUFIT'93*, Aachen, Germany, Sept., 1993.

Driankov, D., Hellendoorn, H. (1995). Chaining of Fuzzy IF-THEN Rules in Mamdani-Controllers. Accepted to the Conf. *FUZZ-IEEE/IFES'95*, Yokohama, Japan, Mar. 20-24, 1995.

Driankov, D., Hellendoorn, H., Reinfrank, M. (1993). *Introduction to Fuzzy Control*. Springer-Verlag, 1993 (a book).

Driankov, D., Lang, J. (1993). Possibilistic Decreasing Persistence. In *Proc. of the Ninth Conf. on Uncertainty in Artificial Intelligence*, Washington DC, USA, July 9-11, 1993.

Driankov, D., Palm, R. (1994). Fuzzy Inputs. In *Proc. of the IEEE'94 Fuzzy Systems Conf.* Orlando, USA, July 26 - Aug. 2, 1994.

Driankov, D., Palm, R., Hellendoorn, H. (1994b). Fuzzy Control with Fuzzy Inputs: the Need for New Rule Semantics. In *Proc. of IEEE'94 Fuzzy Systems Conf.* Orlando, USA, July 26 - Aug. 2, 1994.

Driankov, D., Palm, R., Hellendoorn, H. (1994c). Some Research Directions in Fuzzy Control. Published in *Theoretical Aspects of Fuzzy Control*, eds. R. Yager et al., John Wiley & Sons, 1994.

Eklund, F., Kågedal, A. (1994). Optimization of GAPLog Programs. Presented at *10th Logic Programming Workshop*, Zürich, Switzerland, 1994.

Ekman, A., Strömberg, D., Törne, A. (1994). Exploration of Polygon Environments using Range Data. Accepted for publication in *IEEE Transactions on Systems, Man and Cybernetics.*

Ekman, A., Törne, A., Strömberg, D. (1994). A robot control formalism based on an information quality concept. In *AIAA/NASA Conf. on Intelligent Robots for Factory, Field, Service, and Space,* Houston, Texas, USA.

Eles, P., Kuchcinski, K., Peng, Z., Doboli, A. (1995). Post-synthesis Back-Annotation of Timing Information in Behavioral VHDL. Accepted to *EUROMICRO'95 Conf.* Como, Italy, Sept. 4-7, 1995.

Eles, P., Kuchcinski, K., Peng, Z., Minea, M. (1993). Synthesis of VHDL subprograms and processes in the CAMAD system. In *Proc. Workshop on Design Methodologies for Microelectronics and Signal Processing*, Cracow, Poland.

Eles, P., Kuchcinski, K., Peng, Z., Minea, M. (1994a). Synthesis of VHDL concurrent processes. In *Proc. EURO-DAC'94 with EURO-VHDL'94*, Grenoble, France, Sept. 22-25, 1994.

Eles, P., Peng, Z., Doboli, A. (1994b). VHDL System-Level Specification and Partitioning in a Hardware/Software Co-Synthesis Environment. In *Proc. 3rd Int. Workshop on Hardware/Software Co-Design*, Grenoble, France.

Eriksson, H. (1992a). Metatool support for custom-tailored domain-oriented knowledge acquisition. *Knowledge Acquisition*, 4(4):445–476.

Eriksson, H. (1993). Specification and generation of custom-tailored knowledge-acquisition tools. In *Proc. of the Thirteen Int. Joint Conf. on Artificial Intelligence, IJCAI '93*, Chambéry, France.

Eriksson, H. (1994). Models for knowledge-acquisition tool design. *Knowledge Acquisition,* 6(1):47–74.

Eriksson, H. (1994). A survey of knowledge acquisition techniques and tools and their relationship to software engineering. *Journal of Systems and Software*, 19(1):97–107.

Eriksson, H., Musen, M. A. (1993a). Conceptual models for automatic generation of knowledge-acquisition tools. *Knowledge Engineering Review*, 8(1):27–47.

Eriksson, H., Musen, M. A. (1993b). Metatools for software development and knowledge acquisition. *IEEE Software*, 10(3):23–29.

Eriksson, H., Puerta, A. R., Musen, M. A. (1994). Generation of knowledge-acquisition tools from domain ontologies. *Int. Journal of Human–Computer Studies*, Vol. 41, No. 3, pp. 425-453, 1994.

Eriksson, H., Shahar, Y., Tu, S. W., Puerta, A. R., Musen, M. A. (In press). Task modeling with reusable problem-solving methods, *Artificial Intelligence.*

Fahl, G., Risch, T., Sköld, M. (1993). AMOS – an architecture for active mediators. In *Proc. Workshop on Next Generation Information Technologies and Systems (NGITS 93)*, Haifa, Israel.

Fjällström, P.-O. (1993). Evaluation of a Delaunay-based method for surface approximation. *Computer-Aided Design*, 25(11):711–719.

Fodor, G., Driankov, D. (1994). Fuzzy Logic Control under Violations of Ontological Assumptions. In *Proc. of EUFIT'94*, Aachen, Germany, Sept., 1994.

Forslund, G. (1994). Cooperative or autonomous agents: Considerations for knowledge acquisition and representation. *IEEE Expert*, Special issue on Cooperative Systems for Workgroups.

Forslund, G. (1995). Designing for flexibility: A case study. *Expert Systems*, Feb. issue, Vol. 12, No 1, pp. 27-37.

Fribourg, F., Olsén, H. (1992). A Unifying View of Structural Induction and Computation Induction for Logic Programs. In *Proc. of Logic Program Synthesis and Transformation (LOPSTR 92)*, Springer-Verlag, 1992. Workshops in Computing series.

Fritzson, D., Fritzson, P., Viklund, L., Herber, J. (1994). Object-Oriented Mathematical Modelling – Applied to Machine Elements. *Computers & Structures*, 51(3):241–253, 1994, Pergamon Press.

Fritzson, P. ed. (1993). *Proc. of AADEBUG'93 – the 1:st Int. Workshop on Automated and Algorithmic Debugging*, Linköping, May 3-5, 1993, LNCS 749, Springer-Verlag.

Fritzson, P. ed. (1994). *Proc. of CC '94 – the 5:th Int. Conf. on Compiler Construction*, Vol. 786 of LNCS, Springer-Verlag. Granlund, R. (1994), Developing models for describing actors in a distributed problem solving situation. In Langemyr, L., ed. In *Proc. of the SIMS Simulation Conf.*, pp. 238–243. A previous version was presented at *the 4th IDA Conf. Linköping*, Sweden, 1994.

Fritzson, P., Andersson, N. (1993). Generating Parallel Code from Equations in the ObjectMath Programming Environment. In *Proc of the 2nd Int. Conf. of the Austrian Center for Parallel Computation*, Oct., 1993, LNCS 734, Springer-Verlag.

Fritzson, P., Auguston, M., Shahmehri, N. (1994). Using Assertions in Declarative and Operational Models for Automated Debugging. *Journal of Systems and Software*, 25:223-239, June, 1994.

Fritzson, P., Engelson, V., Viklund, L. (1993). Variant Handling, Inheritance and Composition in the ObjectMath Computer Algebra Environment. In *Proc. of DISCO'93 – Conf. on Design and Implementation of Symbolic Computation Systems*, LNCS 722, Sept., 1993.

Fritzson, P., Viklund, L., Fritzson, D., Herber, J. (1994). ObjectMath – An Environment for High Level Mathematical Modeling and Programming in Scientific Computing. Accepted Sept. 1994 to *IEEE Software*.

Granlund, R. (1994). Developing models for describing actors in a distributed problem solving situation. In Langemyr, L., ed., *Proc. of the SIMS Simulation Conf.*, pp. 238–243. A previous version was presented at *the 4th IDA Conf.*, 1994, Linköping, Sweden.

Goldkuhl, G., Röstlinger, A. (1993). An import aspect of change analysis. Joint elicitation of problems. In *Proc. of IFI 8.2 Conf. "Information systems development – organizational, social and human aspects"*, Noordwijkerhout, The Netherlands, May, 1993.

Gu, X. (1995). RT level testability-driven partitioning. In *Proc. 13th IEEE VLSI Test Symp*, Princeton, USA.

Gu, X., Kuchcinski, K., Peng, Z. (1994). Testability analysis and improvement from VHDL behavioral specifications. In *Proc. EURO-DAC'94 with EURO-VHDL'94*, Grenoble, France, Sept. 19-23, 1994.

Hallberg, J., Peng, Z. (1995). Synthesis under Local Timing Constraints in the CAMAD High-Level Synthesis System. Accepted to *EUROMICRO'95 Conf.*, Como, Italy, Sept. 4-7, 1995.

Haraldsson, **A.** (1993) *Programmering I Lisp*, Studentlitteratur, Lund.

Harmelen, F. v., Mántaras, R. L. de., Malec, J., Treur, J. (1993). Comparing Formal Specification Languages for Complex Reasoning Systems. A chapter in the book: *Formal Specification of Complex Reasoning Systems*. Publ.: Ellis Horwood, eds.: J. Treur, T. Wetter, 1993.

Harrius, J. (1993). From problem solving to solution critiquing by reusing an existing knowledge base. In *Proc. of SCAI*, Stockholm, Sweden, May, 1993.

Heyer, T. (1995). Debugging in GAPLog has been accepted to *AADEBUG '95, 2nd Int. Workshop on Automated and Algorithmic Debugging*, St. Malo, France, May 22-24, 1995.

Hirsch, R. (1994a). Achieving reference: 'A rope of no use'. In Geiger, R. A., ed., *Reference in a Multidisciplinary Perspective Philosophical Object, Cognitive Subject, Intersubjective Process*, Hildesheim: Georg Olms Verlag. To appear.

Hirsch, R. (1994b). Gesture universals: Material and formal constraints. In Sebeok, T. A. and Umiker-Sebeok, J., eds, *Advances in Visual Semiotics*, Univ. of Indiana Press, pp. 475-490, Bloomington, Indiana, USA.

Hirsch, R. (1994c). Interaktive Argumentation: Theorie und Daten 'Auf der Suche nach einer Theorie interaktiver Argumentation'. In Wohlrapp, H., ed., *Wege der Argumentationsforschung*, Stuttgart/ Bad Cannstatt: frommann-holzboog Verlag, pp. 109-119.

Hjerppe, R. (1994). A framework for the description of generalised documents. In Albrechtsen, H. and Oernager, S., eds., *Knowledge Organization and Quality Management. Proc. of the Third Int. ISKO Conf.*, Copenhagen, Denmark, *Advances in Knowledge Organisation*, Vol. 4, pp. 173–180, Frankfurt/Main, Germany, Indeks.

Hoffner, T., Kamkar, M., Fritzson, P. (1995). Evaluation of Program Slicing tools. In Proc. of *AADEBUG '95, 2nd Int. Workshop on Automated and Algorithmic Debugging*, St. Malo, France, May 22-24, 1995.

Hägglund, S. (1993a). A framework for expert critiquing. In Hägglund, S., ed., *Working notes from 11th National Conf. on Artificial Intelligence (AAAI'93) Workshop on Expert Critiquing Systems,* Washington, DC, USA.

Hägglund, S. (1993). Introducing expert critiquing systems. *The Knowledge Engineering Review*, 8(4). Guest Editor's Introduction.

Hägglund, S. (1994). On the need and support for user-centred design. *In Proc. 4th Int. Symp. Human Factors in Organization Design and Management (ODAM'94),* Stockholm, Sweden.

Hägglund, S., Granlund, R. (1994). Computer support for collaborative c3i decision making and training. *SIGOIS Bulletin*, 15(2):5–7. Presented at *the Workshop on Collaborative Realtime Process Management CSCW'94*.

Ingels, P. (1994) Robust parsing with charts and relaxation. *NODALIDA '93 – Proc. of '9:e Nordiska Datalingvistikdagarna*', pp. 123-132, Stockholm, Sweden, June 3-5, 1993.

Johansson, M. (1994a). Identification of quality characteristics for methods in DSS requirements engineering using Quality Function Deployment (QFD). In *Proc. of IRIS'94*, Univ. Press, Oulu, Finland.

Johansson, O. (1994b). Using an extended ER-model based data dictionary to automatically generate product modeling systems. In Litwin, W. and Risch, T., eds., *1st Int. Conf. on Applications of Databases (ADB-94)*, Vol. 819 of LNCS, Vadstena, Sweden, Springer-Verlag.

Jonsson, P., Bäckström, C. (1994a). Complexity Results for State-Variable Planning under Mixed Syntactical and Structural Restrictions. In *Proc. of the 6th Int. Conf. on Artificial Intelligence: Methodology, Systems, Applications, AIMSA'94*, Sofia, Bulgaria, Sept. 21-24, 1994. In *Proc. of the Workshop on Algorithms, Complexity and Commonsense Reasoning at the 11th European Conf. on Artificial Intelligence, ECAI'94*, Amsterdam, The Netherlands, Aug. 8-12, 1994.

Jonsson, P., Bäckström, C. (1994b). Tractable Planning with State Variables by Exploiting Structural Restrictions. In *Proc. of the Twelfth US National Conf. on Artificial Intelligence*, AAAI'94, Seattle, USA, pp. 998-1003, July 30 - Aug. 4, 1994.

Jönsson, A. (1993b). A Method for Development of Dialogue Managers for Natural Language Interface. In *Proc. of the Eleventh National Conf. of Artificial Intelligence (AAAI'93)*, Washington DC, USA, July 13-15, 1993.

Jönsson, A. (1995). A Dialogue Manager for Natural Language Interfaces. In *Proc. of the Pacific Association for Computational Linguistics, Second Conf.*, The Univ. of Queensland, Brisbane, Australia.

Kamkar, M. (1993). A Comparative Classification of Program Slicing Techniques. Accepted 1993 to the *Journal of Systems and Software*.

Kamkar, M., Fritzson, P., Shahmehri, N. (1993). Interprocedural Dynamic Slicing with Applications to Interprocedural Data Flow Testing. In *Proc of the IEEE Conf. on Software Maintenance, CSM-93*, San Diego, USA, Sept., 1993.

Kamkar, M., Shahmehri, N., Fritzson, P. (1993). Three Approaches to Interprocedural Dynamic Slicing. In *Proc. of EUROMICRO'93 (Journal of Microprocessing and Microprogramming)*, Vol. 38, No. 1-5. Published by North-Holland, Sept., 1993.

Kim, E. H-C., Driankov, D. (1993). A Probability-Like Measure for Conditional Statements. In *Proc. of IFSA '93, Int. Fuzzy Systems Association*, Seoul, South Korea, July, 1993.

Kinnunen, K., Timpka, T., Forsum, U. (1995). Decentralised diagnostics of beta haemolytic streptococci group A – introduction of a developmental model for microbiological consultations in primary health care. *Scand. J Prim Health Care 1995*. In press.

Kovordányi, R. (1994). Attention as an inhibiting force in 'creative' reinterpretation of ambiguous images. In *Proc. of the First Int. Summer Institute in Cognitive Science*.

Kuchcinski, K., Drabent, W., Maluszynski, J. (1993). Automatic Diagnosis of VLSI Circuits Using Algorithmic Debugging, In Peter Fritzson, ed., *AADEBUG'93, Proc. of the 1st Int. Workshop on Automated and Algorithmic Debugging*, pp. 350-367, Springer-Verlag, 1993, LNCS 749.

Kågedal, A. (1993). Improvements to Compile-Time Analysis for Ground Prolog. In M. Bruynooghe and J. Penjam, eds., *Programming Language Implementation and Logic Programming 93*, pp. 92-107, Springer-Verlag, 1993, LNCS 724.

Lambrix, P., Rönnquist, R. (1993). Terminological Logic Involving Time and Evolution: A Preliminary Report. In *Proc. of Seventh Int. Symp. on Methodologies for Intelligent Systems ISMIS 93*, pp. 162-171, LNAI 689.

Larsson, M. (1994a). An Engineering Approach to Formal Digital System Design. In T. F. Melham and J. Camilleri, eds., *Higher Order Logic Theorem Proving and Its Applications*, Vol. 859 of LNCS, pp. 300-315, Springer-Verlag, Sept., 1994.

Larsson, M. (1994b). Towards Amalgamating High-Level Synthesis and Proof Systems. *Microprocessing and Microprogramming*, Vol. 40, No. 10-12. Presented as a Short Note on the *EUROMICRO Conf.*, Liverpool, England, 1994.

Lennartsson, B., Shahmehri, N., Bonnier, S. (1994). On exploiting declarative programming and parallel execution in computer based real-time systems. In *Proc. of the IEEE Workshop on Systems Engineering of Computer Based Systems, ECBS'94*, IEEE.

Lindvall, M. (1994a). Object-oriented maintenance. In *Object-Oriented Programming – Issues and Actions*. Working notes from workshop arranged by Oregon Center for Advanced Technology Education, Feb. 28 - Mar 1, 1994.

Lindvall, M. (1994b). Object-oriented modeling in practice. *The Software Practitioner*, 4(2).

Litwin, L., Risch, T. (eds.) (1994). Applications of Databases, *1st Int. Conf., ADB-94*, Vadstena, Sweden, June, 1994, (Proc.) LNCS, Springer-Verlag, ISBN 3-540-58183-9, 1994.

Loborg, P. (1993). Error recovery – an overview. In *LiTH Robotikdagar, Automatiserad tillverkning från högteknologi till tillämpning* (Swedish), Linköping, Sweden. Later version presented at *AAAI 94 Spring Symp. on Detecting and Resolving Errors in Manufacturing Systems*, Stanford, CA, USA, Spring 1994.

Loborg, P., Risch, T., Sköld, M., Törne, A. (1993). Active OO databases in control applications, in microprocessing and microprogramming. In *Proc. of the 19th EUROMICRO Conf.*, Vol. 38, pp. 255–264, Barcelona, Spain, Sept., 1993.

Loborg, P., Törne, A. (1994). Manufacturing control system principles supporting error recovery. In *Working notes of the AAAI 94 Spring Symp.* Series: Detecting and Resolving Errors in Manufacturing Systems, Stanford, CA, USA.

Lukaszewicz, W., Madalinska-Bugaj, E. (1994). Program verification techniques as a tool for reasoning about actions and change. In *Proc. of the 18th German Conf. on Artificial Intelligence*, Vol. 861 of LNAI, pp. 226–236, Springer-Verlag, 1994.

Löwgren, J. (1993a). *Human-computer interaction: What every system developer should know.* Studentlitteratur, Lund.

Löwgren, J. (1993b). Using domain knowledge to support graphical editing. In Salvendy, G. and Smith, M., eds. In *Proc. Fifth Int. Conf. on Human-Computer Interaction (HCI Int. '93)*, Vol. 2, pp. 421–426, Elsevier, Amsterdam, The Netherlands.

Löwgren, J. (1994). Empirical foundations for design rationale as user-interface design support. In Opperman, R., Bagnara, S., and Benyon, D., eds., *Proc. of the 7th European Conf. Cognitive Ergonomics (ECCE 7)*, pp. 305–310, GMD. GMD-Studien No. 233, St. Augustin, Germany.

Löwgren, J., Holmberg, L. (1993). Customizing the SIGCHI curriculum for use in Sweden. *SIGCHI Bulletin*, 25(4):8–11.

Löwgren, J., Laurén, U. (1993). Supporting the use of guidelines and style guides in professional user-interface design. *Interacting with Computers*, 5(4):385–396.

Löwgren, J., Quinn, C., Gasen, J., Gorny, P. (1994). Designing the teaching of HCI: Report from a CHI'94 workshop. *SIGCHI Bulletin*, 26(4):28-31.

Malec, J. (1993). Timed Process Transition Networks. Presented at *Third Conf. on Computer Generated Forces and Behaviour Representation*, Orlando, Florida, USA, Mar. 17-19, 1993.

Malec, J. (1993). On Semantics of Behaviour Languages. In *Proc. of 4th National Conf. on Robotics*. Wroclaw, Poland, Sept. 22-24, 1993.

Malec, J. (1993). On Formal Analysis of Emergent Properties. In *Proc. of EWSP'93, Second European Workshop on Planning*, Vadstena, Sweden, Dec. 9-11, 1993.

Malec, J. (1994a). Autonomous Robot Control Using a Three-Layered Architecture. Presented at the *Scand. Symp. on Robotics*, Stockholm, Sweden — Helsinki, Finland, Oct. 4-6, 1994.

Malec, J. (1994b). Behaviour-Based Autonomous Systems: Towards an Analysis Framework. In *Proc. of the 12th European Meeting on Cybernetics and Systems Research*, Vienna, Austria, Apr. 5-8, 1994.

Malec, J. (1994c). On Implementing Behaviours Using a Three-Layered Architecture. In *Proc. of Intelligent Robotic Systems '94*, Grenoble, France, July 11-15, 1994.

Malec, J. (1994d). A Unified Approach to Intelligent Agency. In *Proc. of ECAI'94, Workshop on Agent Theories, Architectures and Languages*, Amsterdam, The Netherlands, Aug. 8-12, 1994.

Malec, J., Morin, M. (1993). A Pre-Intelligent Driver Information Unit. In *Proc. of IEEE Intelligent Vehicles '93 Symp.*, Tokyo, Japan, July 14-16, 1993.

Malec, J., Morin, M. (1993). An AI-Based Design of a Driver Information Unit. In *Proc. of the Conf. Vehicle Navigation and Information Systems -93*, Ottawa, Canada, Oct. 12-15, 1993.

Malec, J., Morin, M., Palmqvist, U. (1994). Driver Support in Intelligent Autonomous Cruise Control. In *Proc. of IEEE Intelligent Vehicles Symp. '94*, Paris, France, Oct. 24-26, 1994.

Malec, J., Österling, P. (1993). Driver Support System for Traffic Manoeuvres. In the book *Artificial Intelligence in Traffic Engineering* by eds. G. Ambrosino, M. Bielli, M. Boero. Publ.: VSP Int. Science Publ. Zeist, The Netherlands, 1993.

Malec, J., Österling, P. (1994). Situation Assessment in Traffic Scenarios. In *Proc. of Abstract Intelligent Agents '94*, Rome, Italy, Feb. 23-25, 1994.

Maluszynski, J., Bonnier, S., Boye, J., Kågedal, Kluzniak, F., Nilsson, U. (1993). Logic Programs with External Procedures. In K.R. Apt, J.W. de Bakker, and J.J.M.M. Rutten, eds., *Logic Programming Languages, Constraints, Functions, and Objects*, pp. 21-48, The MIT Press, 1993.

Merkel, M. (1993). Semiotics at work: Technical translation and communication in a multilingual corporate environment. In *Proc. of NODALIDA (Nordiska Datalingvistikdagarna)*, Stockholm Univ. Co-author: Arne Larsson, Nokia Telecommunications.

Merkel, M., Nilsson, B., Ahrenberg, L. (1994). A phrase-retrieval system based on recurrence. In *Proc. from the Second Annual Workshop on Very Large Corpora*, Kyoto, Japan, Aug., 1994.

Morin, M., Karlsson, L. (1994). Object Oriented Simulation with WSIM. In *Proc. of the 36th Simulation Conf., SIMS '94*, Stockholm, Sweden, Aug. 17-19, 1994.

Nadjm-Tehrani, S. (1995). Reactive Systems in Physical Environments. Accepted to *AISB-95, 10th Biennial Conf. on AI and Cognitive Science*, Sheffield, England, Apr., 1995.

Nadjm-Tehrani, S., Strömberg, J.-E. (1993). Modelling and Verification of Hybrid Systems: A Case Study. In *Proc. of 4th Int. Conf. on AI, Simulation and Planning in High-Autonomy Systems*, Tucson, USA, Sept. 18-22, 1993.

Nadjm-Tehrani, S., Strömberg, J.-E. (1994). From Physical Modelling to Compositional Models of Hybrid Systems. In *Proc. of the 3rd Int. Symp. on Formal Methods in Real-Time and Fault-Tolerant Systems, FMRTFT'94*, Lübeck, Germany, Sept., 1994, LNCS, No. 863, pp. 583-604, Springer-Verlag.

Nebel, B., Bäckström, C. (1994). On the Computational Complexity of Temporal Projection, Planning and Plan Validation. *Artificial Intelligence*, Vol. 66, pp. 125-160, 1994.

Nilsson, H., Fritzson, P. (1993). Lazy Algorithmic Debugging: Ideas for Practical Implementation. *AADEBUG'93 -1st Int. Workshop on Automated and Algorithmic Debugging*, May, 1993. In LNCS 749, Nov., 1993.

Nilsson, H., Fritzson, P. (1994). Declarative Algorithmic Debugging for Lazy Functional Languages. *The Journal of Functional Languages*, 4 (3), 337-370, July, 1994.

Nilsson, U. (1994). Abstract Interpretation: A Kind of Magic. *Theoretical Computer Science*. To appear, 1995.

Nilsson, U., Maluszynski, J. (1995). *Logic, Programming and Prolog*, (2nd edition). Wiley & Sons. (To appear Aug. 95.)

Nordling, P., Fritzson, P. (1994). Solving ordinary differential equations on parallel computers – applied to dynamic rolling bearings simulation. In Dongarra, J. J., Hansen, P. C., and Wasniewski, J., eds., *PARA'94, Workshop on Parallel Scientific Computing*, UNI-C. Danmarks EDB-Center for Forskning og Uddannelse, Copenhagen, Denmark, June, 1994, LNCS 879, Springer-Verlag.

Nyce, JM., Löwgren, J. (1993). Towards foundational analysis in HCI. In Thomas, P., ed., *Social and Interactional Dimensions of Human-Computer Interfaces*, Cambridge Univ. Press, Cambridge. In Press.

Nyce, JM., Timpka, T. (1993). Work, Knowledge and Argument in Specialist Consultations: Incorporating Tacit Knowledge into System Design and Development. *Med Biol Eng Comp*, 1993;31:S13-7.

Näslund, T. (1992). Injective formative evaluation for enhancing computer support for cooperative staff work. In *Proc. of the Human Interface '92*, pp. 357–364, Kawasaki, Japan.

Näslund, T. (1994). "Usability is extremely important – but it's somebody else's job, I hope". In Kerola, P., Juustila, A., and Järvinen, J., eds. In *Proc. of the 17th Information systems research seminar in Scand. (IRIS17)*, Research paper series A21, pp. 653–667, Univ. of Oulu, Dept. of Information Processing Science, Oulu, Finland.

Näslund, T. (1994). Supporting design communication with explicit representation of evaluation feedback. *Scand. Journal of Information Systems*, 6(2):21–42. Earlier version published in *Proc. of 16th IRIS (Information systems research seminar in Scandinavia)*.

Ohlsson, N. (1994a). Monitoring software projects. In *Proc. of Systems Engineering of Computer Based Systems*, IEEE Press. Position Paper. To appear.

Olhager, J., Rapp, B. (1994). Different approaches to a flow shop problem. *European Journal of Operational Research*, 75(2):469–473, Elsevier Science Publ. B V, Amsterdam, The Netherlands.

Olhager, J., Rapp, B. (1995). Operations research techniques in manufacturing planning and control systems. *Int. Transactions in Operational Research*, 2(1), Pergamon Press Oxford, England.

Orsborn, K. (1994). Applying next generation Object-Oriented DBMS to finite element analysis. In Litwin, W. and Risch, T., eds., *1st Int. Conf. on Applications of Databases (ADB-94)*, Vol. 819 of LNCS, Vadstena, Sweden, Springer-Verlag. LiTH-IDA-R-94-16.

Paakki, J. (1994). Multi-pass evaluation of functional logic programs. In *Proc. of 21th ACM Symp. on Principles of Programming Languages*, The ACM, Portland, Oregon, USA.

Padgham, L., Lambrix, P. (1994). A framework for part-of hierarchies in terminological logics. In Doyle, J., Sandewall, E., and Torasso, P., eds., *Principles of Knowledge Representation and Reasoning: Proc. of the Fourth Int. Conf.*, pp. 485–496, Bonn, Germany, May 24-27, 1994.

Padgham, L., Löwgren, J. (1994). A user interface management approach for object-oriented database applications. *Journal of Systems and Software*, 27(3).

Padgham, L., Nebel, B. (1993). Combining Classification and Non-Monotonic Inheritance Reasoning: A First Step. In *Proc. of Seventh Int. Symp. on Methodologies for Intelligent Systems, ISMIS 93*, LNAI 689, pp. 132-141, Trondheim, Norway.

Padgham, L., Zhang, T. (1993). A terminological logic with defaults: A definition and an application. In *Proc. of 13th Int. Joint Conf. on Artificial Intelligence*, Vol. 2, pp. 662–668, Chambéry, France, July 29 - Sept. 3, 1993.

Peng, Z. (1992). Digital System Simulation with VHDL in a High-Level Synthesis System. Published in *Microprocessing and Microprogramming, the EUROMICRO Journal,* Vol. 35, 1992.

Peng, Z. (1994). Testability-driven high-level synthesis. In *Proc. of the 1994 Int. Conf. on ASIC*. Beijing, China.

Peng, Z. (1995). High-Level Test Synthesis Using Design Transformations. Accepted for the *Second Int. Test Synthesis Workshop*, Santa Barbara, USA, May 8-10, 1995.

Peng, Z., Kuchcinski, K. (1993). An algorithm for partitioning of application specific systems. In *Proc. of the European Conf. on Design Automation EDAC'93*, Paris, France.

Peng, Z., Kuchcinski, K. (1994). Automated transformation of algorithms into register-transfer level implementations. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 13, No. 2.

Peng, Z., Törne, A. (1993). A Petri Net Based Modelling and Synthesis Technique for Real-Time Systems. In *Proc. 5th EUROMICRO Workshop on Real-Time Systems*, Oulu, Finland.

Peng, Z., Törne, A. (1994). Timing analysis and conditional scheduling in a real-time system design environment. In *Proc. 6th EUROMICRO Workshop on Real-Time Systems*, June, 1994, Västerås, Sweden.

Persson, J. (1994). A resource based approach to generalization in the context of GIS. In Harts, J., Ottens, H., and Scholten., H., eds., *EGIS/MARI'94. Fifth European Conf. and Exhibition on Geographical Information Systems*, Vol. 1, pp. 142–150, Paris, France, Utrecht, EGIS Foundation.

Pettersson, M. (1994). RML – A new language and implementation for natural semantics. In *Proc. of PLILP'94, Programming Language Implementation and Logic Programming*. Also in Vol. 844 pp. 117–131 of LNCS, Springer-Verlag, Aug., 1994. Also as report LiTH-IDA-R-94-25, Linköping Univ., Linköping, Sweden.

Ragnemalm, E. L. (1994c). Simulator-based training using a learning companion. In Brouwer-Janse, M. and Harrington, T., eds, *Human-Machine Communication for Educational Systems Design,* Vol. 129, pp. 207–212, Springer-Verlag, Proc. from the 1993 NATO-ASI in Eindhoven, The Netherlands.

Rankin, I. (1993a). Generating argumentative discourse in expert in critiquing systems. In Hägglund, S., ed., *Working notes from AAAI'93 Workshop on Expert Critiquing Systems*, Washington, DC, USA.

Rankin, I. (1993b). Natural language generation in critiquing. *Knowledge Engineering Review*, 8(4).

Rankin, I., Harrius, J. (1993). Critique generation – language games, argumentation and rhetorical aggregates. In Ahrenberg, L., ed., Papers from the *3rd Nordic Conf. on Text Comprehension in Man and Machine*, Linköping, Sweden.

Rapp, B. (1993). Informationshantering på individ- och organisationsnivå (Swedish). In L. Ingelstam, L. Sturesson (eds.) *Brus över landet*, pp. 117-141, Carlssons bokförlag.

Ringström, J., Fritzson, P., Pettersson, M. (1994). Generating an efficient compiler for a data parallel language from a denotational specification. In Fritzson, P. A., ed., *Compiler Construction – 5th Int. Conf., CC '94*, Vol. 786 of LNCS, pp. 248–262, Springer-Verlag.

Rundfelt, R. (1994). *Transnational Accounting; Swede – individual accounts*. MacMillan.

Rundfelt, R. (1994). *Tendenser i börsföretagens årsredovisningar* (Swedish). Stockholms Fondbörs.

Sandahl, K. (1993). Developing knowledge-based systems with active expert participation. In *Proc. of 5th Int. Conf. on Software Engineering and Knowledge Engineering, SEKE-93*, pp. 589–595, San Francisco, CA, USA.

Sandahl, K. (1994). Transferring knowledge from active expert to end-user environment. *Knowledge Acquisition,* (6):1–22.

Sandahl, K., Eriksson, H., Österlund, B. (1994). Tool-based knowledge management and distribution of knowledge to customer's end user in a biochemical domain. In *Proc. of the Int. Symp. on the Management of Industrial and Corporate Knowledge,* pp. 161-175, Compeigne, France, Oct. 26-27, 1994.

Sandewall, E. (1993). Systematic Assessment of Temporal Reasoning Methods for Use in Autonomous Agents. Invited paper to *ISMIS, The Int. Symp. on Methodologies for Intelligent Systems*, Trondheim, Norway, June 15-18, 1993.

Sandewall, E. (1993). The Role of Temporal Reasoning Subsystems in the Architecture of Autonomous Robots. Invited paper to *Conf. on Applications of Artificial Intelligence in Engineering*, Toulouse, France, June 29 - July 1, 1993.

Sandewall, E. (1993). The Range of Applicability of Nonmonotonic Logics for the Inertia Problems. In *Proc. of IJCAI '93, Int. Joint Conf. on Artificial Intelligence*, Inc. Chambéry, France, Aug. 28 - Sept. 9, 1993.

Sandewall, E. (1993). Nonmonotonic Temporal Logics and Autonomous Agents: Each Contributes to the Rigorous Basis for the Other. In *Proc. of the 17. Fachtagung für Künstliche Intelligenz* (invited paper) Berlin, Germany, Sept. 13-16, 1993. Published in O. Herzog et al. (eds): *Grundlagen und Anwendungen der Künstlichen Intelligenz*, Springer-Verlag, 1993.

Sandewall, E. (1994). The Range of Applicability of Some Nonmonotonic Logics for Strict Inertia. In *Journal of Logic and Computation*, 1994.

Sandewall, E. (1994). *Features and Fluents. A Systematic Approach to the Representation of Knowledge About Dynamical Systems*. Book published by Oxford Univ. Press, England, 1994.

Sandewall, E., Shoham, Y. (1994). Nonmonotonic Temporal Reasoning. Published as a chapter in *Handbook of Artificial Intelligence and Logic Programming*, Oxford Univ. Press, England, 1994.

Shahmehri, N., Kamkar, M., Fritzson, P. (1993). Usability Criteria for Automated Debugging Systems. *AADEBUG'93 -1st Int. Workshop on Automated and Algorithmic Debugging*, May, 1993. In LNCS 749, Nov., 1993. Also accepted Mar. 1994 to the *Journal of Systems and Software*.

Shu, H. (1993). The Preferential Semantics of a Multi-Modal Nonmonotonic Logic. In *Proc. of 2nd European Conf. on Symbolic and Quantitative Approaches to Reasoning and Uncertainty, ECSQARU'93*, Granada, Spain, Nov., 1993.

Shu, H. (1993). A Multi-Modal Preferential Logic: Theory and Application. In *Proc. of 5th Int. Conf. on Tools with Artificial Intelligence, TAI'93*, Boston, USA, Nov., 1993.

Sjöberg, C., Timpka, T. (1994). Participatory design in practice: forgetfulness as a design resource. In Bradley, G., ed., *Human Factors in Organizational Design and Management IV*, Elsevier. Amsterdam, The Netherlands.

Sjöberg, C., Timpka, T. (1995). Inside multi-disciplinary design in medical informatics: experiences from the use of an argumentative design method. *Proc. of MedInfo '95*, Amsterdam, The Netherlands, Elsevier, 1995. In press.

Sjölin, M. (1994). A WWW Front End to an OODBMS. In *Proc. of the Second Int. World Wide Web Conf.*, Chicago, USA.

Sjölin, M. (1994). LINCKS – ett verktyg för CSCW. Lars Oestreicher, ed., *Design i MDI – Konferensdokumentation till STIMDI 94*, 1994. (Swedish).

Sköld, M., Risch, T. (1994). Compiling active object-relational rule conditions into partially differentiated relations. Technical report, IDA. Presented at *the Dagstuhl seminar on Active Databases*, Schloss Dagstuhl, Germany, Mar., 1994.

Sparud, J., Nilsson, H. (1995). The Architecture of a Debugger for Lazy Functional Languages. To be presented at *AADEBUG '95, 2nd Int. Workshop on Automated and Algorithmic Debugging*, St. Malo, France, May 22-24, 1995.

Stoy, E., Peng, Z. (1994a). A design representation for hardware/software co-synthesis. In *Proc. EUROMICRO Conf. '94, System Architecture and Integration*, Liverpool, England.

Stoy, E., Peng, Z. (1994b). An integrated modelling technique for hardware/software systems. In *Proc. IEEE Int. Symp. on Circuits and Systems*, London, England, May, 1994.

Strömberg, J.-E., Nadjm-Tehrani, S. (1994). On Discrete and Hybrid Representation of Hybrid Systems. In *Proc. of the European Simulation Multiconference, ESM '94*, Barcelona, Spain. June, 1994.

Strömberg, J.-E., Söderman, U., Top, J. (1994). Conceptual Modelling of Hybrid Systems. In *Proc. of the European Simulation Conf., ESM '94*, Barcelona, Spain, June, 1994.

Strömbäck, L. (1994). Achieving flexibility in unification formalisms. In *Proc. 15th Int. Conf. on Computational Linguistics (Coling'94)*, Vol. II, pp. 842–846, Kyoto, Japan, Aug., 1994.

Strömbäck, L. (1995a). Using Logic Programming Results while Developing a Flexible Unification-Based Grammar Formalism. *Proc. of the 15th Int. Workshop on Natural Language Understanding and Logic Programming*, Lissabon, Portugal, May, 1995.

Strömbäck, L. (1995b). User-Defined Nonmonotonicity in Unification-Based Formalisms. *Proc. of the 1995 Conference of the Association for Computer Linguistics*, Cambridge, Massachusetts, June, 1995.

Svenberg, S. (1994). Representing conceptual and linguistic knowledge for multi-lingual generation in a technical domain. In *Proc. of 7th Int. Generation Workshop*, pp. 245–248, Kennebunkport, Maine, USA, June, 1994.

Söderman, U., Strömberg, J-E. (1995). Switched Bond Graphs: Towards Systematic Composition of Computational Models. Accepted to *The Int. Conf. on Bond Graph Modelling and Simulation, ICBGM95*, Las Vegas, USA, Jan., 1995.

Söderman, U., Top, J., Strömberg, J-E. (1993). The Conceptual Side of Mode Switching. In *Proc. of the IEEE/SMC '93 Conf., IEEE Int. Conf. on Systems, Man and Cybernetics*, Le Touquet, France, Oct. 17-20, 1993.

Thomas, P., Löwgren, J. (1994). From theory to action: Teaching research and practice in technology. In *Proc. of the 11th Int. Conf. Technology and Education (ICTE'94)*, Vol. 2, pp. 984–986, London, England.

Timpka, T. (1993). Teknik och lärande i serviceverksamheter: Datorstöd för primärvården (Swedish). In Lennerlöf L (red.) *Människor, datateknik, arbetsliv*, Stockholm, Allmänna Förlaget, 1993. (Swedish).

Timpka, T. (1994a). The health care crisis: End or beginning of the hay-day of medical decision support systems. Editorial. *Meth Inf Med*, 1994;33:332-5.

Timpka, T. (1994b). Organizational learning in the continuous development of health care: making use of information technology to increase the total service quality. In Bradley, G., ed., *Human Factors in Organizational Design and Management IV*, Elsevier. Amsterdam, The Netherlands.

Timpka, T. (1995). Situated clinical cognition. Editorial. *Artificial Intelligence in Medicine.* 1995;7. In press.

Timpka, T., Arborelius, E. (1994). The primary health care nurse's dilemmas: a study of knowledge use and need during telephone consultations. In Smith JP. ed., *Readings in advanced nursing*, pp. 127–142, Oxford: Blackwells Scientific.

Timpka, T., Hensing, G., Alexanderson, K. (1995). Dilemmas in sickness certification among Swedish physicians. *Euro J Pub Health 1995*. In press.

Timpka, T., Johansson, M. (1994). The need for requirements engineering in the development of clinical decision support systems: a qualitative study. *Meth Inf Med*, (33):227–233.

Timpka, T., Marmolin, H. (1994). Expertext systems at ambulatory clinics. In *Proc. of World Congress on Expert Systems '94*, New York, Pergamon Press. (Invited paper).

Timpka, T., Marmolin, H. (1995). Beyond computer-based clinical remainders: improvement of the total service quality by small-group based organizational learning in primary care. *Proc. of MedInfo '95*, Amsterdam, The Netherlands, Elsevier, 1995. In press.

Timpka, T., Nyce, JM., Sjöberg, C., Johansson, M. (1993). Action Design: From Modeling to Support of the Software Process. In Blum BI, Chang L eds. *Proc. of Software Engineering and Knowledge Engineering '93*, San Francisco: IEEE Press, 1993.

Timpka, T., Nyce, JM., Sjöberg, C., Rehnwall, H., Herbert, I. (1993). Bar code technology in health care: Using a business model for study of technology application and dissemination. *Proc. of MIE '93*, Elsevier, Amsterdam, 1993.

Timpka, T., Rauch, E., Nyce, JM. (1994). Towards productive knowledge based systems in medicine: a methods perspective. *Artificial Intelligence in Medicine*, 1994;6:501-19.

Timpka, T., Sjöberg, C. (1994). Voices of design: the dynamics of participatory information system design. In *Proc. of Participatory Design '94*, Chappel Hill, CPSR.

Timpka, T., Sjöberg, C., Svensson, B. (1995). The pragmatics of clinical hypermedia: experiences from five years of participatory design in the MEDEA project. *Comp Meth Prog Biomed 1995*. In press.

Timpka T., Svensson, B., Molin, B. (1995). Development of community nursing: analysis of the central services and practice dilemmas. *Int J Nurs Stud 1995*. In press.

Wærn, Y., Hägglund, S., Ramberg, R., Rankin, I., Harrius, J. (1995). Computational advice and explanations – Behavioural and computational aspects. In *Proc. of INTERACT'95*, Lillehammer, Norway.

Wirén, M. (1994). Minimal change and bounded incremental parsing. In *Proc. of the 15th Int. Conf. on Computational Linguistics*, pp. 461–467, Kyoto, Japan, Aug., 1994.

Wirén, M., Rönnquist, R. (1993). Fully Incremental Parsing. In *Proc. of the Third Int. Workshop on Parsing Technologies.* Tilburg, The Netherlands and Durbuy, Belgium, Aug. 10-13, 1993.

Yi, C.-H. (1995). Towards the Assessment of Logics for Concurrent Actions. Accepted to *AAAI'95 Spring Symp.: Extending Theories of Action*, Stanford Univ., USA, Mar., 1995.

E.6 Departmental Reports 1993–94

LiTH-IDA-R-93-01	**Roland Bol, Lars Degerstedt:** The Underlying Search for Magic Templates and Tabulation. 1993. In *Proc. of the Int. Conf. on Logic Programming.* Budapest, The MIT Press, Hungary, June, 1993.
LiTH-IDA-R-93-02	**Jukka Paakki:** Multi-Pass Evaluation of Functional Logic Programs. In *Proc. of 21th ACM Symp. on Principles of Programming Languages*. The ACM. Portland, Oregon, USA.
LiTH-IDA-R-93-03	**Brant A Cheikes:** Methodological issues in the design of intelligent and cooperative information systems. In *Proc. of the Int. Conf. on Intelligent and Cooperative Inf. Systems (ICICIS-93)*, pp. 3–12, Rotterdam, The Netherlands. IEEE, Comp. Society Press, 1993.
LiTH-IDA-R-93-04	**Mikael Pettersson:** Main-Memory Linear Hashing-Some Enhancements of Larson's Algorithm.
LiTH-IDA-R-93-05	**Göran Goldkuhl:** Contextual Activity Modelling of Information Systems. In *Proc. of the Third Int. Working Conf. on Dynamic Modelling of Inf. Systems.* Noordwijkerhout, Holland, June 9-10, 1992.
LiTH-IDA-R-93-06	**Göran Goldkuhl:** Att Förändra Informationssystem-En Verksamhetsinriktad Grundsyn. I proc. från *Linköpings Informationssystem Seminarium 1992-LISS 92*, Linköping, May 21, 1992. (Swedish).
LiTH-IDA-R-93-07	**Annie Röstlinger:** Några Problem vid Utvärdering och Förändring av Datorstödda Kommunala Verksamheter. I proc. från *Linköpings Informations-system Seminarium 1992-LISS 92,* Linköping, May 21, 1992. (Swedish).
LiTH-IDA-R-93-08	**Erik Sandewall:** The Range of Applicability of Nonmonotonic Logics for the Inertia Problem. In *Proc. of IJCAI'93, Int. Joint Conf. on Artificial Intelligence 1993.* Chambéry, France, Aug. 28 - Sept. 3, 1993.
LiTH-IDA-R-93-09	**Jonas Löwgren:** Om teknologers inställning till sina studier. (Swedish).
LiTH-IDA-R-93-10	**Patrick Lambrix, Ralph Rönnquist:** Terminological Logic Involving Time and Evolution: A Preliminary Report. In *Proc. of Seventh Int. Symp. on Methodologies for Intelligent Systems- ISMIS 93*, pp. 162-171, LNAI 689.
LiTH-IDA-R-93-11	**Zebo Peng, Anders Törne:** A Petri Net Based Modelling and Synthesis Technique for Real-Time Systems. In *Proc. 5th EUROMICRO Workshop on Real-Time Systems*. Oulu, Finland.
LiTH-IDA-R-93-12	**Peter Fritzson, Mikhail Auguston, Nahid Shahmehri:** Using Assertion in Declarative Models for Automated Debugging.
LiTH-IDA-R-93-13	**Gustav Fahl, Tore Risch, Martin Sköld:** AMOS – An Architecture for Active Mediators. In *Proc. of Int. Workshop on Next Generation Inf. Technologies and Systems (NGITS'93).* Haifa, Israel, June, 1993.
LiTH-IDA-R-93-14	**Jacek Malec, Magnus Morin:** A Pre-intelligent Driver Information Unit. In *Proc. of the IEEE Intelligent Vehicles'90 Symp.*, Tokyo, Japan, July 14 -16, 1993.
LiTH-IDA-R-93-15	**Kjell Orsborn:** Modeling of Product Data Using an Extensible O-O Query Language.
LiTH-IDA-R-93-16	**Arne Jönsson:** A Method for Development of Dialogue Managers for Natural Language Interfaces. In *Proc. of the AAAI-93 Conf.*, Washington DC, USA, 13-15 July 1993.
LiTH-IDA-R-93-17	**Göran Goldkuhl, Stefan Cronholm, Christian Krysander:** Adaptation of CASE Tools to Different Systems Development Methods. In *Proc. of the 15th Inf. Systems Research Seminar in Scand. (IRIS),* Larkollen, Norway, 9-12 Aug., 1992.
LiTH-IDA-R-93-18	**Göran Goldkuhl:** On the Relations between Information Systems and Organizational Activities: Integration and Separation as Thought Models. In *Proc. of the 15th Inf. Systems Research Seminar in Scand. (IRIS)*, Larkollen, Norway, Aug. 9-12, 1992.
LiTH-IDA-R-93-19	**Göran Goldkuhl, Karin Pettersson, Owen Eriksson:** Hur studera realisering och konsekvenser av strategibaserade informationssystemarkitekturer? (Swedish).

LiTH-IDA-R-93-20	**Johan Boye, Jukka Paakki, Jan Maluszynski:** Dependency-Based Groundness Analysis of Functional Logic Programs. A short version accepted to *PLILP'93* under the title *Avoiding dynamic delays in functional logic programs.* Tallinn, Estonia, Aug. 26-28, 1993.
LiTH-IDA-R-93-21	**Jacek Malec, Per Österling:** Driver Support System for Traffic Manoeuvres. In *Proc. of the Artificial Intelligence in Traffic Engineering*. Eds.: G Ambrosino, M Bielli, M Boero, VSP Int. Sc. Publ. Zeist, The Netherlands, 1993.
LiTH-IDA-R-93-22	**Petru Eles, Krzysztof Kuchcinski, Zebo Peng, Marius Minea:** Two Methods for Synthesizing VHDL Concurrent Processes.
LiTH-IDA-R-93-23	**Marius Minea:** A VHDL Compiler for a High-Level Synthesis System.
LiTH-IDA-R-93-24	**Thomas Bennet, Björn Helander, Jan Ollinen, Jaime Villegas:** Business Modelling.
LiTH-IDA-R-93-25	**Tor Berggrund, Kristina Larsen, Anna Moberg, Bengt Savén:** Business Modelling – integration av metoder för affärs- och systemutveckling. (Swedish).
LiTH-IDA-R-93-26	**Stefan Blom, Lars Mattsson, Mehran Noghabai:** Business Modelling – en studie av förändringsprocesser från strategisk planering till systemutveckling. (Swedish).
LiTH-IDA-R-93-27	**Erik Sandewall:** Systematic Assessment of Temporal Reasoning Methods for Use in Autonomous Agents. In *Proc. of the Int. Symp. on Methodologies for Intelligent Systems (ISMIS)'93,* Trondheim, Norway, June 15-18, 1993.
LiTH-IDA-R-93-28	**Peter Loborg, Tore Risch, Martin Sköld, Anders Törne:** Active Object Oriented Databases in Control Applications. In *Proc. of the 19th EURO-MICRO Conf.*, Vol. 38, pp. 255–264. Barcelona, Spain.
LiTH-IDA-R-93-29	**Brant A Cheikes, Eva L Ragnemalm:** Simulator-Based Training-Support Tools for Process-Control Operators. In *Proc. of NATO Workshop on Natural Dialogue and Interactive Student Modeling*, Varenna, Italy, Oct. 12-16, 1992.
LiTH-IDA-R-93-30	**Erik Sandewall:** Nonmonotonic Temporal Logics and Autonomous Agents: each Contributes to the Rigorous Basic for the other. In *Proc. of the 17. Fachtagung für Künstliche Intelligenz* (invited paper) Berlin, Germany, Sept. 13 -16, 1993. Published in O. Herzog et al. (eds.): *Grundlagen und Anwendungen der Künstlichen Intelligenz*, Springer Verlag, 1993.
LiTH-IDA-R-93-31	**Krzysztof Kuchcinski, Zebo Peng:** An Integrated Design Methodology for Digital Systems. Presented at the *Workshop on Design Methodologies for Microelectronics and Signal Processing*, Gliwice-Cracow, Poland, Oct. 20-23, 1993.
LiTH-IDA-R-93-32	**Frank van Harmelen, Ramon López de Mántaras, Jacek Malec, Jan Truer:** Comparing Formal Specification Languages for Complex Reasoning Systems. A chapter in the book: *Formal Specification of Complex Reasoning Systems*. Publ.: Ellis Horwood, 1993, Eds.: J Treur and The Wetter, 1993.
LiTH-IDA-R-93-33	**Mats Wirén, Ralph Rönnquist:** Fully Incremental Parsing.In *Proc. of the Third Int. Workshop on Parsing Technologies.* Tilburg, The Netherlands and Durbuy, Belgium, Aug. 10-13, 1993.
LiTH-IDA-R-93-34	**Christer Bäckström, Bernhard Nebel:** Complexity Results for SAS^+ Planning. Accepted for publication in the journal *Computational Intelligence*, Vol. 12, 1996.
LiTH-IDA-R-93-35	**Jacek Malec, Magnus Morin:** An AI-based Design of a Driver Information Unit. In *Proc. of Vehicle Navigation and Inf. Systems -93,* Ottawa, Canada, Oct. 12-15, 1993.
LiTH-IDA-R-93-36	**Lars Poignant:** Uppskattning av antalet arbetsplatser i Stockholmsområdet som kan avskiljas med hjälp av informationsteknologi. (Swedish).
LiTH-IDA-R-93-37	**Jörgen Andersson, Björn Helander:** Ekonomistyrning för små företag. (Swedish).
LiTH-IDA-R-93-38	**Wlodzimierz Drabent:** On Completeness of SLDNF-Resolution. This material was presented at a poster session at *Int. Logic Programming Symp.,* Vancouver, Canada, Oct., 1993.

LiTH-IDA-R-93-39	**Hua Shu:** The Preferential Semantics of a Multi-Modal Nonmonotonic Logic. In *Proc. of the 2nd European Conf. on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQRU'93)*, Granada, Spain, Nov., 1993.
LiTH-IDA-R-93-40	**Anna Brolin, Stefan Cronholm, Dan Fristedt:** Business Modelling – en empirisk studie. (Swedish).
LiTH-IDA-R-93-41	**Anna Brolin, Stefan Cronholm, Dan Fristedt:** Business Modelling.
LiTH-IDA-R-93-42	**Göran Goldkuhl, Stefan Cronholm:** Customizable CASE Environments: a Framework for Design and Evaluation. In *Proc. of COPE IT'93/NordDATA*, Copenhagen, June 14-16, 1993.
LiTH-IDA-R-93-43	**Annie Röstlinger, Karin Pettersson:** Utvärdering och Förändring av Datorstödda Kommunala Verksamheter. (Swedish).
LiTH-IDA-R-93-44	**Annie Röstlinger:** Verksamhetsinriktad Värdering av Datasystem inför Förändring. Metod och Erfarenheter. Föredrag presenterat på *Sundsvall 42*, Oct. 19-21, 1993. (Swedish).
LiTH-IDA-R-93-45	**Jacek Malec:** On Formal Analysis of Emergent Properties. In *Proc. of the Second European Workshop on Planning (EWSP'93)*, Vadstena, Sweden, Dec. 9-11, 1993.
LiTH-IDA-R-93-46	**Karin Pettersson, Göran Goldkuhl:** IRM på Byggföretag – en Fallstudie om Strukturering av Informationssystem. Föredrag presenterat på *VITS Höstseminarium*, Linköping Univ., Sept. 28-29, 1993. (Swedish).
LiTH-IDA-R-93-47	**Karin Pettersson, Göran Goldkuhl:** VBS på Pappersbruk – en Fallstudie om Strukturering av Informationssystem. (Swedish).
LiTH-IDA-R-93-48	**Jan-Erik Strömberg, Simin Nadjm-Tehrani:** Towards Reliable Models of Hybrid Systems.
LiTH-IDA-R-94-01	**Zebo Peng, Krzysztof Kuchcinski:** An Algorithm for Partitioning of Application Specific Systems. Published in *Proc. of the European Conf. on Design Automation EDAC'93*, Paris, France, Feb. 22-25, 1993.
LiTH-IDA-R-94-02	**Zebo Peng:** Digital System Simulation with VHDL in a High-Level Synthesis System. Published in *Microprocessing and Microprogramming, the EURO-MICRO Journal*, Vol. 35, 1992.
LiTH-IDA-R-94-03	**Jacek Malec:** Behaviour-Based Autonomous Systems: Towards an Analysis Framework. In *Proc. of the 12th European Meeting on Cybernetics and Systems Research*, Vienna, Austria, Apr. 5-8, 1994.
LiTH-IDA-R-94-04	**Göran Goldkuhl:** Välgrundad Metodutveckling. Presenterat på *VITS Höstseminarium*, Sept. 28-29, 1993, Institutionen för Datavetenskap, Linköping Univ. (Swedish).
LiTH-IDA-R-94-05	**Karin Pettersson:** Consequences of an IRM-based System Development – Experiences from a Case Study. In *Proc. of* the *16th IRIS*, pp. 507-521. Dept. of Comp. Sc., Univ. of Copenhagen, Copenhagen, Denmark, Aug. 7-10, 1993.
LiTH-IDA-R-94-06	**Ulf Söderman, Jan Top, Jan-Erik Strömberg:** Modelling Physical Systems with Changing Structure
LiTH-IDA-R-94-07	**Ulf Söderman, Jan Top, Jan-Erik Strömberg:** The Conceptual Side of Mode Switching. In *Proc. of the 1993, IEEE Int. Conf. on Systems, Man and Cybernetics.* Le Touquet, France, Oct., 1993.
LiTH-IDA-R-94-08	**Karin Pettersson, Göran Goldkuhl:** A Comparison between two Strategies for Information Systems Architectures. In *Proc. of the Sundsvall 42*, Sundsvall, Sweden, Oct. 19-21, 1993.
LiTH-IDA-R-94-09	**Olof Johansson:** Prototypbaserad utveckling av ett produktmodellorienterad konstruktionsstödsystem med objektorienterad datamodelleringsteknik. (Swedish).
LiTH-IDA-R-94-10	**Martin Sköld, Tore Risch:** Compiling Active Object-Relational Rule Conditions into Partially Differentiated Relations. Presented at the Dagstuhl-*Seminar on Active Databases*, Schloss, Dagstuhl, Germany, Mar., 1994.
LiTH-IDA-R-94-11	**Peter Fritzson:** *Proc. of the Poster Session of CC'94 – Int. Conf. om Compiler Construction.*
LiTH-IDA-R-94-12	**Lena Strömbäck:** FLUF: A Flexible Unification Formalism – the Idea. The short version: Achieving flexibility in unification formalisms also in *Proc. of the 15th Int. Conf. on Computational Linguistics.* Kyoto, Japan, Aug. 5-9, 1994.

LiTH-IDA-R-94-13	**Lena Strömbäck:** FLUF: A Flexible Unification Formalism – Syntax and Semantics.
LiTH-IDA-R-94-14	**Erik Sandewall:** The Range of Applicability of some Nonmonotonic Logics for Strict Inertia. Published in the *Journal of Logic and Computation*, 1994.
LiTH-IDA-R-94-15	**Erik Sandewall:** Features and Fluents. A Systematic Approach to the Representation of Knowledge about Dynamical Systems. Final Review Version. Published by Oxford Univ. Press, 1994.
LiTH-IDA-R-94-16	**Kjell Orsborn:** Applying Next Generation Object-Oriented DBMS to Finite Element Analysis. LNCS, Vol. 819, Springer Verlag, 1994.
LiTH-IDA-R-94-17	**Bengt Savén:** Beslutsstöd och simulering i verkstadsföretag – En sammanfattning av två enkätstudier. (Swedish).
LiTH-IDA-R-94-18	**Karin Pettersson:** IRM på försäkringsbolag – en fallstudie om strukturering av informationssystem. (Swedish).
LiTH-IDA-R-94-19	**Karin Pettersson:** VBS i industriföretag – en fallstudie om strukturering av informationssystem. (Swedish).
LiTH-IDA-R-94-20	**Christer Bäckström:** Finding Least Constrained Plans and Optimal Parallel Executions is Harder than we thought. In C. Bäckström and E. Sandewall (eds.), *Current Trends in AI Planning: EWSP'93---2nd European Workshop on Planning*, Vadstena, Sweden, Dec., 1993. IOS Press. Also presented at the workshop on *Algorithms, Complexity and Commonsense Reasoning at the 11th European Conf. on Artificial Intelligence (Ecai'94.,* Amsterdam, The Netherlands, Aug., 1994.
LiTH-IDA-R-94-21	**Jacek Malec:** On Implementing Behaviours Using a three-layered Architecture. In *Proc. of Intelligent Robotic Systems'94,* Grenoble, France, July 11-15, 1994.
LiTH-IDA-R-94-22	**Christer Bäckström:** Executing Parallel Plans Faster by Adding Actions. *Proc. of the 11th European Conf. on Artificial Intelligence, ECAI'94*, pp. 615–619, Amsterdam, The Netherlands, Wiley.
LiTH-IDA-R-94-23	**Jacek Malec:** A Unified Approach to Intelligent Agency. In *Proc. of the ECAI'94 Workshop on Agent Theories, Architectures and Languages,* Amsterdam, The Netherlands, Aug. 8-12, 1994.
LiTH-IDA-R-94-24	**Annie Röstlinger:** STYRKEANALYS Ett arbetssätt för att tillvarata positiva aspekter i verksamheten. Föredrag presenterat på *VITS Höstseminarium.* Sept. 28-29, 1993, Institutionen för datavetenskap, Linköping Univ. (Swedish).
LiTH-IDA-R-94-25	**Mikael Pettersson:** RML – A New Language and Implementation for Natural Semantics. In *Proc. of PLILP'94, Programming Language Implementation and Logic Programming*. Also in Vol. 844 pp. 117–131 of LNCS.
LiTH-IDA-R-94-26	**Jacek Malec, Magnus Morin, Ulf Palmqvist:** Driver Support in Intelligent Autonomous Cruise Control. In *Proc. of IEEE Intelligent Vehicles Symp. '94*. Paris, France. Oct. 24-26, 1994.
LiTH-IDA-R-94-27	**Xinli Gu, Krzysztof Kuchcinski, Zebo Peng:** Testability Analysis and Improvement from VHDL Behavioral Specifications. In *Proc. EURO-DAC'94 with EURO-VHDL'94,* Grenoble, France, Sept. 19-23, 1994.
LiTH-IDA-R-94-28	**Lars Karlsson:** Specification and Synthesis of Plans Using the Features and Fluents Framework.
LiTH-IDA-R-94-29	**Roger Germundsson, Jojan Gunnarsson, Arne Jansson, Petter Krus, Magnus Morin, Simin Nadjm-Tehrani, Jonas Plantin, Magnus Sethson, Jan-Erik Strömberg:** Complex Hybrid Systems I – A Study of available Tools and Specification of planned Work
LiTH-IDA-R-94-30	**Lin Padgham, Patrick Lambrix:** A Framework for Part-of Hierarchies in Terminological Logics. Accepted to *Principles of Knowledge Representation and Reasoning, Proc. of the Fourth Int. Conf.* Eds. Doyle, Sandewall, Torasso, pp.:485-496 Bonn, Germany, May 24-27, 1994.
LiTH-IDA-R-94-31	**Lin Padgham, Bernhard Nebel:** Combining Classification and Nonmonotonic Inheritance Reasoning: A First Step. Accepted to *Proc. of the Seventh Int. Symp. on Methodologies for Intelligent Systems,* LNAI 689, pp. 132-141, Trondheim, Norway.

LiTH-IDA-R-94-32	**Lin Padgham, Tingting Zhang:** A Terminological Logic with Defaults: A Definition and an Application. Accepted to *Proc. of the 13th Int. Joint Conf. on Artificial Intelligence*, Vol. 2, pp. 662-668 Chambéry, France, July 29 - Sept. 3, 1993.
LiTH-IDA-R-94-33	**Lin Padgham, Jonas Löwgrem:** A User Interface Management Approach for Object-Oriented Database Applications. Accepted to *Journal of Systems and Software*.
LiTH-IDA-R-94-34	**Kristian Sandahl:** Transferring Knowledge from Active Expert to End-user Environment. Accepted to *Knowledge Acquisition*, Vol. 6, No. 1, pp. 1-22.
LiTH-IDA-R-94-35	**Kristian Sandahl, Henrik Eriksson, Bengt Österlund:** Tool-Based Knowledge Management and Distribution of Knowledge to Customers' End-User in a Biochemical Domain. In *Proc. of the Int. Symp. on the Management of Industrial and Corporate Knowledge.* Compiègne, France, Oct. 26-27, 1994.
LiTH-IDA-R-94-36	**Petru Eles, Krzysztof Kuchcinski, Zebo Peng, Alexa Doboli:** Back-Annotation of VHDL Behavioral Models for Postsynthesis Simulation.
LiTH-IDA-R-94-37	**Petru Eles, Krzysztof Kuchcinski, Zebo Peng, Alexa Doboli:** Specification of Timing Constraints in VHDL for High-Level Synthesis.
LiTH-IDA-R-94-38	**Ulf Söderman, Jan-Erik Strömberg:** Switched Bond Graphs: Towards Systematic Composition of Computational Models. In *Proc. of The Int. Conf. on Bond Graph Modeling and Simulation ICBGM95,* Las Vegas, USA, Jan., 1995.
LiTH-IDA-R-94-39	**Ulf Söderman:** Conceptual Modelling of Physical Systems – a Frame of Reference.
LiTH-IDA-R-94-40	**Jan-Erik Strömberg, Ulf Söderman, Jan Top:** Conceptual Modelling of Hybrid Systems. In *Proc. of The European Simulation Conf. ESM'94*, Barcelona, Spain, June, 1994.
LiTH-IDA-R-94-41	**Thomas Bennet, Jan Olhager, Birger Rapp:** Datorisering av MPS-system i svenska företag 1987-1993. Denna rapport är inskickad för publicering i tidningen Verkstäderna i början av 1995. (Swedish). En något annorlunda version är också publicerad som proc. från *The IFIP WG5.7 Working Conf. on Evaluation of Production Management Methods Gramado*, Brazil, Mar. 21-24, 1994.
LiTH-IDA-R-94-42	**Patrick Doherty, Witold Lukaszewicz, Andrzej Szalas:** Computing Circumscription Revisited: A Reduction Algorithm.
LiTH-IDA-R-94-43	**Patrick Doherty:** Notes on PMON Circumscription.
LiTH-IDA-R-94-44	**Pär Carlshamre:** A Field Study in Usability Engineering: Bringing in the Technical Communicators.
LiTH-IDA-R-94-45	**Laurent Fribourg, Hans Olsén:** Direct, Dual and Contrapositive Proofs by Indication.
LiTH-IDA-R-94-46	**Staffan Flodin:** An Incremental Query Compiler with Resolution of Late Binding.
LiTH-IDA-R-94-47	**Karin Axelsson:** Informationssystemstrategiers betydelse vid förändring av informationssystem. Presenterat på *Sundsvall 42*, Sundsvall, Oct. 18-20, 1994. (Swedish).
LiTH-IDA-R-94-48	**Karin Pettersson, Owen Eriksson:** VBS på Skogsbolag – en fall studie om strukturering av informationssystem. (Swedish).
LiTH-IDA-R-94-49	**Karin Pettersson:** Några problem vid verksamhetsbaserad systemstrukturering. Erfarenheter från tre fallstudier. Presenterat på *LISS 94 (Linköpings InformationsSystem Seminarium 1994),* Apr. 20, 1994, Linköping Univ., Linköping. (Swedish).
LiTH-IDA-R-94-50	**Karin Pettersson, Annie Röstlinger, Anna Brolin:** Datadriven utveckling i kommun – en fallstudie om strukturering av informationssystem. (Swedish).
LiTH-IDA-R-94-51	**Stefan Cronholm, Göran Goldkuhl:** Meanings and Motives of Method Customizations in CASE Environments – Observations and categorizations from an empirical study. Accepted to *the 5th European Workshop on next Generation of CASE Tools,* Utrecht, The Netherlands, June 6-7, 1994.
LiTH-IDA-R-94-52	**Göran Goldkuhl, Dan Fristedt:** Metodanalys – en beskrivning av metametoden SIMM. Föredrag presenterat på *VITS Höstseminarium 1994*, Linköping, Nov. 23-24, 1994. (Swedish).

LiTH-IDA-R-94-53	**Göran Goldkuhl:** Några problem vid datadriven strukturering av informationssystem. I proc. från *LISS 94 (Linköpings InformationsSystem Seminarium 1994),* Apr. 20, 1994, Linköping Univ., Linköping. (Swedish).
LiTH-IDA-R-94-54	**Annie Röstlinger:** Att tillvarata styrkan i gjorda IT-investeringar. Föredrag presenterat på *Sundsvall 42,* Sundsvall, Oct. 18-20, 1994. (Swedish).
LiTH-IDA-R-94-55	**Lisbeth Linge et al.** (eds.): Department of Computer and Information Science Activity Report 1993-94.
LiTH-IDA-R-95-01	**Tommy Hoffner:** Evaluation and Comparison of Program Slicing Tools.
LiTH-IDA-R-95-02	**Pavlos Pappas:** Epistemic Entrenchment and the Possible Models Approach.
LiTH-IDA-R-95-03	**Christer Bäckström:** Expressive Equivalence of Planning Formalisms. An abridged version of this report will appear in *Artificial Intelligence*, Special Issue on Planning and Scheduling.
LiTH-IDA-R-95-04	**Ulf Söderman, Jan-Erik Strömberg, Jan Top:** Mode Invariant Modelling.
LiTH-IDA-R-95-05	**Jakob Axelsson:** Analysis and Improvement of Task Schedulability in Hardware/Software Codesign. This is an extended version of a paper accepted to the *7th Euromicro Workshop on Real Time Systems*, Odense, Denmark, June 14-16, 1995.
LiTH-IDA-R-95-06	**Jan-Erik Strömberg, Ulf Söderman:** Modular Modelling of Mode-switching Systems.
LiTH-IDA-R-95-07	**Ulf Söderman, Jan-Erik Strömberg:** Switched Bond Graphs: multiport switches, mathematical characterization and systematic composition of computational models.
LiTH-IDA-R-95-08	**Mikael Lindvall, Kristian Sandahl:** Exempel på spårbarhet i en industriell tillämpning av Objectory (Industrial series).
LiTH-IDA-R-95-09	**Göran Goldkuhl:** Information as Action and Communication. Also appearing in Dahlbom B (Ed.1995) *The Infological Equation. Essays in Honour of Börje Langefors.* Gothenburg Studies in Information Systems, Gothenburg University.
LiTH-IDA-R-95-10	**Staffan Flodin, Tore Risch:** Processing Object-Oriented Queries with Invertible Late Bound Functions. To be presented at *21st International Conference on Very Large Data Bases*. Sept. 11-15, 1995, Zurich, Switzerland.
LiTH-IDA-R-95-11	**Gustav Fahl, Tore Risch:** Query Processing over Object Views of Relational Data.
LiTH-IDA-R-95-12	**Christer Bäckström, Peter Jonsson:** Planning with Abstraction Hierarchies can be Exponentially Less Efficient. To appear in *Proc. 14th Int.Joint Conf. on Artificial Intelligence (IJCAI'95),* Montreal, PQ, Canada, Aug., 1995.
LiTH-IDA-R-95-13	**Anders G Nilsson:** Utveckling av metoder för systemarbete - ett historiskt perspektiv. Föredrag presenterat på *VITS Höstseminarium*, Nov. 23-24, 1994, Institutionen för datavetenskap, Linköpings universitet. Även publicerad i Dahlbom, B. (1995) (red): *"The Infological Equation - Essays in Honour of Börje Langefors"*, Gothenburg Studies in Information Systems, Report 6, Göteborgs universitet.
LiTH-IDA-R-95-14	**Anders G Nilsson (red), bidrag från 21 doktorander:** Klassisk Informationssystemteori: Fokus på Börje Langefors infologiska teoribildning.
LiTH-IDA-R-95-15	**Erik Sandewall:** Systematic Comparison of Approaches to Ramification.
LiTH-IDA-R-95-16	**Peter Jonsson, Christer Bäckström:** Tractable Planning with State Variables by Exploiting Structural Restrictions. A Short version of this report appears in the *Proc. of the Twelth National Conf. of the American Association for Artificial Intelligence (AAAI-94),* Seattle, WA, USA, July-Aug., 1994, pp. 998-1003. Using Restricted Minimization of Change. *Visit at LAAS-CNRS.*

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 the following net address:

Internet   *username*`@ida.liu.se`

Each person's *username* is listed in the column *Username* in the staff list below.

Some materials may be obtained via anonymous FTP:

FTP: ftp.ida.liu.se (130.236.30.131)

Information concerning the department and it's employees is presented via WWW. The department's home page:

URL: http://www.ida.liu.se/

IDA staff

Below follows an alphabetically ordered list of the IDA staff members, up to March 1995, with phone numbers (including area code), and user names on the local computer network.

Name	*Phone*	*Username*
Ahlenbäck, Britt-Marie	013-28 1318	bma
Ahrenberg, Lars	013-28 2422	lah
Aleryd, Anders	013-28 1707, 010-223 17 04	aoa
Andersson, Ingalill	013-28 1706	ingan
Andersson, Jörgen	013-28 2537	joran
Andersson, Magnus	013-28 2416	magan
Andersson, Mats S	013-28 1425	msa
Andersson, Niclas	013-28 1464	nican
Arendell, Pia	013-28 2403	par

Name	*Phone*	*Username*
Axelsson, Jakob	013-28 1396	jakax
Axelsson, Karin	013-28 2679	karax
Bayers, David	-	davby
Berglund, Eva-Britt	013-28 1458	evabe
Bilos, Rober	013-28 2164	rob
Björklind, Andreas	013-28 1969, 070-540 63 83	abj
Björklund, Lisbeth	013-28 2420	lbj
Blom-Lingenhult, Gunilla	013-28 2297	glt
Bonnier, Staffan	013-28 1426	stabo
Boye, Johan	013-28 2694	johbo
Bäckström, Christer	013-28 2429	cba
Carlshamre, Pär	013-28 2696	parca
Carlsson, Erik	013-28 1799	erica
Cronholm, Stefan	013-28 1638	scr
Dahlbäck, Nils	013-28 1664	nda
Degerstedt, Lars	013-28 2168	larde
Doherty, Patrick	013-28 2426	patdo
Drabent, Wlodzimierz	013-28 1483	wdr
Drakengren Thomas	013-28 2693	thodr
Driankov, Dimiter	013-28 1716	ddr
Dössing, Åsa	013-28 2859	-
Edgar, Camilla	013-28 1781	camed
Eklund, Fredrik	013-28 4022	freek
Ekman, Barbara	013-28 1410	bek
Elfinger, Eva	013-28 1524	evael
Emanuelson, Inger	013-28 2180	ime
Engelsons, Vadim	013-28 1979	vaden
Engström, Kent	013-28 4023	kenen
Ericsson, Mikael	013-28 4021	miker
Eriksson, Henrik	013-28 2673	her
Eskilsson, Anne	013-28 1460	annes
Fagerström, Johan	013-28 1364, 010-291 80 62	jfa
Fahl, Gustav	013-28 2670	gusfa
Fahlgren, Christer	-	-
Falk, Thomas	013-28 2538	
Falkenroth, Esa	013-28 4019	esafa
Finmo, Leif	013-28 1443	lbf
Fjällström, Per-Olof	013-28 2412	pof

Name	*Phone*	*Username*
Flodin, Staffan	013-28 2675	stala
Forslund, Göran	013-28 2602	gorfo
Franzén, Birgitta	013-28 2692	bfr
Fritzson, Peter	013-28 1484	paf
Fäldt, Arne	013-28 1962	arf
Goldkuhl, Göran	013-28 1452	ggo
Granlund, Rego	013-28 2465	reg
Gu, Xinli	013-28 1970	xgu
Hall, Thomas	013-28 1465	thoha
Hallberg, Jonas	013-28 1427	jogha
Hallberg, Niklas	013-28 2413	nikha
Haraldsson, Anders	013-28 1403	aha
Helander, Björn	013-28 1547	bjohe
Hirsch, Richard	013-28 1937	richi
Hjerppe, Roland	013-28 1965	rhj
Holmgren, Hans	013-28 1966	heh
Håkegård, Jan	013-28 2698	janha
Häger, Madeleine	013-28 1652	madha
Hägglund, Patrik	013-28 4018	patha
Hägglund, Sture	013-28 1431	sth
Höiem-Flyckt, Patrik	-	-
Ingels, Peter	013-28 1466	petin
Jacobson, Annemarie	013-28 1496	amj
Jenvald, Johan	013-28 2465	-
Johansson, Mikael	013-28 1479, 0708-28 14 79	mjo
Johansson, Olof	013-28 2390	ojo
Jonsson, Peter	013-28 2415	petej
Jonsson, Torbjörn	013-28 2467	tjo
Jönsson, Arne	013-28 1717	arj
Kalmelid, Stefan	013-28 1470	steka
Kamkar, Mariam	013-28 1949	mak
Karlsson, Joachim	013-28 1973	joaka
Karlsson, Jonas	013-28 2668	jonka
Kim, Ezra Hee-Cheol	013-28 1891	heeki
Kovordanyi, Rita	013-28 1430	ritko
Krajina, Patrik	013-28 4023	patkr
Krysander, Christian	013-28 1411	ckr
Kuchcinski, Krzysztof	013-28 1883	kku
Kågedal, Andreas	013-28 2419	aka

Name	*Phone*	*Username*
Lambrix, Patrick	013-28 2605	patla
Larsen, Kristina	013-28 2607	krila
Larsson, Erik	013-28 4046	erila
Larsson, Lena	013-28 44 28	lenla
Larsson, Mats	013-28 1819	mla
Lecerof, Andreas	-	-
Lennartsson, Bengt	013-28 1968	bln
Lilja, Carita	013-28 1463	cli
Lin, Ling	013-28 4429	linli
Lindeberg, Lena	-	lenli
Lindh, Jens-Olof	013-28 2493	jenli
Lindström, Jörgen	013-28 1526	
Lindvall, Mikael	013-28 1950	mikli
Linge, Lisbeth	013-28 1472	lli
Litwin Witold	-	-
Loborg, Peter	013-28 2494	plo
Lorenzen, Jeffrey	013-28 2608	jeflo
Lundqvist, Mattias	013-28 2859	-
Löfstrand, Katarina	013-28 1481	ksd
Löwgren, Jonas	013-28 1482	jlo
Malec, Jacek	013-28 2362	jam
Maleki, Jalal	013-28 1963	jma
Maluszynski, Jan	013-28 1483	jmz
Mattsson Kihlström, Bodil	013-28 2169	bmk
Merkel, Magnus	013-28 1964	mme
Moberg, Anna	013-28 2542	annmo
Nadjm-Tehrani, Simin	013-28 2411	snt
Nilsson, Bernt	013-28 1975	bkn
Nilsson, Björn	013-28 1476	ban
Nilsson, Fredrik	013-28 2859	freni
Nilsson, Henrik	013-28 2601	henni
Nilsson, Ivan	013-28 2423	ivani
Nilsson, Nils	013-28 2418, 010-220 74 07	nni
Nilsson, Peter J	013-28 1354	pjn
Nilsson, Rolf	013-28 2405	ron
Nilsson, Ulf	013-28 1935	urn
Nordberg, Sara	-	-
Nordling, Patrik	013-28 1948	patno
Näslund, Torbjörn	013-28 1957	tor

Name	*Phone*	*Username*
Ohlsén, Lennart	013-28 1555	lenoh
Ohlsson, Niclas	013-28 2594	nicoh
Ollinen, Jan	013-28 2544	janol
Olsén, Hans	013-28 2660	hanol
Olsson, Tommy	013-28 1954	tao
Orsborn, Kjell	013-28 1821	kjeor
Padgham, Lin	013-28 2066	lin
Padron-McCarthy, Thomas	013-28 2677	tpm
Peng, Zebo	013-28 2067	zpe
Persson, Jonas	013-28 1477	jonpe
Persson, Tommy	-	tpe
Petersson, Jan	013-28 1936	jpe
Pettersson, Mikael	013-28 2683	mpe
Poignant, Lars	013-28 1832	larpo
Ragnemalm, Eva L.	013-28 1947	elu
Rankin, Ivan	013-28 2424	ira
Rapp, Birger	013-28 1525	birra
Rauch, Ewa	013-28 1938	era
Raunio, Lise-Lotte	013-28 2430, 070-593 47 26	llr
Ringström, Johan	013-28 2406	jri
Risch, Tore	013-28 2541	torri
Rundfelt, Rolf	013-28 2538	-
Röstlinger, Annie	013-28 2407	aro
Sandahl, Kristian	013-28 1467	krs
Sandewall, Erik	013-28 1408	ejs
Savén, Bengt	013-28 2549	bensa
Sedvall, Göran	013-28 1977	gse
Shahmehri, Nahid	013-28 2066	nsh
Shu Hua	-	-
Sivertun, Åke	013-28 1497	akesi
Sjöberg, Cecilia	013-28 2596	cecsj
Sjölin, Martin	013-28 2410	marsj
Sköld, Martin	013-28 2417	marsk
Stoy, Erik	013-28 2691	est
Strömberg, Lars	013-28 1976	lst
Strömbäck, Lena	013-28 2676	lestr
Svenberg, Stefan	013-28 2690	ssv
Svensson, Eva-Chris	013-28 1457	ecs
Svensson, Lise-Lott	013-28 2360	lissv

Name	*Phone*	*Username*
Takkinen, Juha	013-28 2603	-
Thurée, Åke	013-28 2492	ath
Timpka, Toomas	013-28 1471	tti
Törne, Anders	013-28 2365	ato
Viklund, Lars	013-28 2689	lvi
Villegas B., Jaime Eduardo	013-28 2539	jaivi
Wallgren, Jonas	013-28 2682	jwc
Wallgren, Lillemor	013-28 1480	lew
Werner, Magnus	013-28 1844	magwe
Westman, Rickard	013-28 4018	ricwe
Wigh, Lena	013-28 1756	len
Wiklund-Johansson, Magnus	013-28 2859	-
Willén, Olle	013-28 1413, 010-678 13 55	owl
Yi, Choong-Ho	013-28 1314	choyi
Öberg, Fredrik	013-28 2669	freob
Överström, Pål	013-28 4024	palov

IDA-Events

IDA expands

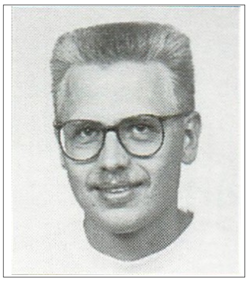
Inauguration of new premises

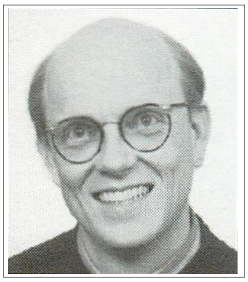
Department of Computer and Information Science

Undergraduate Education

Undergraduate Organization – New Directors

Mikael Johansson, new director of the SVP program.





Nils Dahlbäck, new director of the Cognitive Science program.

Jörgen Andersson has replaced Stefan Blom as director of the subject area Economic Information Systems.

Jalal Maleki has replaced Arne Jönsson as director of the subject area Cognitive Science.

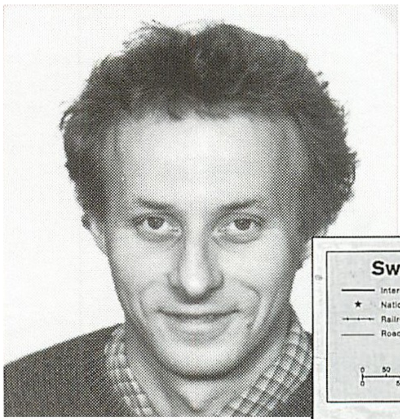
IT Program

Mariam Kamkar and **Bengt Lennartsson** are engaged in the development of the IT program in the undergraduate education.

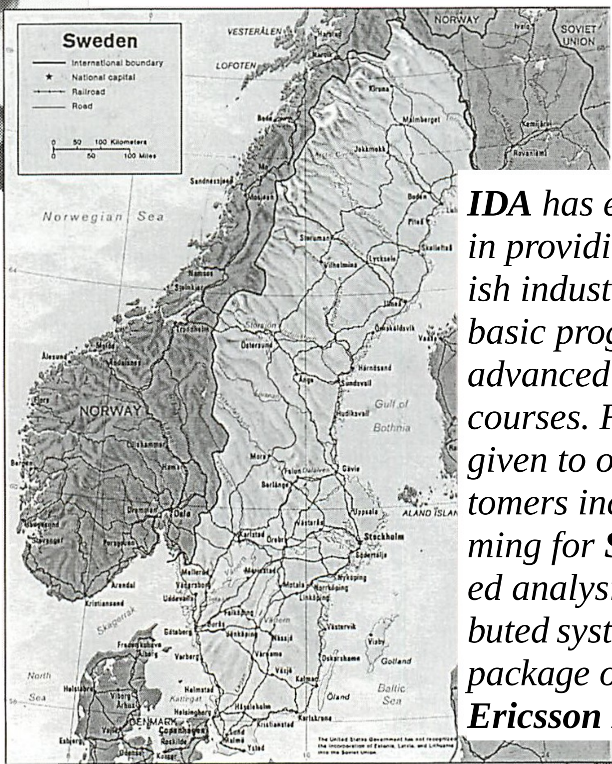
International group FHS

Lise-Lotte Raunio is IDA's contact person for the philosophical faculty's international group.

External Undergraduate Education



Johan Fagerström *is responsible for all of IDA's external undergraduate education.*



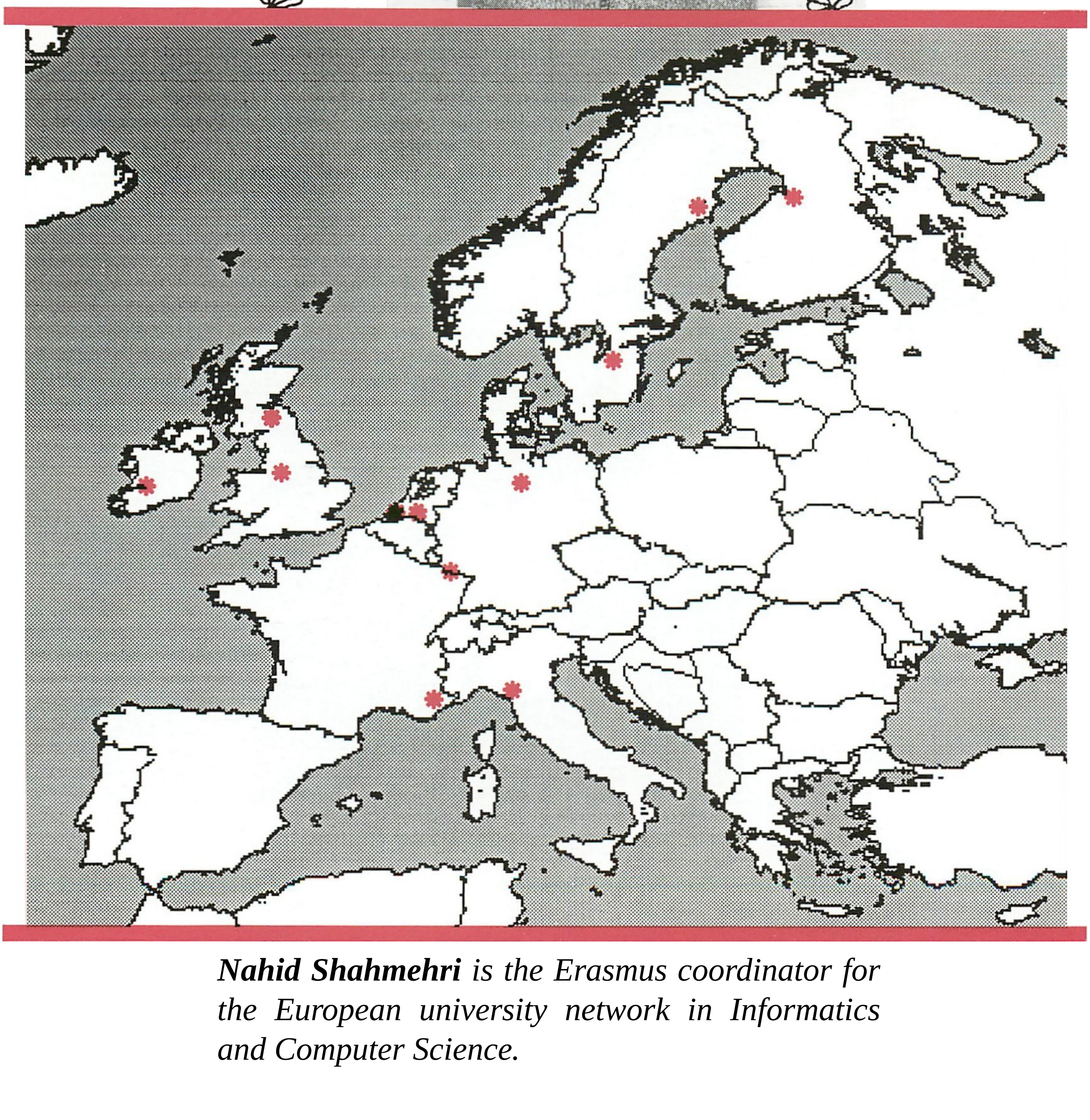
IDA *has extensive experience in providing courses for Swedish industry. These vary from basic programming courses to advanced level, custom made courses. For instance, courses given to our latest three customers include Ada programming for* ***SAAB****, object oriented analysis/design and distributed systems for* ***SMHI****, and a package of 14 courses for* ***Ericsson Radio Systems****.*

Christian Krysander *is responsible for developing and delivering courses on the development of large technical systems for Celsius Tech AB, Järfälla. The aims of this education are to raise the competence of engineers, who have had 4 to 10 years of industrial experience.*

IDA has an extensive external education.

Erasmus Coordinator

The European university network *is active in student mobility. For the next academic year the network will consist of the nodes: Oulo/Finland, Nice/France, Braunschweig/Germany, Saarbrucken/Germany, Limerick/Ireland, Pisa/Italy, Amsterdam/The Netherlands, Leiden/The Netherlands, Linköping/Sweden, Umeå/Sweden, Edinburgh/United Kingdom, and Manchester/United Kingdom.*

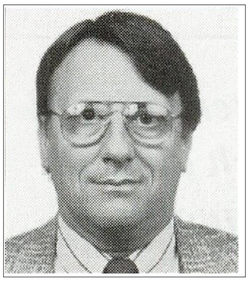


Nahid Shahmehri *is the Erasmus coordinator for the European university network in Informatics and Computer Science.*

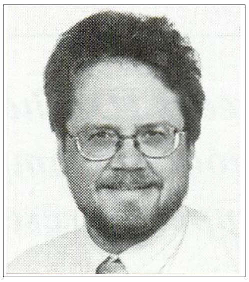
Research and Research Education

Research Organization

During the academic year 1994/95 the Laboratory for Computer Assistance in Engineering (CAELAB) was divided into two labs, **EDSLAB** headed by **Tore Risch**, and **RTSLAB**, headed by **Anders Törne**.



EDSLAB, the Laboratory for Engineering Databases and Systems, does research on new database services and advanced applications in particular for supporting engineering information systems. The research is centered around the umbrella project AMOS – Active Mediators Object System, a next generation object-relational database mediator between applications and data sources. The laboratory also conducts database-oriented application projects for engineering support in cooperation with Swedish industry.



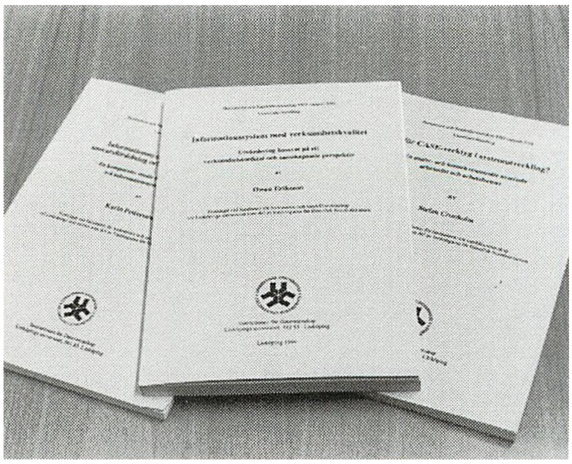
RTSLAB, the Real-Time Systems Laboratory, research covers tools, methods and architectures for the design of software intensive real-time systems. In particular this includes timing analysis and design-synthesis tools based on discrete modelling methods, database modelling of activities for control and simulation, and stratified real-time software architectures. The laboratory also conducts applicative research and case studies in cooperation with industry, for example in the automation and control area.

New Laboratory Leader

Nahid Shahmehri, new leader for the Laboratory for Intelligent Information Systems (**IISLAB**).

New Graduate Study Programme

During Autumn 1994 IDA established a new research programme in Information Systems Development, headed by **Göran Goldkuhl**. This programme falls withing



A number of licentiate theses has already been presented.

Graduate School and Strategic Research Program in Computer Science and Systems Engineering

Since many years IDA has organized an International Graduate School in Computer Science which aims at educating competent researchers at a high quality establishment

IDA is also involved in a new proposal from Linköping University to create a Graduate School and Strategic Research Program in Computer Science and Systems Engineering.

The main idea is to create an integrated research program and graduate school across the fields of Computer Science and Systems Engineering, in order to:

- significantly increase the number of Ph. D.s and Tech. Lic.s with broad and integrated competence over these areas
- strengthen and develop existing research in basic areas, with a particular emphasis to bridge the gap between the disciplines.
- focus on a handful of industrial problem areas which require joint approaches and tools.

Graduate Program in Economic Information Systems

Research in EIS falls within four main areas which are:

Information Technology and New Organization Structures.

Simulation and Production.

Information and Control Systems.

A three-year licentiate program. *

* In 1991 the cooperation between Economic Information Systems and several accounting firms was established and the special research education program was developed. During three years the graduate students study in parallel with working half-time at an accounting firm.

Professor Birger Rapp, head of EIS, hears how graduate student Peter Carlsson feels that his work at an accounting firm has stimulated both his studies and research at IDA.
(Source: *Bilder av revision och revisorer*, Öhrlings Reveko, 1995.)

New Adjunct Professors and Advisors

Thomas Falk

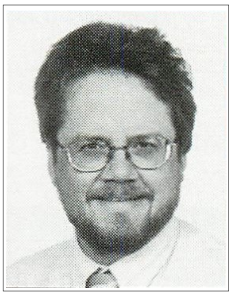
Nils-Göran Olve

Anders G. Nilsson

Cooperation with Industry

IDA is involved in the following CENIIT-supported projects.

- Engineering Databases, Tore Risch (project leader).
- Computer-Aided Design of Real-Time Systems, Anders Törne (project leader).
- Algorithms for Geometric Contact Searching, Per-Olof Fjällström (project leader for IDA).
- Autonomous Real-Time Systems, Jacek Malek and Erik Sandewall (project leaders).
- Industrial Software Engineering, Kristian Sandahl (project leader).
- Parallel Programming Environments for Scientific Computing, Peter Fritzson (project leader for IDA).
- Reliable Software by Automated Testing, Mariam Kamkar (project leader).



Anders Törne, IDA, is manager of the Center for Industrial Information Technology (CENIIT).

Examples of Cooperation between Research Labs and Industry

RTSLAB and **RKLLAB** are cooperating with **Saab Military Aircraft**, **Volvo Areo Corporation** and **VOAC** within the area of methods and tools for system development. The project "Specification, Synthesis, Modelling and Verification of Complex, Heterogeneous Systems" is concerned with tools and methods for the system development process and the goal is to investigate the applicability of different tools and methods.

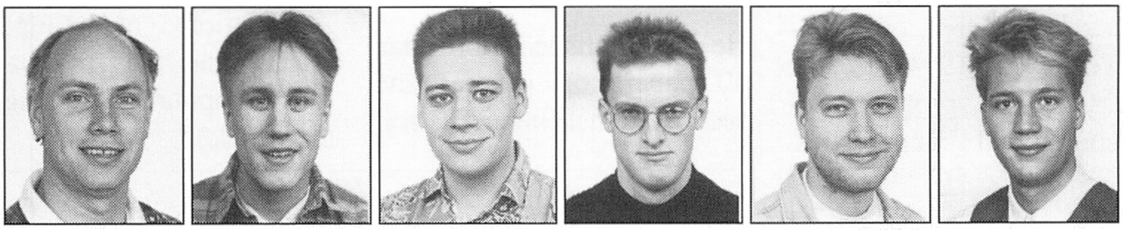
The database research within **EDSLAB** has resulted in considerable interest from Swedish industry. **EDSLAB** is presently having cooperation with **Ericsson** within the telecom switching area and contacts with **ABB Industrial Systems** are promising. Engineering database technology has been applied successfully in a cooperation project with **ABB Stal**.

All three groups are involved in the establishing of a competence center, **ISIS**, together with **Automatic Control and Vehicular Systems** at the **Electrical Engineering Department**. **ISIS** or "Integrated Control and Information Systems" is funded by **NUTEC** and involves cooperation with several Swedish industries - **Saab Military Aircraft**, **ABB Industrial Systems**, **Ericsson**, **Mecel**, **Saab Dynamics**, and others. ISIS will be launched during 1995.

Examples of Cooperation between Research Labs and Industry

SKF's new revolutionizing rolling bearing CARB, announced April 4, 1995, is to a large extent the result of advanced computer simulations. The simulation programs originate from mathematical models with formulae that describe the bearing, which kind of load it can take, etc.

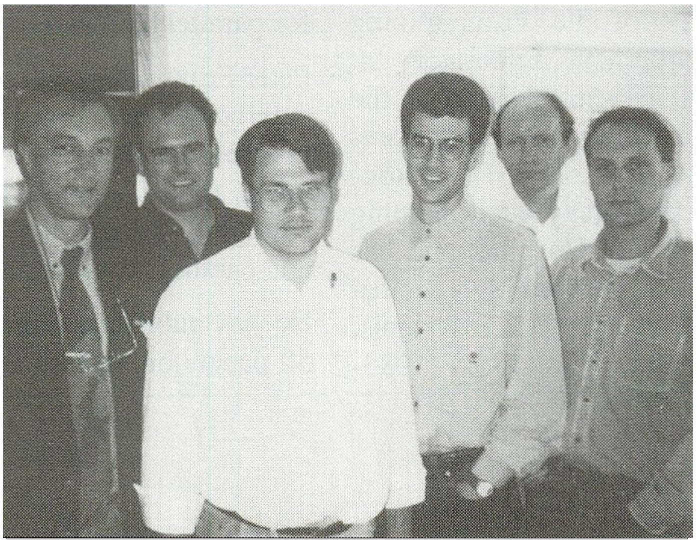
The formulae and equations have then been transformed into simulation programs which have been run on multi-processors, on a 20-processor SparcCenter 2000 at the Department of Computer and Information Science at Linköping University, and on an 8-processor SparcServer 1000 at **SKF ERC** in The Netherlands. Linköping University and **PELAB** have contributed in parallelizing the computations in order to find a way to further decrease the development times and handle larger simulation problems.



PELAB members engaged in the project.

Within the Software Engineering area, **ASLAB** has established intensive cooperation with **Ericsson Radio Systems**, **Ericsson Telecom**, **SAAB Military Aircraft** and others with special support from the **Wallenberg Foundation**. Within this programme, prominent guest professors from abroad, such as Robert Glass, David Parnas, Vic Basili, Kevin Ryan and others have been visiting. Ericsson Radio is sponsoring several graduate students, working with industry-based Ph.D. projects.

This programme is managed by Dr. **Kristian Sandahl**, and the first licentiate, **Mikael Lindvall**, presented his thesis on traceablity in object-oriented systems development in 1994.



ASLAB members together with representatives from Ericsson Radio Systems AB.

There are also numerous other joint activities and contacts with industry.

Assignments

New Professor

Lars Ahrenberg, professor in Computational Linguistics.

Lars Ahrenberg is director of the Natural-Language Processing Laboratory, where he was one of the founding members in 1985. Previously he has worked at Uppsala University, Dept. of Linguistics (1978-84) and University of Göteborg, Dept. of Swedish (1984-85, 1994-95), the latter part as professor of Computational Linguistics. His research interests include computer-aided translation, natural-language dialog systems and linguistic knowledge representation from morphology and syntax to semantics and pragmatics. A recently started project considers the architecture of linguistic databases for fast generation of language models adapted to a given application.

He has published more than 40 papers on general and computational linguistics and

is a member of the Nordic Association of Linguists (NAL) and the Association for Computational Linguistics (ACL).

New Professor

Peter Fritzson, professor in Software Engineering.

Peter Fritzson is professor of programming systems and software engineering, and director of the Programming Environment Laboratory at IDA. He has worked at the Dept. of Physics an Measurement Technology at Linköping University, at Linköping University Hospital, and has been a member of technical staff at Sun Microsystems, Mountain View, USA (1985-86). His research interests include programming environments, scientific computing, debugging tools, incremental compilation technology, compiler generation, compilers, and programming tools for parallel hardware.

Peter Fritzson has edited several conference proceedings in areas such as programming tools, compiler construction, debugging tools, and parallel programming.

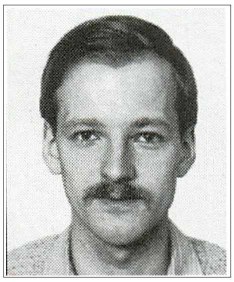
He has published more than 50 papers on various aspects

of programming environments and software engineering. He is a member of IEEE Computer Society and of ACM.

Assignments

New Docents

Lin Padgham



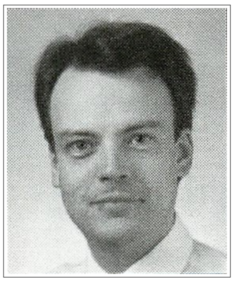
Wlodzimiertz Drabent

Krzysztof Kuchcinski

Jonas Löwgren

Zebo Peng

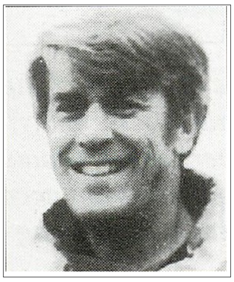
Sten Andler



Henrik Eriksson

Two Honorary Doctors in Computer Science

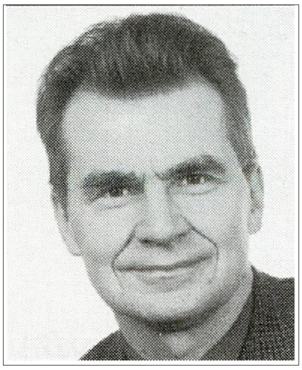
Bjarne Däcker
Ellemtel



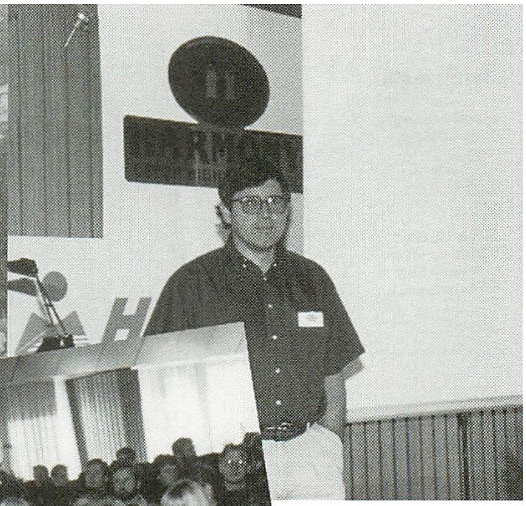
Robert L. Glass
Computing Trends
Bloomington, Indiana

IDA employees with special assignments

Erik Sandewall has been appointed pro-vice-chancellor at Linköping University for the period of 1995-2001.



IDA Conference...

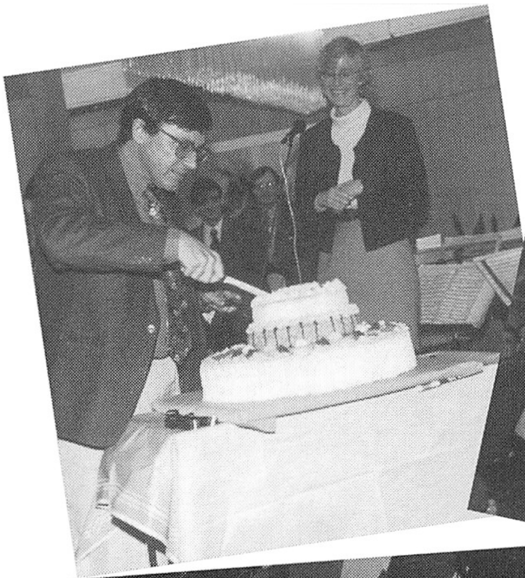


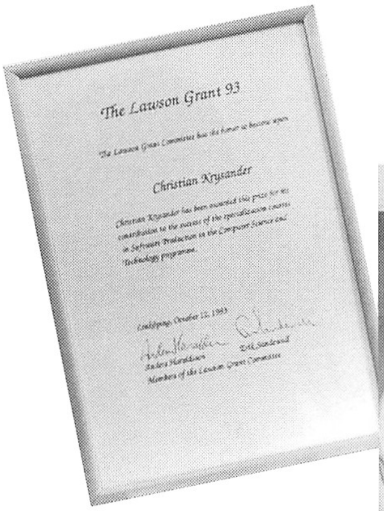
The members of the department discussed:
What is IDA today?
Where are it's roots?
What has been accomplished?
What is happening around IDA?

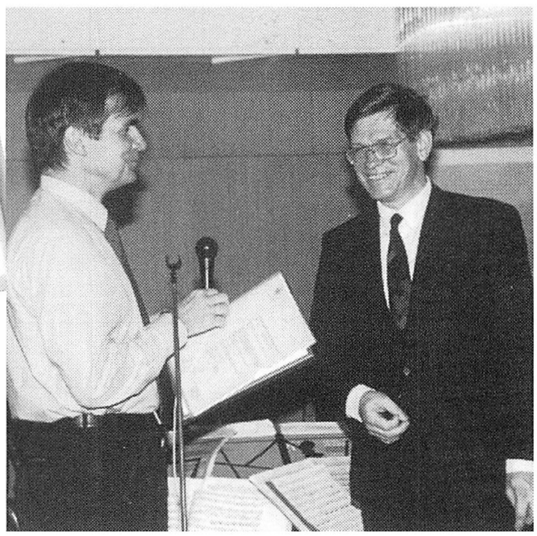
IDA conference in 1993 in Spindleruv Mlyn

In October 17-20, 1993, the whole department participated in a conference to celebrate the 10 year annivesary and to discuss the department's activities and goals.

... and IDA celebrates 10 years







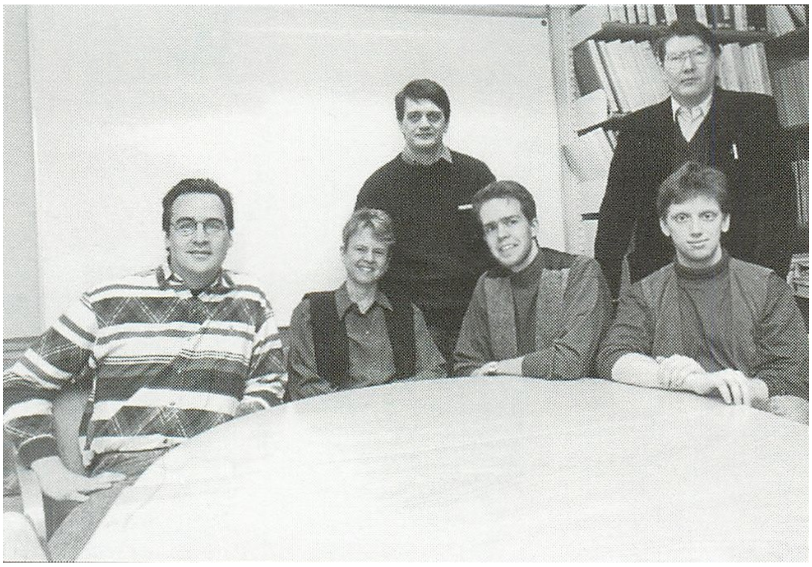
Christian Krysander receives the Lawson Prize from Erik Sandewall.

The Computer and Information Science Department was founded in 1983.

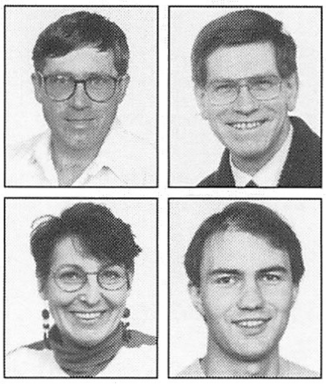
Equal Opportunity, Quality and IT

The IDA Board identifies three work groups for Equal Opportunity, Quality and IT

IDA's equal opportunity group makes efforts to make IDA an institution which stresses equal opportunity aspects in work, and organization, as well as in personnel structure. Concrete work group areas are: to make the personnel aware of equal opportunity issues, identification/elimination of salary differences based on gender, recruitement and delegation of work, and career planning. IDA received 200' SEK to support this work from the Swedish Government.



The IT Strategy Group (right) deals with issues related to using new technologies to structure and make efficient, information flow between students, teachers, researchers and administrators.



The Quality Group's most important tasks are

- to inspire quality improvement initiatives within IDA.
- to follow up and report the results reached by the university quality group.
- to cooperate with other departments concerning concrete quality activities.
- to collect ideas and report on fullfilled activities.

Equal Opportunity, Quality and IT

Twice a year the project "Flickor och teknik" (Girls and technology) brings about 70 high school girls to visit IDA, and gives them an introduction to computer science and the department. The main purpose is to introduce computer science as an option for their university studies and as a future labour market. Another goal is to give girls female role models. The organizers of the project are **Simin Nadjm-Tehrani**, **Kristina Larsen**, **Mikael Eriksson**, **Göran Forslund**, **Eva Ragnemalm**, **Cecilia Sjöberg**, **Katarina Löfstrand**, and **Lena Strömbäck**.

The subsequent evaluation of the project, showed that over 50% of the participants were very satisfied with the presentations, whole the rest were satisfied. Here are some examples of their answers:

”Good to learn how women can handle this.”

”I feel more inspired to learn more about computing after all. That could be fun and probably useful.”

”My prejudices (fears) against computers have diminshed somewhat, that is more than anybody else has managed to do before.”

IDA wants to attract more women to computer science.

The organizers of the project ”Flickor och teknik” have reached good results in changing attitudes to computers.

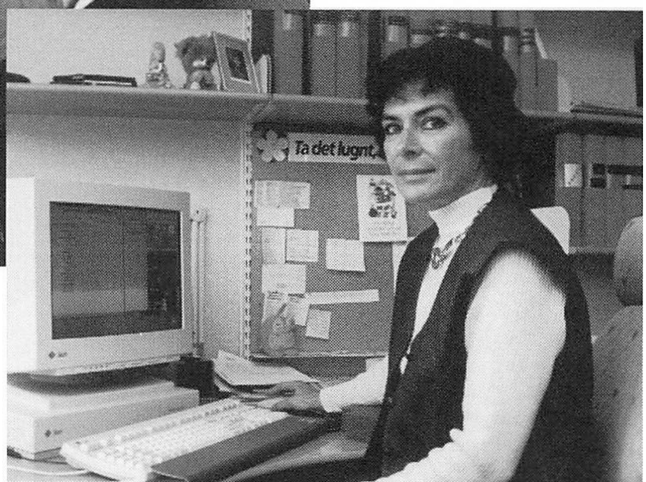
Party arranged by IDA's Equal Opportunity Group


Holland
FRISKIS &

Renewal of Computer Facilities

The new multimedia lab at IDA is equipped with eight Macintosh 7100 computers.

In January 1995 the undergraduate computing system's Sun workstations were replaced by newer SPARCstation 10 with Colour monitors.



During the spring IDA personnel have had their older Sun work-stations replaced. This brings the number of replaced workstations to 270.

IDA invests in providing an environment with good computer facilities.

Supercomputers to IDA

Parsytec parallel computer with 128 processors.

...dag 9 november 1994

Universitetet samarbetar med EU

Nya datorn del i super-supernät

LINKÖPING
Två små klipp. Två höga tjänstemän. Och ett viktigt samarbete är formellt bekräftat.

Linköpings universitets senaste och Sveriges snabbaste superdator är invigd. Den klarar 10 miljarder multiplikationer i sekunden.

Det är start.

Sverige och Linköping ... en datamaskin som vät ... cetar universitetet till sa... forskning...

snabba datorer inom kort ska anslutas med ännu snabbare telekommunikation och att det ska gå att se dem som ett superdatornät där räknehästarna samarbetar för att lösa ett avancerat beräkningsproblem.

Superdatorn i Linköping är att se som ett frö av teknisk muskelstyrka och uppbyggnad av mänsklig kunskap. Och inte minst en spirande ... utgångspunkt ... mella...

skörda de största frukterna av de resultat som kommer ur superdatorforskningen. Åtminstone inte om det inte finns andra sätt att lösa finansieringen vid ett eventuellt nej på söndag.

Universitetskansler Stig ... ström ser EU ... sönda... fo...

...sök slå det . . .

...amarbete mellan EU och universitetet

Här är Sveriges snabbaste dator

På tisdagen invigdes Sveri...s snabbaste dator vid uni...rsitetet i Linköping. Da...n är frukten av ett intimt ...marbete med EU. Sverige ...har bekostat hälften och EU ...hälften av den nya super...snabba maskinen.

Ett nej till EU kan riskera projekt av detta slag. Dels av ekonomiska skäl, dels av mänskliga.

— Jag tror att det är psykologiskt viktigt med ett ja till EU för att det ska bli bra fortsatt samarbete, säger Stig Hagström, universitetskansler. Vi måste vara på bettet och inte bara sitta på sista bänk och avvakta utvecklingen.

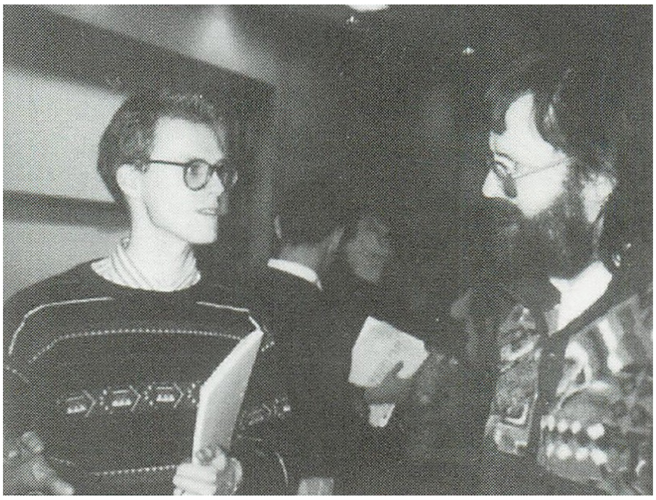
På tisdagen klippte han bandet parallellt med Jean-Francois Omnes från EU-kommissionens direktorat för satsning på superdatorer.

— Kommissionen ger pengar till projekt som gynnar de länder som bidrar med pengar till EU:s budget. Men jag vet inte hur samarbetet utvecklas om Sverige säger nej, kommenterar Jean-Francois Omnes.

Sidan B3

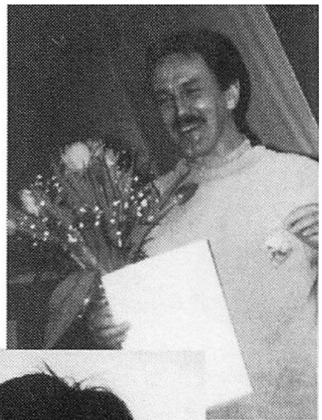
Source: Östgöta Correspondenten, November 9, 1994.

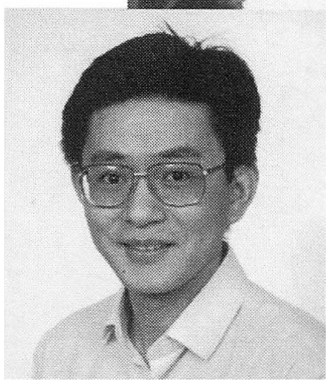
4th Annual Graduate Student Conference



Magnus Werner and John Hughes in conversation.

Best Paper Award to Xinli Gu and Jan Petersson.





Lillemor Wallgren receives the Lawson prize from Harold W. Lawson.

4th IDA Annual Conference, March 16-17, 1994

Professor John Hughes, Chalmers, Gothenburg, gives an invited talk on how to present research. During seven sessions, 29 papers were presented by graduate students. Best papers awards, and the Lawson Prize were given out.

Books recently written by IDA-authors

Objektorienterad systemutveckling – en introduktion

Johan Fagerström

Studentlitteratur

Objektorienterad analys och design – en andra generationens metod

Johan Fagerström

Studentlitteratur

Human-computer interaction
– what every system developer should know

Jonas Löwgren

Studentlitteratur

Programmering i Lisp

Anders Haraldsson

Studentlitteratur

Bilder av revision och revisorer

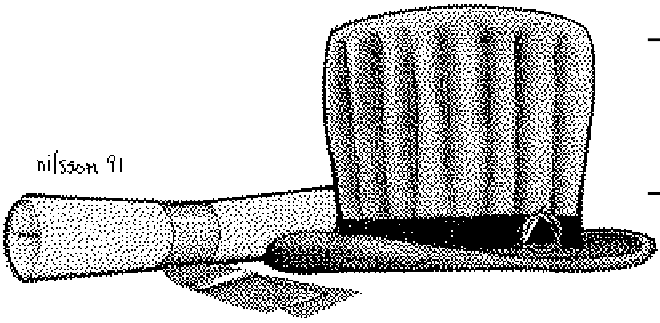
Birger Rapp et al.

Öhrlings Reveko

Verksamhetsutveckla datasystem

Göran Goldkuhl

Intention



Ph. D. Dissertations

from November 1992 to June 1995

Mats Wirén

Studies in Incremental Natural-Language Analysis. (No 292, 1992.)

Mariam Kamkar

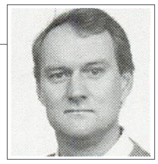
Interprocedural Dynamic Slicing with Applications to Debugging And Testing. (No 297, 1993.)

Tingting Zhang

A Study in Diagnosis Using Classification and Defaults.
(No 302, 1993.)

Arne Jönsson

Dialogue Management for Natural Language Interfaces – An Empirical Approach. (No 312, 1993.)



Simin Nadjm-Tehrani

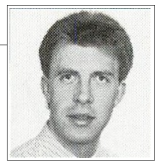
Reactive Systems in Physical Environments: Compositional Modelling and Framework for Verification. (No 338, 1994.)

Bengt Savén

Business Models for Decision Support and Learning.
(No 371, 1995.)

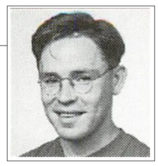
Ulf Söderman

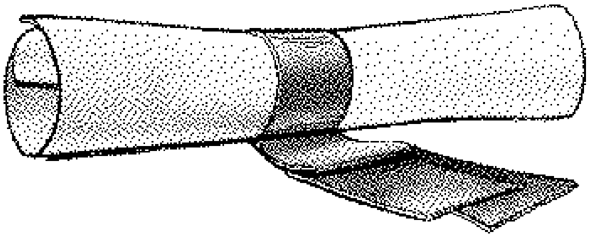
Conceptual Modelling of Mode Switching Physical Systems.
(No 375, 1995.)



Andreas Kågedal

Exploiting Groundness in Logic Programs.
(No 383, 1995.)





…and Lic. Theses

from November 1992 to June 1995

Ulf Cederling

Industrial Software Development - a Case Study.
(No 348, 1992.)

Magnus Morin

Predictable Cyclic Computations in Autonomous Systems:
A Computational Model and Implementation.
(No 352, 1992.)

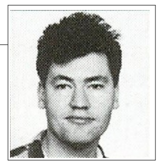
Mehran Noghabai

Evaluation of Strategic Investments in Information Technology.
(No 371, 1993.)

Mats Larsson

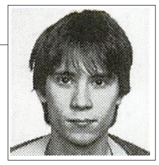
A Transformational Approach to Formal Digital System Design.
(No 378, 1993.)

Johan Ringström



Compiler Generation for Parallel Languages from
Denotational Specifications.
(No 380, 1993.)

Michael Jansson



Propagation of Change in an Intelligent Information System.
(No 381, 1993.)

Jonni Harrius

An Architecture and a Knowledge Representation Model
for Expert Critiquing Systems.
(No 383, 1993.)

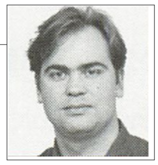
Per Österling

Symbolic Modelling of the Dynamic Environments of Autonomous Agents.
(No 386, 1993.)

Johan Boye

Dependency-based Groudness Analysis of Functional Logic Programs. (No 398, 1993.)

Lars Degerstedt



Tabulated Resolution for Well Founded Semantics.
(No 402, 1993.)

Anna Moberg

Satellitkontor - en studie av kommunikationsmönster vid arbete på distans.
(No 406, 1993.)

Peter Carlsson

Separation of Management and Financing, Case Studies of Management Buy-Outs in an Agency Theoretical Perspective.
(No 414, 1994.)

Camilla Sjöström

Revision och lagreglering - ett historiskt perspektiv.
(No 417, 1994.)

Cecilia Sjöberg

Voices in Design: Argumentation in Participatory Development.
(No 436, 1994.)

Lars Viklund

Contributions to a High-level Programming Environment for Scientific Computing.
(No 437, 1994.)

Peter Loborg

Error Recovery Support in Manufacturing Control Systems.
(No 440, 1994.)

Owen Eriksson

Informationssystem med verksamhetskvalitet - utvärdering baserat på ett verksamhetsinriktat och samskapande perspektiv. (FHS 3, 1994.)

Karin Pettersson

Informationssystemstrukturering, ansvarsfördelning och användarinflytande - En komparativ studie med utgångspunkt i två informationssystemstrategier. (FHS 4, 1994.)

Lars Poignant

Information Technology and Industrial Location - Effect on Productivity and Regional Development. (No 441, 1994.)

Gustav Fahl

Object Views of Relational Data in Multidatabase Systems. (No 446, 1994.)

Henrik Nilsson

A Declarative Approach to Debugging for Lazy Functional Languages. (No 450, 1994.)

Jonas Lind

Creditor - Firm Relations: an Interdisciplinary Analysis. (No 451, 1994.)

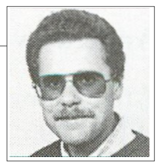
Martin Sköld

Active Rules based on Object Relational Queries - Efficient Change Monitoring Techniques. (No 452, 1994.)

Pär Carlshamre

A Collaborative Approach to Usability Engineering: Technical Communicators and System Developers in Usability-Oriented Systems Development. (No 455, 1994.)

Stefan Cronholm



Varför CASE-verktyg i systemutveckling? - en motiv- och konsekvensstudie avseende arbetssätt och arbetsformer. (FHS 5, 1994.)

Mikael Lindvall

A Study of Traceability in Object-Oriented Systems Development.
(No 462, 1994.)

Fredrik Nilsson

Strategi och ekonomisk styrning - En studie av Sandviks förvärv av Bahco Verktyg.
(No 463, 1994.)

Hans Olsén

Collage Induction: Proving Program Properties by Program Synthesis.
(No 464, 1994.)

Lars Karlsson

Specification and Synthesis of Plans Using the Features and Fluents Framework.
(No 469, 1994.)

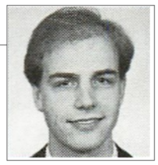
Ulf Söderman

On Conceptual Modelling of Mode Switching Systems.
(No 473, 1995.)

Choong-ho Yi

Reasoning about Concurrent Actions in the Trajectory Semantics.
(No 475, 1995.)

Bo Lagerström



Successiv resultatavräkning av pågående arbeten. Fallstudie i tre byggföretag.
(No 476, 1995.)

Peter Jonsson

Complexity of State-Variable Planning under Structural Restrictiones.
(No 478, 1995.)

Anders Avdic

Arbetsintegrerad systemutveckling med kalkylprogram.
(FHS 7/95.)

Eva L. Ragnemalm

Towards Student Modelling through Collaborative
Dialogue with a Learning Companion.
(No 482, 1995.)

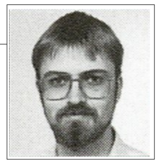
Eva Toller

Contributions to Parallel Multiparadigm Languages:
Combining Object-Oriented and Rule-Based Programming.
(No 488, 1995.)

Erik Stoy

A Petri Net Based Unified Representation for
Hardware/Software Co-Design.
(No 489, 1995.)

Johan Herber



Environment Support for Building Structured Mathematical Models.
(No 497, 1995.)

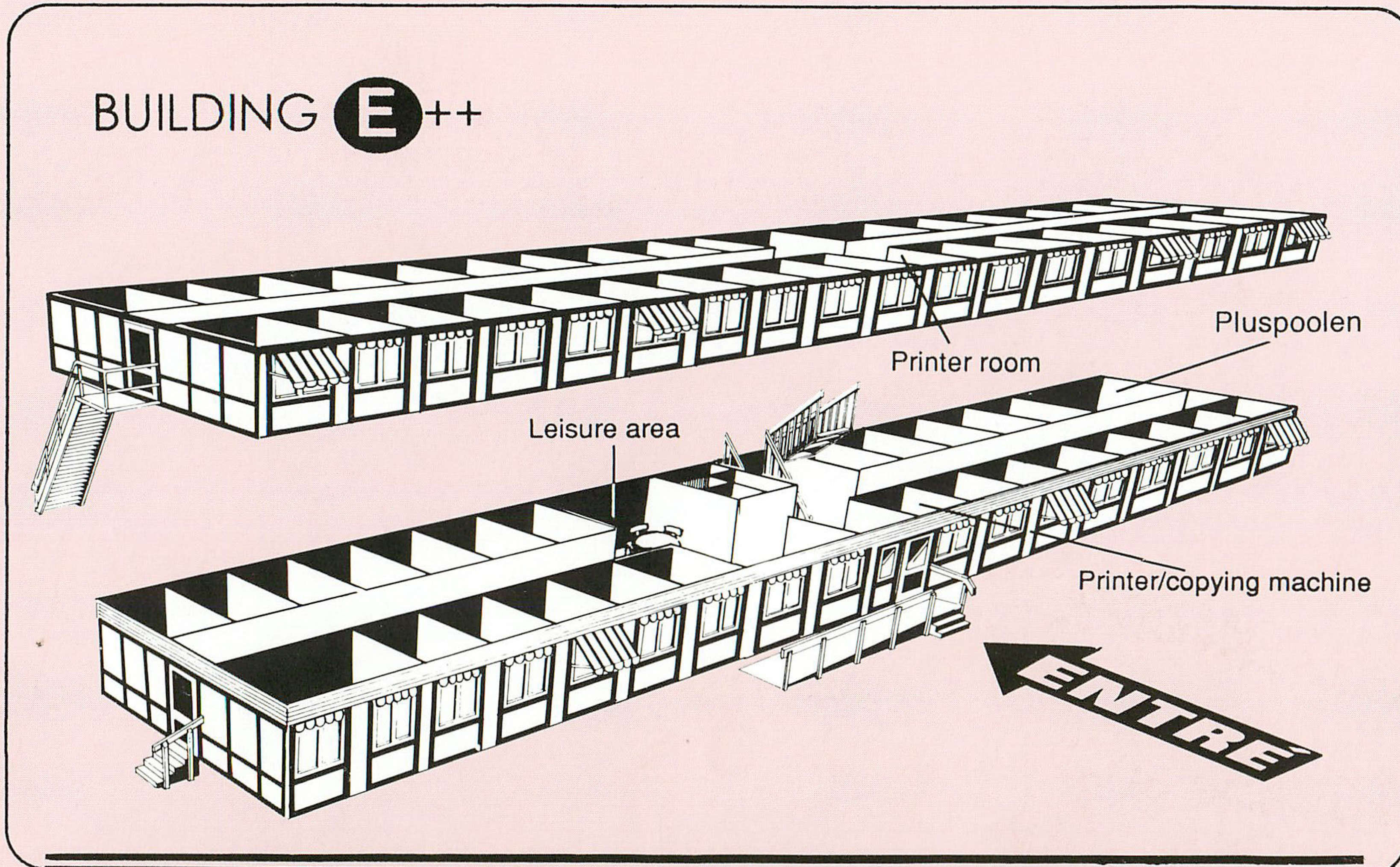
Stefan Svenberg

Strucure-Driven Derivation of Inter-Lingual Functor-Argument
Trees for Multi-Lingual Generation.
(No 498, 1995.)

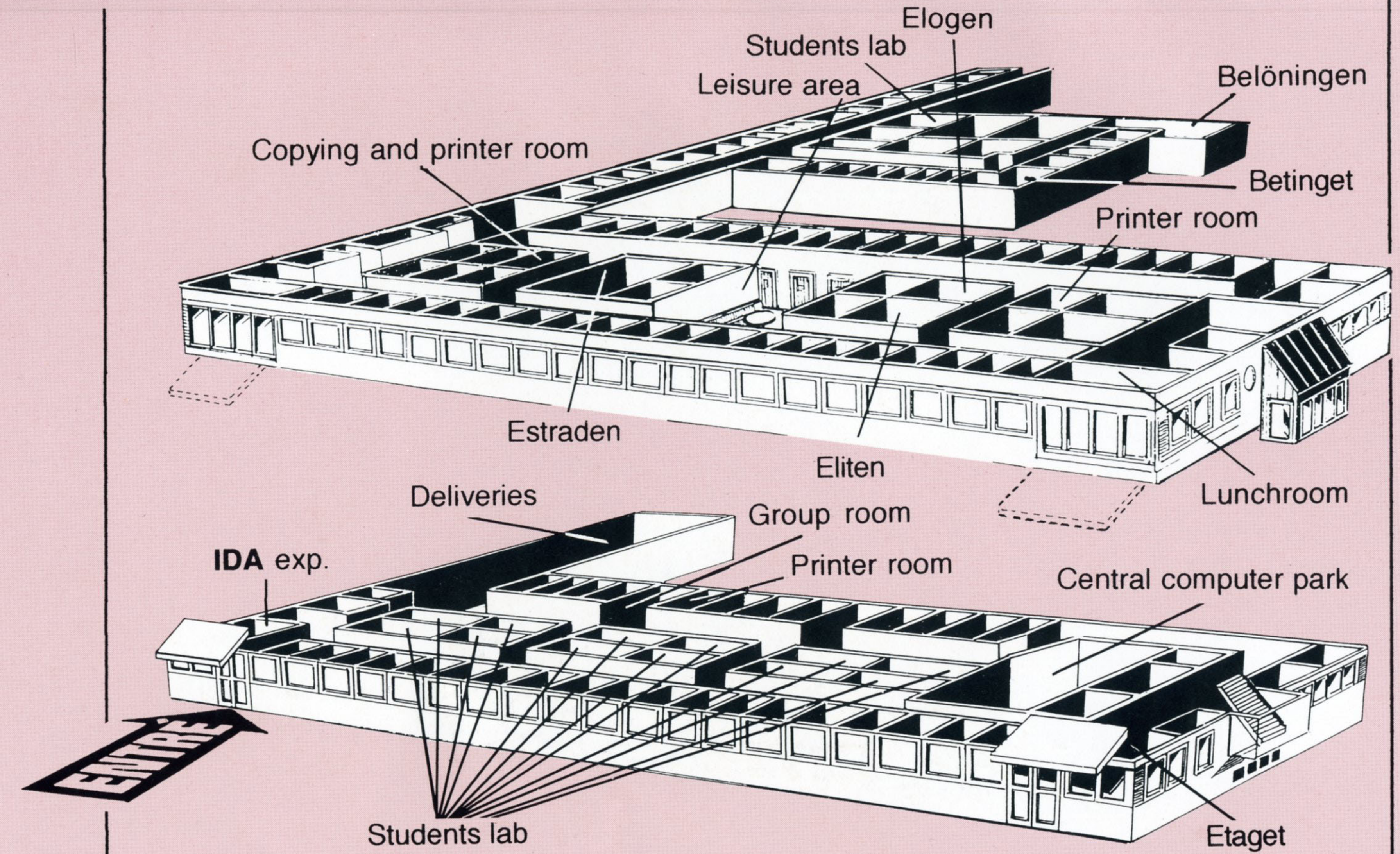
MASTER OF ENGINEERING THESES

Ying Shu

The Analysis and Implementation of a Terminological
Reasoner.
(No 2, 1993.)



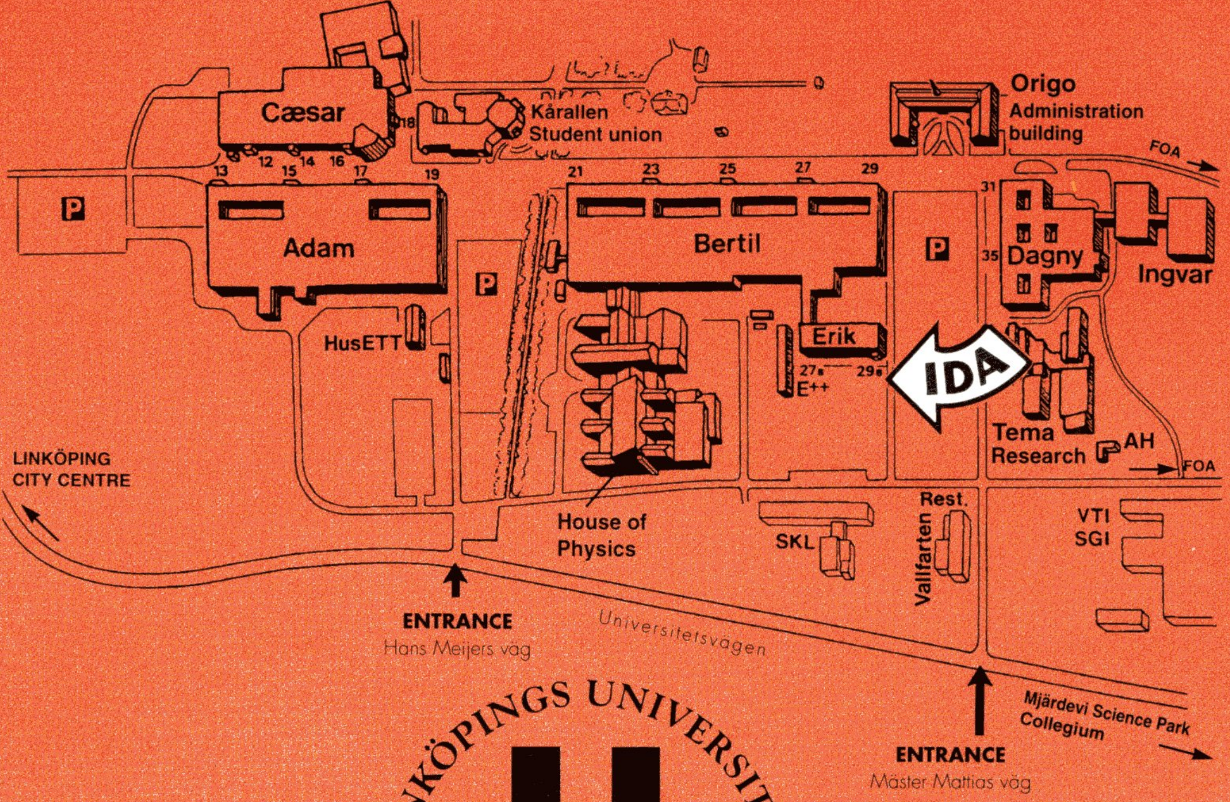
Department of COMPUTER and INFORMATION SCIENCE



Elogen
Students lab
Leisure area
Belöningen
Copying and printer room
Betinget
Printer room
Estraden
Eliten
Lunchroom
Deliveries
Group room
IDA exp.
Printer room
Central computer park
ENTRÉ
Students lab
Etaget

BUILDING E

Main University Campus



LINKÖPINGS UNIVERSITET