Computer Science
Aalborg University

Research Evaluation
1991–1995

Institute of Electronic Systems
Department of Mathematics and Computer Science
Fredrik Bajers Vej 7E ▪ DK 9220 Aalborg Ø ▪ Denmark

Computer Science, Aalborg University—Research Evaluation 1990–1995

Edited by Christian S. Jensen

Cover design by Kent Hansen
Cover reproduction by Schølin Grafisk
Printed by Centertrykkeriet, Aalborg University

R-96--2018
ISSN 0908--1216

Preface

It is the policy of the Faculty of Engineering and Science at Aalborg University that its research units be evaluated every five years. Following that policy, the present report documents the second research evaluation of the Computer Science unit and covers the period 1991–1995.

According to the Faculty's guidelines, the objectives of the evaluation are threefold. First, it is to be assessed whether there is a satisfactory agreement between allocated internal and external research resources and the research accomplished. Second, it must be determined whether the research unit's actual research efforts are consistent with the unit's goals and plans for the period as they are expressed in the long-term research plans of the Faculty of Engineering and Science. Third, the evaluation is intended to aid the research unit in planning its future research efforts and its organization of research.

Realizing that a research evaluation itself is a learning experience that provides an opportunity to rise above the daily routine and reflect upon research, we agreed upon an additional objective: Within the research unit, the evaluation—the process as well as the final report—should constructively aid the staff members in evaluating and improving their effectiveness as researchers, group leaders, and administrators of research at the departmental level. In particular, the evaluation presents an opportunity to assess the effectiveness of departmental policy.

In part to aid the reader in better understanding this report, we next describe the process that brought it about.

The research evaluation process began in early 1995 when discussions on the structure of the process were initiated. Among the early, important tasks, a three-member evaluation panel was appointed. Professor Kim G. Larsen accepted to be the local member of the panel (being local, he would be without a vote in the panel), and the search went on for external members. The unit was delighted when professor Sture Hägglund from Linköping University and professor Stig Skelboe from the University of Copenhagen agreed to be on the panel. We felt that this panel of recognized and experienced senior computer scientists would be able to cover all the rather diverse research areas in the unit and provide an insightful evaluation.

Also, at a series of meetings during spring of 1995, various members of the unit presented their perspectives on the evaluation of research. Specifically, the mem-

bers were asked to describe the criteria they themselves would use for evaluating their own research. These presentations amply demonstrated the diversity of perceptions of research and helped to break the ice, enabling research evaluation to become a topic for open discussion.

Finally, the general structure of the evaluation process and the report were agreed upon. Having completed one evaluation already helped substantially.

The report was to have two main parts, namely the unit's description of its research and the actual evaluation of the research unit. The description was to have separate chapters for the unit as a whole and for each of the unit's six research groups. Templates for these chapters were designed—it would then be up to the department chair and the groups to flesh them out. This design was chosen to allow the diverse research groups some degree of autonomy in how they wanted to report their research while achieving also an adequately coherent report with comparable levels of description. I was given the responsibility of steering the process, coordinating with Kim G. Larsen when appropriate, and of editing the groups' contributions. By early December 1995, a 170+iv-page preliminary version of the research description (Part II of this report) was completed and sent to the panel members along with representative research publications.

In January 1996, a two-day research evaluation seminar was held at a conference facility. All research staff, including Ph.D. students, and secretarial staff from the unit were invited. On the first day (the evaluation panel arrived a day in advance, to allow for initial discussions), the unit's research groups gave oral presentations of their research, allowing the panel and other attendees to ask questions of the presenters and to ask questions related to the written descriptions. During the first half of the second day, the panel again worked in isolation, and the attendees took part in various activities related to the general theme of research evaluation. In the afternoon, the panel presented its evaluation to the unit and answered questions.

After the seminar, the panel finalized its written evaluation. In parallel, the research groups produced the final versions of their contributions, generally expanding somewhat the substance of their preliminary descriptions, based on the input from the seminar. It is those updated descriptions that, subjected to some editing, appear as the chapters in Part II of this report. Consequently, the evaluation in Part I and the descriptions in Part II are in some places slightly discordant.

Part I of the report is devoted to the actual evaluation of the research unit. It provides a brief description of the formal context of the evaluation and then presents the panel's evaluation. As mentioned above, Part II then has a chapter for the research unit in general and for each of the research groups. The general chapter describes global aspects of the research unit. It covers the background and organizational context of the unit and the overall organization of the unit's research. It presents the research plan for 1991–1995 and the resources available for implementing that plan. Finally, it evaluates the global administration and

organization of research and presents a plan for 1996–2000. The chapters for the research groups survey the groups research activities and describe their organization and staff. Descriptions of research collaboration, Ph.D. projects, and service and research-related activities then follow. Finally, self-evaluations and plans for the next period are given. Part III provides a brief summary in Danish of the report.

Designing the research evaluation and producing this report was intrinsically a consensual and collective effort involving to varying degrees all the researchers in the unit. However, Mike Manthey deserves special thanks for the time he took to proofread substantial portions of this report. Graphics artist Kent Hansen designed the cover page of the report.

Aalborg, Denmark CHRISTIAN S. JENSEN
May, 1996

Contents

Part I

The Evaluation and its Framework

Chapter 1

Evaluation Guidelines

The guidelines from the Faculty of Engineering and Science that constitute the formal context for the research evaluation are paraphrased below.

1. The evaluation is conducted for each research unit separately and covers the research unit's work since the previous evaluation.

2. The evaluation of each research unit is conducted every five years. The specific time is set by the unit's institute.

3. The evaluation is conducted by an evaluation panel of normally three persons, with at most one from the Institute and not less than one from outside the university. Normally, at least one member must be employed by an international institution. A panel member from the unit has no vote on the panel. The panel is appointed by the Faculty following the recommendation of the Institute.

4. The evaluation includes the following.

   (a) Published research results; (b) other research results, such as results yet to be published and results that are unpublished in a traditional sense, but which are disseminated to industry, public institutions and organizations, etc. through curricula or otherwise; (c) research activities in progress; (d) other research activities, including international research collaboration, participation in scientific congresses, conferences, and symposia, editorship, refereeing, and evaluation of theses and applicants for academic positions; and (e) research collaboration.

5. The evaluation has these objectives.

   First, it is to assess whether there is a satisfactory agreement between allocated internal and external research resources and the research accomplished. Second, it must determine whether the research unit's actual research efforts are consistent with the unit's goals and plans for the period

as they are expressed in the long-term research plans of the Faculty of Engineering and Science. Third, the evaluation is intended to aid the research unit in planing its future research efforts and its organization of research.

6. The evaluation panel prepares a preliminary evaluation that is made available to and is discussed with the unit's staff. On this basis, the final report is prepared and submitted to the Faculty with comments from the research unit and the Institute.

   The evaluation report may be written in English or Danish. If it is written in English, a summary in Danish must be included. Conversely, if it is written in Danish, a summary in English must be included.

7. The Institute organizes and conducts its research evaluation while adhering to the above general guidelines, so that the evaluation fits the best with the wishes and activities of the Institute.

Realizing that a research evaluation is a learning experience that provides a welcome opportunity to rise above the daily routine and reflect upon research, the unit agreed upon an additional objective: Within the research unit, the evaluation—the process as well as the final report—should constructively aid the staff members in evaluating and improving their effectiveness as researchers, group leaders, and administrators of research at the departmental level. In particular, the evaluation presents an opportunity to evaluate the effectiveness of departmental policy.

Chapter 2

The Research Evaluation

Sture Hägglund	Kim G. Larsen	Stig Skelboe
professor	professor	professor
Linköping University	Aalborg University	University of Copenhagen

2.1 The Evaluation Process

The members of the Evaluation Panel are senior computer scientists with essentially disjunct areas of expertise. Together they cover a substantial part of the activities in the Computer Science Research Unit but with varying levels of expertise. This aspect in addition to the limited amount of time assigned for the evaluation mean that most of the evaluation is based on indirect quality measures and not on an independent assessment obtained by reading the research papers produced by the Unit.

The Evaluation Panel had two main sources of information, this report and the presentations by the six research groups at the review seminar, including discussions with the members of the groups. Besides, a small number of key publications from each group were made available to the Panel.

The Evaluation Panel spent a few days preparing for the review, which took place during a two-day seminar in January 1996. During the first day of the review seminar, each of the six groups was given 45 minutes for presenting its research, with at least 15 minutes reserved for questions and discussion. The second day was used by the members of the Panel for a discussion of the presentations and the submitted material. The final phase of this discussion involved only the external members of the Evaluation Panel. The seminar was concluded by a presentation in plenum of the preliminary conclusions and a discussion of the evaluation of individual groups. The ambition of the Panel was here to provide fairly specific comments and recommendations to all the groups participating in the review seminar.

Due to the broad scope of the research covered, as well as the relatively short

time available for the review process, the written evaluation and recommendations from the Panel are not overly detailed and specific with respect to individual researchers and groups. However, the part of the process leading up to the evaluation is considered the more important part of the process. Here the research groups had the opportunity to see the research of five years in perspective, compare previous research plans with obtained results and consider plans and visions for the next five years. The groups' own evaluations in this report are equally useful to the Evaluation Panel as to the groups themselves.

In general, the Evaluation Panel wants to express its appreciation of the assessment process. We are quite impressed with the organization and management of the evaluation activities. It has been a stimulating experience and hopefully well worth the effort for all parties involved.

The present report has been most useful to the Research Panel. However, we should like to draw the attention to the format of the brief accounts describing the individual research areas. These accounts are very terse and virtually incomprehensible to others than experts in the different areas. An alternative to this attempt to cover all activities could be to select one or two activities from each group and write a short (3–5 page) "popular" article to present these activities.

2.2 Evaluation of the Research

The research conducted by the Computer Science Research Unit was evaluated based on a number of different criteria. Below, each criterion is considered in turn.

The quality of research is basically measured by originality, scientific depth and the knowledge of and application of results, tools and techniques from the area. The present evaluation is indirect and reports the ability of the groups to get results published in recognized and refereed journals and conferences where the abovementioned quality criteria are supposed to prevail. The Panel found the quality of the research in all of the groups to be good to high. No running activities were judged to have poor quality while several of the groups exhibit projects with results of high, and in some cases maybe even excellent, international standard.

Productivity and quality are closely related. Considering these two criteria together, the Unit has produced a very satisfactory number of quality publications. Indeed, the productivity is quite impressive and reflects a young and very enthusiastic department.

The productivity in terms of titles varies substantially over the groups, but there are also substantial variations in the number of researchers having been involved (faculty, guests and Ph.D. students) and in the amount of work, notably experimental work, required to produce an article. Finally, titles cover anything from brief papers to entire books. Taking these circumstances into consideration,

the variations in productivity is not a matter of concern.

The relevance of the research can be evaluated against several different measures. Virtually all of the research reported by the Computer Science Research Unit is motivated by practical problems, and the research is either directly applicable for solving practical problems, or it is the background for methods and tools for the solution of practical problems. The relevance of the research for undergraduate education was not much commented upon in the written material, but seems to be good and also a matter of concern for most, not to say all, of the groups.

The Ph.D. programme seems quite successful, with a high degree of completion. All of the groups have Ph.D. students. The numbers have fluctuated over the years, and there are obvious explanations for the fluctuations.

The number of industrial and international contacts by the different groups vary considerably. A large number of external contacts is not necessarily an advantage, but we encourage the research groups to consider the possible benefits of external contacts and to seek more contacts (or reduce the current number of contacts). In several cases, good applications based on external cooperation were presented by the groups. Interestingly enough, some of the most successful applications were demonstrated by groups with a strong theoretical background.

Similarly, external funding gives some benefits but also entails some expenses. Again the level of external funding differs widely within the groups, and they should reconsider the current status. For instance, the BRICS Centre and The Software Factory Programme are good examples of the benefits of externally funded activities. We feel that there is a potential for several of the groups to attract and benefit from increased external funding.

Finally the groups function very differently. Some of the groups seem to work together in a very focused way obtaining a substantial synergy. Other groups seem to work more like the sum of the individual members with very little research interaction. The latter groups should probably try to increase the cooperation within the group. In cases where the group cohesion is low, one possibility to reach critical mass is of course to cooperate over the group borders. There are good prospects for such cooperation which already is in effect in several cases.

2.3 Evaluation of Research Plans

The Review Panel prefers research plans to be visionary and to set goals and directions for future research rather than being detailed. It is in the very nature of research to respond to recent results within the group and the community and to pursue the most interesting and promising problems.

The Computer Systems Group opens its planning section with some attractive intentions of increasing overlap between student projects and research, increasing external funding, increasing external cooperation and improving the local milieu.

The research plan falls into two parts both of which can be considered natural continuations of current activities. The first group including protocols, real-time and distributed systems can be considered low risk topics with a high practical relevance. The second group includes self-organizing distributed systems, a high-risk area with a potentially high relevance.

The Formal Systems Group exhibits a well managed research programme with a good balance between theoretical topics and practical applications, though the vision for the future embodied in the research plan was more vividly described in the oral presentation than in the written material. The proposed undertakings are definitely interesting and well motivated. In particular, the hybrid systems area with interacting discrete and continuous components might provide exciting challenges. The group has a well established cooperation with other research groups, both nationally and internationally, and should be able to attract highly qualified visiting scientists.

The Programming Systems Group has only reported preliminary plans, and they include a continuation of current activities. The plans also mention better contact to the industry to get inspiration and feedback from the practical reality. The plans seem overly modest, and the Panel feels that the Programming Systems Group has a potential for more ambitious research than reflected by the plans.

The Information Systems Group has formulated an extensive agenda for its research in the next five years, the ITEM 2001 paradigm. Compared to its current standing as a highly recognized research group belonging to the Scandinavian school in the information systems field, the plans represent a stronger orientation towards hard-core software engineering. The Panel expressed a certain concern for the dangers involved in trying to cover the broad scope of research areas indicated in ITEM 2001. But the current move in the direction of an increased emphasis on software engineering is well motivated, and the group is comparatively well staffed for such an undertaking.

The Database Systems Group has in a very short period of time established itself as a well recognized group with international visibility in its special area, temporal databases. The plans for the coming five-year period are well focused and involve international cooperation, combining basic research with technology transfer activities. The proposed initiative in the area of data warehousing may be considered to be a bit diverse in this context, but represents an interesting new research area with a potential also for cooperation with other groups in the department.

The written research plans by the Decision Support Systems Group are somewhat sketchy but well focused on important problems in the area. The plans presented at the seminar were considerably more detailed, and they should present ample challenges for the next five years. The cooperation with both internal AAU research groups and external research groups seems crucial for future success. In fact, all the groups within AAU using the Bayesian approach for decision support systems must cooperate in order to reach a critical mass for this activity.

2.4 Discussion

The research in the Computer Science Unit is of course strongly affected by the conditions imposed by the University and by the organization it has chosen for the daily operations. In this section, the Evaluation Panel has collected some thoughts and reflections inspired by the evaluation activities, though in part only indirectly related to the research assessment as such.

Aalborg University does not seem to have a clear strategy for the research conducted by the departments. Such a strategy should not only cover issues related to the selection of research areas and criteria for setting goals, but also aid in putting different and competing ambitions in order of priority. Thus research may be conducted primarily in order to produce new scientific results, with an emphasis on the number and quality of resulting publications. But research may also be seen as an undertaking motivated to a large degree by the needs to provide a foundation and support for undergraduate (and graduate) education. Still another possible motivation for research activities may be the desire to develop, maintain and distribute knowledge and competence to the surrounding society.

It is our impression that the Computer Science Unit assesses its research achievements primarily in terms of its scientific output. However, it is obvious that for instance the educational needs also play an important role for the design of the research organization. We feel that a more explicit discussion of the various research activities in the light of the different ambitions mentioned above would have been helpful. In particular, we believe that the research groups would benefit from having such an elaborated strategy spelled out from the University and the Department.

The persistent teaching overload of the Unit is a problem of concern for the Panel. Although the Unit is compensated in terms of additional funding, the extra teaching jeopardizes the continuity of the individual researcher's work.

The organization of the Computer Science Unit into fairly small groups is seen as a potential danger by the Review Panel, and the "peace treaty" among the groups tends to consolidate this organization. Unless there is a certain mobility among the groups, it may result in stagnation and inhibition of new research ideas and directions.

However, the purpose and the advantages of the organization at a given time are clear, e.g., when teaching is assigned. If emphasis is on group cooperation with due respect for the individual researcher's right to pursue alternative interests, the organization may be viable also in the long run.

The Evaluation Panel has noted some peculiarities in the internal organization. Some of the groups exhibit what appears to be a strong leadership with a focused research profile, while other groups seems to be more loosely organized and defined by the individual researchers currently belonging to the group. This is understandable, given the way research is funded within the department, but it is felt by the Evaluation Panel that a more distinct leadership might help some

of the groups to make good use of their research potential.

It is also felt by the Panel that there is a potential for increased external funding and cooperation in several of the groups. Again it would be important to have a more clear statement from the University regarding the appreciation of such efforts. For the individual groups, such funding would result in a more rapid feedback concerning the relevance and quality of their research. Likewise, external funding might contribute to provide active partners in more applied research. In the long run, more emphasis on external funding might help to handle the process of promoting the necessary renewal of research.

Although presumably outside the scope of this evaluation, it was noted by the Panel that the Computer Science Unit, but to a very limited degree, covers areas related to human factors and cognition. We understand that teaching in these areas belong to another department, but there is a general trend in the direction of creating interdisciplinary IT research environments where mathematical rigour can be combined with for instance usability engineering. Considering the increasingly important issues concerning the interplay between the human and the machine, it might be wise to reconsider the border lines between departments.

2.5 Conclusions

The Computer Science Unit at Aalborg University has documented fully satisfactory research activities for the five-year period. All of the six research groups show good to high performance with respect to quantity and quality of published results. The Ph.D. programme seems to be working with high efficiency and a large percentage of the students complete their exams in time. The interaction with undergraduate and master's students is less well documented, but appears to be functioning well. International contacts are good in several areas, though the amount of formalized international cooperation is not very high. Industry contacts and other external cooperation is satisfactory, but may be increased.

The plans for the coming five-year period could in some cases have been a bit more visionary regarding the long term goals of the research, but are in general well balanced, realistic and relevant. The Computer Science Unit is still in an expansive phase with comparatively young and enthusiastic researchers pursuing topics close to the research frontier. The relatively slow and careful build-up of faculty has resulted in a situation with a good age distribution and well functioning activities. However, some groups are in fact under-critical in terms of size with respect to the topics studied, and the persistent teaching overload makes this situation even worse.

The evaluation process as such has in our opinion been a success, and the Evaluation Panel is very satisfied with the material and presentations provided.

Part II

The Unit's Research Report

Chapter 3

The Research Unit

This chapter gives a general description of the computer science research unit with respect to topics, history, resources, and organization. Furthermore, the chapter gives a general description and evaluation of the unit's research plan for the period 1991–1995. Finally, a plan for the period 1996–2000 is sketched.

3.1 Introduction

The research of the computer science unit focusses on computers, auxiliary computer equipment, and computer systems. The scientific approach embraces the formally logical, the experimentally constructive, as well as the empirically descriptive.

Programming and programming languages constitute vital links between computers and human problem solving, and the study of these is fundamental. Computer science is not least concerned with how to carry out effective and quality-based program development, with design and implementation of formalized, linguistic means of expression, as well as development and application of program development tools in general.

Concurrently, the computer science unit is engaged in the exploitation and development of the possibilities created by computer techniques—e.g., utilization of parallelism by interlinked computer systems—as well as in the development and adaption of concrete systems to specific applications.

The area has immediate bearing on mathematics and logic, on engineering, and to some extent on the humanities and social sciences.

Seven central computer science domains are covered by the research unit: computer systems, formal systems, programming systems, information systems, database systems, decision support systems, and applied computer science.

3.2 Background and Working Conditions

We first describe the local organizational context of the research unit, then describe the history of the unit.

3.2.1 Computer Science at Aalborg University

Aalborg University (AAU) was established in 1974 and is the newest university in Denmark. It consists of three faculties.

- The Faculty of the Humanities
- The Faculty of Social Sciences
- The Faculty of Engineering and Science

The Faculty of Engineering and Science is at present divided into 9 institutes. One of these institutes actually also belongs to the Faculty of Social Sciences.

The Institute of Electronic Systems is by far the largest institute within the Faculty of Engineering and Science. This institute has a scientific staff of more than 100 (professor[1], associate professor, and assistant professor), and moreover approximately 100 researchers employed on contracts. It is divided into 4 departments: (i) Communications Technology, (ii) Medical Informatics and Image Analysis, (iii) Control Engineering, and (iv) Mathematics and Computer Science. The researchers in the first three departments are mainly electrical engineers. The Department of Mathematics and Computer Science's researchers have very mixed backgrounds and broad research interests. They include mathematicians, statisticians, and computer scientists; and moreover, some have a background or research interests at the border of natural sciences and the humanities or social sciences.

With respect to research, the Faculty of Engineering and Science has subdivided the Department into two planning units, Computer Science and Mathematics, which at present have staffs of about equal size.

Decisions about budgets, staffing, and teaching are presently taken at various levels. Briefly, the Faculty decides the budget for the Institute, based on the budget it gets from the Ministry of Research and Education, and the Department can, via the Institute, influence the budget by, e.g., proposing new positions, etc.

Decisions about study programs are taken by separate, local curriculum committees (in Danish, "studienævn"). These bodies consist of equally many staff members and students. Studies in mathematics, computer science and engineering, physics, and optical engineering are administered by separate bodies.

[1] In this report, we use the terms (full) professor, associate professor, and assistant professor for the Danish-language terms professor, lektor, and adjunkt, respectively.

Comparing AAU with the other universities reveals a significant difference: AAU is the only university in Denmark that offers degrees in both the engineering and natural sciences.[2] Only a few of the natural sciences are, however, present at Aalborg University: Mathematics, Computer Science, and Physics.

3.2.2 History of the Computer Science Unit

The history of the unit goes back to 1976, when a minor (i.e., a B.S.) in computer science was established in combination with a major (i.e., an M.S.) in mathematics. In the early eighties, a major in computer science was established, with associated minors in both mathematics and electronics engineering. From this early beginning and until 1990, the number of M.S. and M.E. (Master of Engineering) candidates grew from less than 5 per year to approximately 40, and the number of staff (full, associate, and assistant professors) grew from 4 to 8. The first full professor was appointed in 1987.

From the mid-eighties to 1993, the Ministry of Research and Education reserved special budgets for computer science and engineering at all Danish universities. This meant that the research unit was able to initiate a relatively high volume of Ph.D. studies (3–4 per year), and it meant that assistant- and associate-level positions were granted by the Faculty on request. At the end of the present period (1995), the staff consists of 20 full, associate, and assistant professors.

Research Groups

In 1987, the research staff divided itself into five research groups (Information Systems, Programming Systems, Formal Systems, Computer Systems, and Applied Computer Science). The main purpose was to make the unit visible through its research profile and also to de-centralize the responsibility for course planning and teaching. In 1990, a Database Systems Group was added, and in the middle of the present period, a seventh group, Decision Support Systems, was established. The research groups form the basic social environment for the individual researcher, and the groups are therefore also the basic fora for strategic discussions. All groups assemble each week for a one-hour meeting. These meetings are moderated by representatives from different groups (at present, three representatives).

Teaching Computer Science

At Aalborg University (as at all Danish universities), research and teaching at the university level are intimately connected, and we feel it appropriate to include

[2] This again is quite different from American universities that often have a College of Engineering.

a brief description of the teaching of computer science at this university.

The teaching duties of the staff in computer science were and are mainly directed towards the following areas:

1. The basic year at the Faculty of Engineering and Science that initiates all the students of this faculty to scientific studies, and in particular to the pedagogical model used at Aalborg University.

2. Computer science at the bachelor level (for students of mathematics, computer science, computer engineering, and physics), at the master's level, and at the Ph.D. level.

Aalborg University employs a very particular pedagogical model, only found in few places around the world. Every term has a particular theme, and the studies are structured as follows: The students divide themselves into groups of 5–8, and each group is assigned a full-time working room. Approximately half of the time, the students work in the groups on topics within the theme chosen for the term. It is the group and its advisor that agree on the particular topic. The other half of the time is taken up with courses of a more traditional form. About half of the courses are relevant to the projects, and the remaining courses form part of the general education. Each group is assigned one or more staff members as advisors for the project. Staff members serve as both project advisors and lecturers for the courses.

Teaching at AAU usually means both advising one or more groups, and giving lectures (including the problem solving sessions). More specifically:

- Courses are usually given in four-hour blocks, consisting of two hours of lectures and two hours of supervised problem solving sessions. Such blocks usually account for 14 hours each[3], and includes preparation, design of exercises, briefing teaching assistants, teaching administration, oral exams at the end of the term, etc.

- Supervising the project work of a group of students for one term would typically account for 100–170 hours. This includes preparation, introduction to the students, weekly meetings, correction of working papers, correction and grading of the project report, and an oral examination with individual grading.

Research and Teaching

The main part of our candidate production consists of five-year M.S. or M.E. degrees in computer science (see the table that follows). Towards the end of the fourth year, the students attend a series of seminars offered by the research

[3]So, a typical 15-week, once-per-week course equals $15 \times 14 = 210$ hours.

groups. Based on these seminars, the students choose an area of specialization for the final year within one research group. The project work during the final year is carried out in close collaboration with the chosen group, whereby the close relation between research and teaching is assured.

Ph.D. students devote the full three-year period to a particular research project that is within the interests of one of the groups. This work is supplemented by additional advanced courses and normally also by a six-months visit at another university.

Candidate Production

Below, the candidate production is shown for the years 1991–1995. The decrease in M.E.'s for 1992 is due to the transition from one study plan to another.

Candidate Production					
	1991	1992	1993	1994	1995
M.S.	27	20	21	10	11
M.E.	24	7	19	22	24
B.S.	0	0	13	9	5
B.E.	3	0	1	0	0
Ph.D.	0	2	5	3	1

3.3 The Unit's Research Plan for 1991–1995

We do not have an explicit research plan for 1991–1995. However, based on the previous research evaluation of the unit (1986–1990) and the latest strategic plan from The Faculty of Engineering and Science (1989–1993), we have retrieved the following goals.

- We want to cover a variety of research areas, based on strategic and scientific considerations. However, the expansion of the four original groups (Formal, Computer, Programming, and Information Systems) to satisfactory level is given priority.
- Three of the original groups (i.e., DS, PS, IS) should have more publications in high-quality contexts.
- Each research group should develop more detailed research plans.
- The groups should continue to make themselves as visible as possible, e.g., through external guests and visits, publications, and conference participation.
- The permanent scientific staff should not be expanded too fast.

- The research groups and their profiles should not become institutionalised.

We did not formulate an overall plan to fulfill the above goals. However, we have followed the following principles throughout the period.

- The individual research groups are responsible for quality assurance. Also, the groups must themselves decide which research subjects should be given priority.
- High quality must be assured when permanent positions are allocated.
- The general resources from the Faculty should be distributed in a way that guarantees a certain minimum amount of resources to each researcher.
- In order to support the consolidation of the smaller groups, we have decided to initiate new major research areas through external collaboration rather than by forming new groups.

3.4 Resources Allocated to the Unit

This section covers the resources allocated to the unit as a whole. Later sections cover resources allocated to individual research groups in more detail.

3.4.1 Overview of Staff

Figure 3.1 shows the research staff for the period 1991–1995. Each number identifies our resources measured in full time positions. The percents indicate the amount of research time allocated per full-time position. The numbers in parenthesis in the row for Ph.D. students show the number of externally funded Ph.D. students. We note that the unit has been overloaded by teaching duties throughout the period. More precisely, the teaching for each professor (full/associate) has been approximately 30% above the official norm of the Faculty, that is, professors have been teaching 65% instead of 50% of their time.

3.4.2 Working Grants from Faculty

Below we show the yearly working grants (in DKK) provided by the Faculty of Engineering and Science. The figures exclude grants dedicated to salaries. The table clearly shows that the payment for teaching overload forms a substantial part of the available resources.

Faculty Grants					
	1991	1992	1993	1994	1995
Regular Grant	800.000	850.000	900.000	925.000	975.000
Teaching Overload	375.000	400.000	425.000	475.000	500.000

Teaching Positions					
	1991	1992	1993	1994	1995
Full Professors, 40%	1	1	2	2	2
Associate Professors, 40%	9	10	10	11	12
Assistant Professors, 50%	5	6	6	6	7
Teaching Assistants, 20%	1	1	2	2	2
Research Positions					
Visiting Professors, 100%					1
Visiting Researchers, 100%	1	1.5	3	2.5	2
Research Assistants, 100%	0.5	1	2	2.5	1
Ph.D. Students, 85%	10.5(1)	11.5(1)	10(1)	8(3)	11.5(3)
Administrative Positions					
Secretaries	3	3	3	3	3
Technicians	2.5	2.5	2.5	2.5	2.5

Figure 3.1: Overview of Research Staff 1991–1995

3.4.3 External Grants

Figure 3.2 gives an overview of the external research grants obtained during the evaluation period.[4]

3.5 Research Administration and Organisation

3.5.1 General Administration

As mentioned previously, the computer science research unit forms a department in association with the research unit mathematics. This department is one of four departments in the Institute of Electronic Systems. The unit is led by a small number (currently three) of permanent staff members, which are elected every third year. All staff members meet at an informal coordination discussion once a week. The unit is subdivided into seven research groups, and each group has a well-defined research profile. In order to simplify course planning, each group is responsible for teaching certain courses in the computer science and

[4] We apply the following abbreviations for the groups: Formal Systems (FS), Computer Systems (DS), Programming Systems (PS), Information Systems (IS), Database Systems (DBS), Decision Support Systems (DSS).

	1991	1992	1993	1994	1995
Danish Natural Science Research Council (DBS)				75	125
Danish Natural Science research Council (DBS)		91			75
Danish Natural Science Research Council, "Organizing" (PS)				142	
Danish Technical Research Council, DaCapo (PS)	400	400	400		
Danish Natural Science Research Council, The Software Engineering Program (IS, PS, FS/DBS)	300	400	350	350	
Danish Natural Science Research Council, High Speed Networks, (misc. equipment, DS)					524
Danish Natural Science Research Council, The Software Factory Program (PS, DBS, IS)					500
Danish Research Councils, PIFT Program, DART (FS)	900	900	900		
Danish Research Councils, PIFT Program, ODIN (DSS)	1000	1100	1200	1300	500
Esprit BRA CONCUR2 (FS)			300	300	300
Esprit BRA DRUMS (DSS)		150	150	150	
Danish Technical Research Council, Environments for Real Time Systems (DS)			125	570	570
Danish Research Foundation, Centre of Basic Research, BRICS (FS)				1600	1600

Figure 3.2: Overview of External Research Grants

engineering programs. Our administrative support includes three secretaries and two technicians.

Each research group has its own computer laboratory with both ordinary and special equipment. These laboratories are mainly used by M.S. students. Furthermore, each staff member has a personal work station.

3.5.2 Research Groups

Research in the computer science unit is split into the following seven groups.

Programming Systems, covering the design, implementation and application of programming languages, their environments, and tools.

Computer Systems, covering in particular those aspects of programming languages, environments , techniques and tools, that are relevant for architecture-dependent subjects like distributed systems and networks.

Formal Systems, covering formalisms for the construction of computer systems.

Information Systems, covering in particular the development and application of computer based systems in organisations.

Applied Computer Science, covering applications of contemporary information technology based on established computer science theories and methods.

Database Systems, covering data organisation, and maintenance, as well as techniques and tools for data access.

Decision Support Systems, covering techniques and tools for decision support systems. The work is primarily based on Bayesian nets.

3.6 Own Evaluation of Research Administration and Organization

At the beginning of the period, the permanent staff formed less than half of the teaching group. This implied that assistant professors had to bear a number of administrative tasks. Therefore, the primary task during the period 1991–1995 has been to make sure that the fraction of permanent staff exceeds its critical size in a way that consolidates the individual research groups. Also, it has been a subtask to guarantee sufficient basic funding for travel, equipment, and technical support.

As seen from our point of view, the above task has only partly been solved in a satisfactory way: The four original groups now have at least two members of the permanent staff, and all groups now have a reasonable number of resources for both research and teaching. Also, the number of permanent positions is now sufficient to handle research administration without major problems. Furthermore, we consider the combination of research groups and weekly coordination to be a good basis for future organization. *However,* our teaching duties have grown at the same rate as the number of permanent positions. As mentioned previously, this means that the amount of teaching has remained constant throughout the period (approximately 30% above the official norm). This situation is very unsatisfactory.

3.7 The Unit's Research Plan for 1996–2000

We have recently completed our proposal for a strategic plan for 1995–2000. Its main goals are described next.

- The number of research groups should not be changed. Rather the existing groups must be consolidated. New research topics should be initiated through external or internal collaboration. Especially, the possibilities for collaboration within the Institute of Electronic Systems and with industrial partners should be exploited. The leadership of the unit can be instrumental in bringing about such collaboration. However, each individual research group should decide on the final priorities among its potential research activities.

- Due to the growth of permanent positions, an additional full professorship should be allocated.

- The technical platform (computers and network) must gradually be renewed and expanded (in the case of insufficient national funding).

- There should be a controlled growth in the number of permanent positions.

- The problem of teaching overload *must* be solved. We note that hiring more permanent faculty conflicts with the previous goal.

- Assuming that the problem of teaching overload will be solved, the annual extra income (approximately DKK 500,000) from this activity will vanish. This implies the need for at least a doubling of the external funding in order to maintain the quality and update of the technical platform (workstations and network).

Chapter 4

Computer Systems

4.1 Profile

The Computer Systems Group covers the areas of low-level system software, operating systems, computer architecture, computer networks, and the application of concurrency and parallelism in a contemporary technological setting. It is perhaps the group that is singly most heavily influenced by contemporary technological realities and trends (e.g., computer networks and multi-cpu computers). In spite of its ultimate interest in application, the group strives to maintain a close contact with theory and collaborates with the Formal Systems Group.

The history of the sub-field of computer systems goes back to the mid-1950's with the development of the first assemblers, loaders, and compilers. Nevertheless, the theoretical and practical issues which now dominate and permeate everything (and have as well invaded much of the rest of computer science) first appeared in the mid-1960's, with the first operating systems, and especially time-sharing systems. The basic issue can be succinctly characterized as the difference between having a single—"sequential"—program, and having two which interact (if they don't interact, then there is nothing new). This may seem a minor difference—and for most computer scientists until about 1980, it did so seem—but in fact the difference has caused a fundamental change of both the theories for the semantics of programs and the pragmatics for program construction. This change is as revolutionary and conceptually challenging as the shift from classical to modern physics that occurred in 1905–1927.

It turns out that the problem of time-sharing a single computer between N programs is a special case of having N (interacting) programs, each on its own computer. And it is a fact that powerful technological and economic forces together are rapidly driving computer systems in the direction of many interconnected computers, from several to hundreds of thousands. Such systems are generally of two forms: (1) workstation/PC networks, which facilitate sharing of resources and cooperation between geographically distinct parties, and (2)

multi-cpu computers, for high-speed and intensive (often numerical) computation. Experience with these two kinds of systems, running relatively traditional programs, has in addition given birth to (3) truly distributed computations in which one strives to achieve globally coherent behavior in terms of purely local computations. Examples are air-traffic and process control systems.

While the computations which take place on all three kinds of systems are often quite different, the underlying and supporting system software for all of them displays the same qualities and problems, both theoretically and practically. We will henceforth refer to all such multiple, interacting computations as "distributed."

A short list of these qualities, issues, and problems includes:

Qualities:

- non-determinism ("unpredictability")
- non-repeatability (different behavior on successive "runs")
- a Heisenberg-like uncertainty principle (observation disturbs the future)
- conceptual complexity combinatorially greater than for sequential systems
- emergent phenomena (combining pieces can produce new behavior)
- relativity of time frames is a technological reality

Issues:

- how to control/exploit non-determinism
- how to test a distributed system, given non-repeatability and uncertainty
- how to control conceptual complexity and harness emergence
- how to design software that suitably hides/reveals these qualities

Specific Research Problems:

- how best to program distributed systems
- implementation of theoretically well-founded testing tools
- special issues of real-time systems
- protocols for high-speed data communication
- self-organizing distributed systems

This list is biased by the interests of the three faculty members, but is nevertheless in many ways characteristic of the field of distributed systems. In the following, we will describe our investigation of the above-mentioned research problems. We emphasize that, despite intense activity on all fronts, there is virtually no broad consensus on the "best solutions" to any of the items in the three lists.

4.2 Activities and Results

4.2.1 Programming Distributed Systems

For historical reasons, both at AAU and generally, research into programming paradigms for distributed systems is relatively divorced from its parent discipline of programming language research. The two lines of research tend to pursue issues which are mutually orthogonal. In particular, the world of distributed programming demands particular attention to such matters as synchronization, mutual exclusion (i.e., resource sharing), communication between processes, process creation and termination, distribution of the computation over multiple computers or insulation from same, matters of real time, etc.

We focused in the period 1991–1995 primarily on a particular paradigm for programming distributed systems called "Linda", which originated with Gelernter and Carriero (currently at Yale University). Linda is not in itself a language, but rather a small set of "primitive" facilities which can be embedded into any given sequential programming language. The result is a new language which Linda proponents claim is particularly well-suited to the demands of programming distributed systems. We found the Linda concept interesting and promising, and decided to gain experience with Linda via implementation.

Linda is simple to explain. It consists of a conceptually global memory—called "tuple space"—and four operations which add or remove tuples from this space. A "tuple" is a list of typed data items, and can represent a record of data, a list of data items, a row of a matrix, a parameter list, etc. The Linda primitive which adds a tuple to tuple space is called `out`, e.g., `out('foobar', real 3.2, integer 382)` places the items in the parentheses together in tuple space and the program issuing the `out` continues execution.

There are two ways to remove a tuple from tuple space. The `rd` primitive will return to the program which issues it a copy of a tuple in tuple space which "matches" the tuple in the `rd` statement. For example, `rd('foobar', real ?X, integer 382)` will match—in terms of number of items and their respective types and values—the tuple `out`'ed in the previous paragraph. If and when such a match succeeds, a copy of the matching `out` tuple will be returned with free variables (`?x` in the example) bound. In the example, the program issuing the `rd` will proceed with `x` bound to the value `3.2`.

If tuple space does not currently contain a tuple which matches the tuple in the

`rd` operation, the issuing program will be suspended until this occurs. Therefore, no matter in which order otherwise matching `out`'s and `rd`'s are issued, the two parties involved will be properly synchronized. Said in other words, they are decoupled from each other in both space and time. This property of Linda is particularly valuable for freeing programs from dependence on the underlying multicomputer topology, thus contributing strongly to program portability.

The third Linda primitive, `in`, is like `rd` except that instead of returning a copy, it removes the tuple it matches from tuple space. It is in addition guaranteed that if there are two `in`'s competing for the same tuple, only one of them will succeed. This property provides "mutual exclusion," which is a fundamental necessity in distributed programming.

The fourth and final Linda primitive is `eval`, whose tuple provides initializing parameters for a process to be therewith created. The processes so created may in turn issue any of the same four primitive operations. Thus `eval` is the means which Linda provides for creating the multiple processes which are the sine qua non of distributed programming.

It is easy in Linda to express any of the popular forms of distributed programs. Indeed, it is hard to fault Linda's expressibility. The concept of a global tuple space, which in reality is distributed over the memories of the individual computers comprising the system, together with the associative matching, is both powerful and general.

In 1990 we and our students produced a compiler for C++Linda and an accompanying run-time system for our 16×4Mb transputer. This system was used to implement the first Topsy system (1993, described later), and Linda's approach is still key to subsequent implementations thereof. Keld Kondrup continued the inquiry into Linda by designing an extension to multiple tuple spaces, transactions, and processes as "first-class citizens". This work [33], of a highly formal but practically oriented character, closed the circle regarding "persistent objects." In sum, we (and many others) have found that our initial interest in Linda was justified, and it continues to influence our work. At the same time, it is no longer a focus, but rather has simply been added to our conceptual furniture, to be used or gather dust as circumstance dictates [30].

Having investigated the Linda paradigm to its satisfaction, the group turned its attention to underlying mechanisms for supporting both Linda and other embodiments of parallel and distributed computation. In particular Keld Kondrup advanced the idea of a low-level "generic target language" (GTL) and associated (interpreted) machine. The intent with GTL was to provide a single target machine, suited of course for distributed systems, for *any* language compiler—e.g., Linda and real-time specification languages—in order to decouple our language and tool development from that of execution environments, as well as to promote the portability of both. The result, implemented on both our Suns and the transputer, was an interesting blend of data-flow and global memory which was used in two master's projects. A third project, based on Mike Manthey's ideas,

designed a novel subscription-oriented communication system based entirely on the concept of co-occurrence of data in a global tuple space. We hope that an implementation will result in the coming period.

4.2.2 Testing Distributed Programs

The overall aim of this activity is to derive testing suites from formal specifications of distributed programs. At the beginning of the period 1991–1995, Larsen and Skou had shown how to derive such tests—assuming the availability of a copying facility, i.e., it is assumed that a copy of the implementation system state can be saved at arbitrary points [6]. Applying this, they showed how to derive tests which could test for bisimulation equivalence between the specification and the implementation with an arbitrary level of confidence.

The copying facility may be an unrealistic assumption in many situations. Sharp and Skou therefore loosened it in several ways, and they showed how to generalize the bisimualtion result to both trace and failure equivalence. Also, they combined their results with a notion of coverage of the specification by the given test suite [32].

4.2.3 Environments and Paradigms for Real-Time Systems

Two projects in this area were pursued in this period, Emma (by Lunau) and New Programming Paradigms and Environments for Real-Time Software Development (by Skou). These are described in turn.

Emma is part of the ATOMOS project financed by EEC, and was active in the period 1993–1994; its total funding was DKK 2.2 million. Emma is an Emergency Management System for use on board ships. Emma constantly monitors a large set of sensors and is able to detect damage to the hull and fire on-board. When am emergency has been detected, Emma provides decision support to remedy the situation. Emma has a hypertext-based graphical user interface, and we also gained experience with reflective object-oriented programming and reflective architecture. We built a reflective architecture to monitor the sensors and the graphical user interface uses reflection to determine which object should be visible on the screen. Reflection makes it possible to separate the application into a domain part and a monitoring part, and makes it easy to adapt an application to a changing environment. Emma was carried out in collaboration with the Danish Maritime Institute (located in Lyngby, Denmark) and Lloyds Register of Shipping (located in London, England). The Emma project resulted in one technical report [31] and five articles [12, 13, 20, 21, 26].

The New Paradigms project focuses on *new programming paradigms* and *environments* for *software development* of industrial systems with special attention

to the problems related to reliability, distribution, and real-time [15, 16, 17]. It is a continuation of a six-month pilot project (funded by the Danish Technical Research Council) dedicated to the identification of industrial problems through personal contacts with industry and case studies. The pilot project identified three major industrial problems that are also unsolved research problems.

- There is a lack of systematic methods to handle *component faults* during system operation. This holds both for the derivation of *the component properties* to be monitored during system operation, and also for the actual system reaction when faults occur.
- There is a lack of well-founded environments for *testing* distributed real-time software in a systematic way, such that all possible errors are likely to be found.
- Industrial systems often consist of a *large number of components*. This implies that their combined behaviour cannot be fully analysed without support from automated tools. Such tools do not exist today—although promising work is under way.

This activity aims to contribute to solutions of the first two of the above research problems, and to demonstrate results through practical applications. This will be done by extending existing results on testing and real-time validation as well as contemporary paradigms and environments developed by the participating research groups. The activity is funded by the Danish Technical Research Council through two Ph.D. scholarhips for the period 1994–1997, and is coordinated by the Computer Systems Group in collaboration with the Distributed Systems Group at Copenhagen University. Industrial partners evaluate the work through yearly meetings.

The testing part of the activity includes the design and implementation of a real-time programming language for distributed systems. As a result of the pilot project, the basic primitives of such a languange have been designed [28].

4.2.4 Protocols for High Speed Data Communications

This project is part of a collaboration between researchers at DTU (Technical University of Denmark), KU (University of Copenhagen) and AAU, whose overall aim is to investigate and demonstrate the applicability of modern high speed data communication technologies for distributed systems. Our work is supported in 1995 by the Danish Natural Science Research Council through an equipment grant. At AAU, we are focusing on protocol analysis of the ATM technology. More specifically, the following activities have been carried out:

- Identification of the performance characteristics of existing transport protocols through experiments.

- Initial development and experiments on improved transport protocols.

So far, these experiments indicate the following [29]:

- The actual throughput rates for existing protocols are far below the theoretical maximum. This fact has already been demonstrated for Local Area Networks (LAN), where the physical distance between computers is below 1km, and it is foreseen that the problems will become substantially worse for Wide Area Networks (WAN).

- Modern network technologies allow one to guarantee a maximum transmission delay. However, current experiments show that existing protocols need substantial improvements in order to provide this service to applications. We have designed initial protocol improvements, and our preliminary results are promising. It is likely that new applications, such as video conferencing, will require the introduction of completely new protocol techniques in order to perform well.

4.2.5 Self-Organizing Distributed Systems

Our research over the past ten years into basic principles of distributed systems and their application to the planning and learning problems of artificial intelligence (AI) have recently begun to bear tangible fruit. We are now confident that we have a solid, workable, and extremely promising approach, are working on an "industrial strength" version of the basic software, and seeking funding to perfect, experiment with and distribute this software; and as well to consolidate our position and communicate our results to the scientific community. Due to the very general nature of the problem, and our results, we expect this work to appeal to a broad spectrum of interests, both academic and industrial, basic and applied, and including the computational, physical, life, and social sciences. The approach, while unique in many respects, nevertheless reveals deep affinities with various areas of mathematics (such as topology) and physics (quantum mechanics) not usually associated with computing.

This research program, although couched in the following in terms of AI issues, should be understood by the reader to apply equally to distributed systems in general. We call the conceptual core of this program *the phase web paradigm*; this paradigm is embodied in a running program called Topsy. We currently stand in the following position:

- We have an entirely new model for distributed computation: *everything* is process, there are no "data structures" at all, and very nearly no "algorithms."

- We will soon be able to back up our claims regarding our paradigm and Topsy's abilities with actual demonstrations, which in turn means that we can begin to publish seriously.

- We need to expand the circle of people who work with Topsy, both into AI and other domains: robotics, real-time and factory-automation systems, neuro-sciences, biochemistry, psychology, and others.
- We need to attract at least one Ph.D. student for continued work in this area.

The key ideas of the phase web paradigm, and therefore the principles on which Topsy is based, are two:

> **Extract information** from the environment via observation of *co-occurrences* of the values of sensors.
>
> **Build categories/actions** based on the *co-exclusion* of complementary co-occurrences.

These two principles reflect the basic synchronization patterns "occur together" (true concurrency) and "can/do not occur together" (mutual exclusion). Since these are as well the only possibilities, it follows that everything Topsy *can* know can be expressed in these terms. All of our research, and common sense as well, argues that this is the case.

The use of sensory co-occurrences as the fundamental input to Topsy means that Topsy *learns from its experience*. Furthermore, new experiences are immediately and fully integrated into the program's memory, automatically.

In so doing, Topsy resembles the *neural net* approach, but differs otherwise fundamentally from the latter, in that the phase web paradigm is explicitly "symbolic" and hierarchical; there are no "weights" to be adjusted; and hence only a hand-full of repetitions is necessary to "cement" the knowledge.

Our approach is kindred to that of Brooks and his group at MIT, in that it is decidedly sensor-based. On the other hand, the particular "subsumption architecture" which Brooks must put in *ad hoc* by hand occurs automatically in Topsy as a direct result of its basic functioning. On could say that we provide the theoretical underpinnings and a rather broader canvas on which Brooks' work neatly fits.

Similarly, one can find a parallel to Topsy in the so-called "blackboard" strategies popular in the 1980's, and again, our approach is much deeper conceptually and thus places this work in perspective. The same applies to Shanks' *trans* models.

Finally, it became obvious in the course of the 1980's that the traditional approach, "first plan, then execute," was simply not viable (and indeed Brooks' work was a reaction to this). A truly remarkable property of the way Topsy's knowledge is obtained and stored is that the same structures are used both for categorization and for plan execution, and both of these automatically can (and do) occur simultaneously, intertwined in each other, at any and all levels of abstraction [24]. Furthermore, learning also occurs at the same time, e.g., learning from

mistakes. Finally, the underlying structure *automatically* handles inter-process synchronization.

In short, the phase web approach and resulting software structures in Topsy are *considerably* more sophisticated than what is otherwise to be found in contemporary AI. Indeed, we are *certain* that we have in hand new and fundamental insights into the realms of perception, learning, cognition, and generation of coherent distributed behavior.

As though this were not enough, we have discovered that the *style* of computation represented by Topsy can be directly described—not by the usual automata and logic—but by classical mathematical tools such as vector algebra and discrete algebraic topology. Hence an entirely new, and deep, relationship between computing and mathematics is established [22, 23, 25]. It is particularly intriguing that not only do the fundamental characteristics of 20^{th} century physics have many analogs in the phase web's world of discrete synchronization structures, but that they are both (!) described by the same vector-oriented algebra, a Clifford algebra.

4.3 Organization and Staff

The staff of the Computer Systems Group currently consists of associate professors Mike Manthey and Arne Skou, assistant professor Charlotte Lunau[1], and Ph.D. students Brian Nielsen, Birgitte Lønvig, and Peter Krogsgaard Jensen.

Arne is informally considered to be shared 50/50 by the Computer Systems and Formal Systems Groups. Since the Computer Systems Group is currently normed to three full-time members, this means that the group is currently understaffed.

The group as currently constituted covers most of its teaching at the advanced level. The exceptions are the introductory course in assembly language programming, covered by faculty from another department in the Institute; and networking technology, covered by Wladyslaw Pietraszek, our department's extremely competent system administrator.

It is important to understand that, for both inherent and contextual reasons, the Computer Systems Group is necessarily oriented toward the actual realization of working system software. At the same time, the software in question is large in scope, general in application, and complex and time-consuming to implement. In addition, most of our students are interested in, and will make a career of, designing, implementing, or maintaining such software.

These considerations have resulted in our collaborating closely with our advanced students in our research to produce successively improved and more complete versions of sophisticated and state-of-the-art software for the programming

[1]Replacing Keld Kondrup Jensen, who completed his Ph.D. in the period and was assistant professor for two years therein.

and application of distributed systems. This strategy for software creation forms both a pedagogical stepping stone for the students and proofs of principle for the staff.

The following table provides an overview of our staff during the evaluation period. Numbers shifted left and right indicate spring and fall semesters, respectively.

Teaching Positions					
	1991	1992	1993	1994	1995
Associate Professors	1.5	1.5	1.5	1.5	1.5
Assistant Professors		1	1	1	1
Teaching Assistants	1		1	2 1	
Research Positions					
Research Assistants				1	
Ph.D. Students	1 3	2	2 1	1	1 3

Charlotte Pii Lunau

Academic Degrees

1989	Ph.D. in Computer Science, University of Copenhagen
1982	Cand.scient. (M.S.) in Computer Science, University of Copenhagen
1979	Bachelor in Sociology, University of Copenhagen

Positions

1994–	Assistant professor, Department of Mathematics and Computer, Science, Aalborg University (AAU).
1993–94	Research associate professor, the EEC ATOMOS project, AAU
1993–93	Research assistant, ATOMOS, AAU
1992–93	Teaching assistant (amanuensis), 6 months, AAU
1992–92	Research assistant (3 months), AAU
1989–90	Research assistant, supervising M.S. students, CS Department, University of Copenhagen
1988–89	External associate professor vikar (part-time position), CS Department, University of Copenhagen
1988–92	Project leader at an EEC funded research project at Søren T. Lyngsø, Hørsholm
1984–88	Ph.D. Scholarship, CS Department, University of Copenhagen
1984–85	Visiting scholar, University of California, San Diego
1982–84	System developer at NCR's development center, Copenhagen

Michael Manthey

Academic Degrees

1980	Ph.D. in Computer Science, SUNY/Buffalo
1972	Ab.D. in Computer Science, SUNY/Buffalo
1969	M.S. in Computer Science, SUNY/Buffalo
1966	B.S. in Mathematics, Rensselaer Polytechnic Institute, Troy, NY

Positions

1988–	Associate Professor, Department of Mathematics and Computer Science, Aalborg University
1986–87	Visiting Researcher, Computing Research Laboratory, New Mexico State University, Las Cruces NM
1980–85	Assistant Professor, Computer Science Dept., University of New Mexico, Albuquerque NM
1978–80	Director of CS Computation Lab, SUNY/Buffalo
1972–78	Assistant Professor, Computer Science Dept., Aarhus University; Aarhus, Denmark
1969–71	Real-time Systems consultant to US Navy (Aegis), Comptek Research, Buffalo NY
1966–67	Scientific programmer, General Mills Inc. Minneapolis, MN

Arne Skou

Academic Degrees

1990	Ph.D. in Computer Science, Aalborg University
1975	Cand.scient. (M.S.) in Computer Science, Copenhagen University

Positions

1989–	Associate professor, Department of Mathematics and Computer Science, Aalborg University
1986–89	Ph.D. Scholarship, Department of Computer Science, Århus University. Visiting University of Sussex, UK, for 7 months during the Ph.D. study.
1986	Research assistant, Copenhagen University.
1975–86	Member of the senior scientific staff at the computing center of Aalborg University. Head of the Computer Networking Group.

4.4 Collaboration

Charlotte Lunau has collaborated with Danish Maritime Institute (located in Lyngby, Denmark) which collaboration resulted in two articles. Collaboration has also taken place with Lloyds Register, and resulted in the definition of a mathematical model for how a fire spreads on board a ship. This model is implemented as part of the Emma system, described earlier.

Mike Manthey has been been involved, over the past fifteen years, with physicists, mathematicians, and computer scientists in the USA and England who are working on a *combinatorial* reformulation of physical theory, a reformulation which has a strong computational flavor. Over the past five years, his and their work has converged strongly, and his Clifford-algebraic approach has moved into the foreground of this effort. Adding not a little to the inherent interest of this endeavor is the fact that the model which is emerging is the very same phase web model that Topsy uses.

Arne Skou. At the *internal* level, there has been intensive collaboration with the Formal Systems Group throughout the period. This includes application of automated validation tools on industrial case studies. Also there has been fruitful collaboration with the Department of Process Control in our Institute. At the *national* level, there has been intensive collaboration with the Design Group at DTU on testing frameworks. Also there has been some collaboration with the Distributed Systems Group in Department of Computer Science, Univeristy of Copenhagen, on programming environments and on coordination of research grants for modern equipment investments. At the *international* level there is a loose collaboration on Ph.D. student exchange with Gul Agha and his Open Systems Laboratory at the University of Illinois, Urbana Champaign.

4.5 Ph.D. Projects

Towards a Multiple Tuple Space Model

Name:	Keld Kondrup Jensen
Education:	Cand.polyt. (M.E.) in Computer Science, June 1989
Duration:	August, 1989–March, 1994
Status:	Degree awarded March 1994
Funding:	Aalborg University
Advisor:	Michael Manthey

This dissertation studied the synchronization constraints involved in promoting processes and concurrent objects—groups of processes and their shared state—to first-class citizens in a shared data space. That is, we proposed that processes and concurrent objects should be manipulable in a manner similar to static data objects, and through this provide for their suspension, replication, resumption,

death.

The basic problem in promoting processes to first-class objects is that a process which is to be manipulated is itself an active agent. Thus, a set of synchronization constraints is imposed on the actions of processes, and on their manipulation, in order that the concurrent execution of coordination operations be causally consistent.

Structured Operational Semantics (SOS) is used to recognize, analyze, and refine the synchronization constraints in a concurrent system. The specification itself was used as metric for the simplicity of the resulting model.

Although the dissertation was primarily concerned with the fundamental synchronization issues raised by treating processes as first-class citizens in a shared data space, it also touched on the practical issues related to implementation and use in the form of a hierarchical Multiple Tuple Space generalization of the Linda programming model.

Open Distributed Heterogeneous Computing

Name:	Christian J. Callsen
Education:	Cand.polyt. (M.E.) in Computer Science, June 1991
Duration:	August, 1991–March, 1994
Status:	Degree awarded March, 1994
Funding:	Aarhus University, collaborative agreement
Advisor:	Arne Skou

This Ph.D. project started by acknowledging the fact that future distributed systems will need to interact across architectural barriers, in the form of differences in representation of data at the hardware level. A new object-based concurrent coordination paradigm called ActorSpace was then proposed. Based on Actors, it offers a set of coordination primitives with the necessary properties, allowing transparent heterogeneity across architectural barriers. Objects compute locally and coordinate using basic asynchronous message passing. Interaction is facilitated via a built-in naming facility, available at all levels. Objects are thus able to coordinate using either direct object identifiers or using patterns that are descriptions of the potential receiver. A formal definition of ActorSpace was then given through an operational semantics, defining the exact meaning of the ActorSpace primitives. A prototype was designed and implemented, consisting of a compiler and a runtime system. The runtime system consists of components responsible for multi-tasking actors, interpreting actor definitions, carrying out ActorSpace primitives, and maintaining coherence between several runtime systems distributed across a set of nodes. Finally potential concrete extensions to ActorSpace were considered and discussed, concerning explicit specifications of the synchronization of interaction and increasing the customizability of ActorSpace [5, 10, 34].

Paradigms and Environments for the Development of Distributed Real-time Systems

Name:	Brian Nielsen
Education:	Cand.polyt. (M.E.) in Computer Science, 1993
Duration:	April, 1994–March, 1997 (expected)
Status:	In progress
Funding:	Danish Technical Research Council
Advisor:	Arne Skou

Real-time systems are used for controlling and monitoring of physical environments. They must respond to events within given absolute time constraints; failure to do so results in degraded quality, system faults, and potential damage to the environment. Current development techniques lack (1) suitable programming paradigms and *implementation environments* that support distribution and real-time; (2) techniques and tools to *validate* that an implementation satisfies its requirements; and (3) *traceability* of information flow (time constraints in particular) between analysis, design, and implementation.

My main goal is to propose and evaluate a new programming paradigm and environment for the development of distributed real-time systems. It is a central thesis that this can be realized by introducing appropriate extensions into existing paradigms, rather than inventing completely new ones altogether. Specifically, this project examines the *Actors* paradigm, a distributed concurrent object-based paradigm where each object (actor) encapsulates a state that is accessible from the outside only method invocations using asynchronous message-passing. However, real-time considerations have not previously been integrated in *Actors*. The project proposes automatic testing as a way to validate that an implementation satisfies its timing constraints. Given a formal specification of the time constraints, the goal is to automatically be able to synthesize and select "good" test cases, execute these, and evaluate their outcome.

The feasibility of the proposed techniques is to be evaluated through case-studies. That is, we prototype-implement the proposed paradigm and testing technique and use the resulting development environment to solve the cases.

Object Oriented Methods, Processes and Tools for Developing Telecommunications Management Network Systems

Name:	Birgitte Lønvig
Education:	Cand.polyt. (M.E.) in Computer Science, 1992
Duration:	September, 1995–August, 1998 (expected)
Status:	In progress
Funding:	Institute of Electronic Systems, Aalborg University
Advisor:	Charlotte Lunau (and Arne Skou)

Carried out in cooperation with L. M. Ericsson A/S, the purpose of this project is to develop an object oriented method dedicated to the development of software for managing telecommunication in networked systems. The method will be based on existing Object Oriented methodologies: James Rumbaugh's Object Modeling Technique (OMT), Grady Booch's Method, and Ivar Jacobson's Object Oriented Software Engineering (OOSE). Generally, the methodologies existing today describe the actual development process insufficiently. For example the transition from analysis to design is often neglected or only superficially treated. Issues of reuse are another important aspect of system development, to which, e.g., design patterns and frameworks contribute as solutions.

This Ph.D. project will integrate the system development process into the method, with a focus on the transition from analysis to design, and wherein design patterns will be considered as a primary element of interest.

Handling Time Requirements in Real-Time Software Development

Name:	Peter Krogsgaard Jensen
Education:	Cand.polyt. (M.E.) in Computer Science, 1992
Duration:	September, 1995–August, 1998 (expected)
Status:	In progress
Funding:	Aalborg University
Advisor:	Michael Manthey and Arne Skou

This project is a part of the program for Environments and Paradigms for Development of Reliable and Distributed Real-Time Software. My personal entry to the subject is that of a practitioner, and after having participated in development and maintenance of applications which communicated with and controlled more or less intelligent hardware, my opinion is that a lack of tools and methods to support the time aspect in software development makes creation of reliable real-time applications overly difficult. I am searching for harmony in the development process, which will make the developer certain that his time-requirements are treated correctly from design through implementation to execution.

At the design level, a lack of expressive power exists in the usual development methods. Suppose an interaction with a radar system requires an position message every 500 milliseconds: how can this be formalised? I propose creating a general design method, or an extension to an existing method to make it possible to include timing constraints in the software development from the very first day.

The configuration level is concerned with creating a guaranteed plan for execution. The development process could analyze a set of applications, with described real-time abilities, and an execution system, with described characteristics. Suppose that the execution system is an OS with prioritized round-robin scheduling, and the application is the radar system from before. How can an execution gua-

rantee that the position message is generated every 500 milliseconds? The result could be a global scheduling-plan for an execution system or it could be that only a subset of the applications can run simultaneously if their demands are to be guaranteed.

The aim of the project is to verify experimentally the developed methods and tools with cases from industry.

4.6 Service and Research-related Activities

Arne Skou has, in this period, served as reviewer for a number of international journals and conferences. He has been the co-organizer of two international conferences. He has been head of department for the period 1993–1995.

Michael Manthey has, in this period, served as consulting editor for the international journal Cybernetics and Human Knowing, refereeing and editing articles, and writing book reviews. He is also a member of the executive board of the Alternative Natural Philosophy Association, based in Cambridge, England.

4.7 The Group's Own Evaluation

We cover the positive and negative aspects separately, starting with the positive aspects.

In spite of persistent overwork, we feel that we have succeeded in developing and promoting our research at an entirely acceptable level. Furthermore, this research is in step with, and in places anticipates, the state of the art. We are satisfied (indeed, proud) of our student "product" and the level of student engagement in research-level topics, and the working milieu in our group is good.

During the period we have continued the very positive collaboration with the Formal Systems Group. Furthermore, we have succeeded in establishing collaborations with other groups at the national level. The diversity of interests within the group's permanent staff has both advantages and drawbacks. On the positive side is the fact that the group covers a main part of the classical curriculum within distributed systems for computer system engineers.

On the negative side, we planned at the beginning of the period to work on semantically well-founded programming environments for distributed systems in general, in collaboration with the Formal Systems Group. Unfortunately, when an assistant professor left the group in the middle of the period, the resources for this collaboration became unavailable.

The diversity of our staff's interests leads to a lack of common focus, making it harder to establish a visible profile. In general, the lack of numbers (both staff, with the breadth this would give, and Ph.D. students) has hindered us both in seeking funding for and participating in more ambitious projects, local

and otherwise, and in imparting to our software the necessary "final finish" that would allow us to distribute it.

4.8 Research Plan 1996–2000

Our overall goals for the coming five years are threefold.

- We will attempt to increase the overlap between the research interests and student projects so as to both maximize the results of our individual efforts and clarify our external "profile."
- Another goal is to increase the amount of external funding, allowing us to support additional Ph.D. students, increase our collaboration with both industrial and academic entities, and improve and intensify our local milieu.
- Third, we would like one additional member to our staff because we are in need of both additional teaching resources and topical breadth.

In addition to these broad group-level goals, we sketch in the following our plans relative to the previously described research areas.

Programming Distributed Systems

We expect that our research in this area will be pursued primarily via Charlotte Lunau's investigations of reflective programming, and the Ph.D. pursuits of Brian Nielsen and Peter Krogsgaard Jensen, building on the tools and experience from the previous period.

Environments and Paradigms for Real-Time Systems

As mentioned previously, two Ph.D. projects are currently investigating different aspects of real-time systems design and implementation. It is our plan to continue this work. In particular, we expect during the period to establish a programming environment to support the development of such systems all the way from an initial specification to the final implementation. Part of this work will be in collaboration with the Formal Systems Group.

Charlotte Lunau is planning to develop reflective architectures for advanced and reliable real-time applications. The reflective architecture developed in Emma needs to be extended to cover automatic recovery. A set of design rules for structuring reliable real-time applications will be developed. This work will be part of the RRELCO project, carried out in cooperation with the Department for Process Control, AAU and DTU.

Protocols for High Speed Data Communications

We have established collaborations with other groups at the national level on operating system support for modern information technologies such as high speed nets. It is our plan to continue this positive development. In particular, we intend to continue our work on high speed protocols, and we plan to apply the results to contemporary applications.

Self-Organizing Distributed Systems

Our research plans for the next five years are (1) to perform as many experiments with Topsy as time and personnel allow, while (2) improving and distributing our software and (3) extending our mathematical understanding.

At the current stage of our understanding of how to actually *use* Topsy's abilities, we seek problems/experiments which allow us to focus on basic issues one at a time.

For example, Topsy cannot currently solve the very simple problem of interchanging two blocks' positions on a table, but for a very sophisticated reason: because sensorially the two blocks are (assumed) indistinguishable, interchanging them requires that Topsy acquire the concept of an object's being independent of its immediate sensory signature.

On the other hand, we are also eager to investigate the cognitive effect of coupling vision to behavior, a decidedly non-simple problem. Our current experiment in this area is to use three planes of sensory input—table "places" and two orthogonally positioned "eyes"—to investigate the emergence of hand-eye coordination by setting Topsy to stacking blocks.

Among future experiments are more focused attempts to design low-level pattern-sensitive "eyes," where we will take our inspiration from experimental data from the ape visual system, which elucidate simple and explicit neural structures. The Laboratory for Image Analysis at AAU has been consistently sympathetic toward our efforts over the years, and we expect our informal collaboration with them to continue.

Another experiment is simply to investigate Topsy's performance in solving the Pentomino problem: tiling a 12×5 area with the twelve possible planar configurations of five face-attached cubes. This is a computationally "hard" problem, but (bearing in mind recent results showing that a quantum mechanical machine can do such problems in polynomial time) we will not be surprised if Topsy finds it easier!

Toward the end of the period, we hope to attract problems from other areas. One genuine possibility is the extension of our informal collaboration with AAU's Institute of Production, dealing with assembly line control.

There is also a considerable amount of work to be done in (1) challenging, evaluating and polishing the learning mechanisms; (2) elucidating the abstrac-

tions and knowledge concepts possible and used; (3) merging separately acquired knowledge structures (in principle no problem, but there can also be conflicts at higher levels).

The above can only succeed if we are able to solidify our software base. Hitherto we have relied on the software generated by student projects, but this is no longer possible for two reasons: (a) despite students' best efforts, such software simply cannot attain the quality-level necessary for more sophisticated experiments, nor to distribute elsewhere and therewith attain international collaboration, and (b) there is an overwhelming need to develop sophisticated and high-bandwidth graphical "windows" between Topsy and its user, but their development contains too little academic content and too much code to be defensible as student projects. We are therefore seeking funding for improving Topsy's graphical user interface, modifying it to run on various parallel platforms, and creating and linking material for Internet availability.

We plan of course to continue our mathematical investigations, but this will be, as it has been, a secondary priority. However, we hope via a greater visibility of our work to attract collaborators. There is much—interesting—work to be done here.

Bibliography

Editorships and Book Chapters

[1] U. H. Engberg, K. G. Larsen, and A. Skou, editors. *Proceedings of the Workshop on Tools and Algorithms for the Construction and Analysis of Systems*, number NS-95-2 in BRICS Notes Series, May 1995.

[2] K. G. Larsen and A. Skou, editors. *Proceedings of the Third Workshop on Computer Aided Verification*, Aalborg, Denmark, July 1991, Volume 575 of *Lecture Notes in Computer Science*. Springer-Verlag, 1992.

[3] K. G. Larsen and A. Skou. "Bisimulation through Probabilistic Testing." In J. Stage, K. Nørmark, and K. G. Larsen, editors, *Quality Software - Concepts and Tools*. Software Engineering Program, Aalborg University. ISBN 87-985170-0-7.

Patents and Refereed Journal Articles

[4] A. Børjesson, K. G. Larsen, and A. Skou. "Generality in Design and Compositional Verification Using TAV." *Formal Methods in Systems Design*, Vol. 6, pp. 239–258, 1995.

[5] C. J. Callsen and G. Agha. "Open Heterogeneous Computing in Actor-Space." *Journal of Parallel and Distributed Computing*, special issue on heterogeneous computing, Vol. 21, pp. 289–300, 1994.

[6] K. G. Larsen and A. Skou. "Bisimulation through probabilistic testing." *Information and Computation*, Vol. 94, No. 1, 1991.

[7] C. P. Lunau and K. G. Larsen. "On the Design of Encapsulated CLOS Applications." *Journal of Object Oriented Programming*, Vol. 7, No. 7, pp. 34–38, November–December, 1994.

[8] M. Manthey. "Synchronization: The Mechanism of Conservation Laws." *Physics Essays*, Vol. 5, No. 2, June 1992.

[9] M. Manthey. "Artificial Intelligence System". U.S. Patent # 5,367,449, 1994.

Refereed Conference Articles

[10] G. Agha and C. J. Callsen. "ActorSpaces: A Model for Scalable Heterogeneous Computing" (extended abstract). In *Proceedings of the 26th Hawaii International Conference on System Sciences*, Vol. 2, pp. 100–101, January 1993.

[11] G. Agha and C. J. Callsen. "ActorSpaces: An Open Distributed Programming Paradigm." In *Proceedings of the 4th ACM Symposium on Principles & Practices of Parallel Programming*, pp. 32–32, May 1993.

[12] S. Andersen and C. P. Lunau. "Voyage Optimization, The DMI Expert Voyage Pilot." In *Proceedings of IFAC Workshop on Artificial Intelligence, Control, and Advanced Technology in Marine Automation*, pp. 87–97, Geneva, April 1992.

[13] M. Blanke, C. P. Lunau, and C. Hornsby. "KBSSHIP - Communicating Expert Systems for Ship - Wide Decision Support." In *Proceedings of IFAC Workshop on Artificial Intelligence, Control, and Advanced Technology in Marine Automation*, pp. 73–86, Geneva, April 1992.

[14] A. Børjesson, K. G. Larsen, and A. Skou. "Generality in Design and Compositional Verification Using TAV." In Michel Diaz and Roland Groz, editors, *Formal Description Techniques, V - FORTE '92*, pp. 449–464, North-Holland, October 1992.

[15] J. C. Godskesen, K. G. Larsen, and A. Skou. "Automatic Verification of Real-Time Systems Using Epsilon." In S. T. Voung and S. T. Chanson, editors, *Fourteenth International IFIP Symposium on Protocol Specification, Testing and Verification - PSTV '94*, pp. 232–330, Chapman and Hall, June 1994.

[16] J. C. Godskesen, K. G. Larsen, and A. Skou. "Verification of Real-Time Applications Using The Epsilon System." In *Nordic Seminar on Dependable Computing Systems - NSDCS '94*, pp. 41–52, Technical University of Denmark, August 1994.

[17] C. H. Kristensen, J. H. Andersen, and A. Skou. "Specification and automated verification of real-time behaviour – a case study." In *3rd IFAC/IFIP workshop on Algorithms and Architectures for Real-Time Control (AARTC '95)*, pp. 613–628, Belgian Federation of Automatic Control, June 1995.

[18] K. G. Larsen and A. Skou. "Compositional Verification of Probabilistic Processes." In R. Cleaveland, editor, *Proceedings of CONCUR '92*, Volume 630 of *Lecture Notes in Computer Science*, pp. 456–471, Springer-Verlag, 1992.

[19] K. G. Larsen and A. Skou. "Testing and Verification of Probabilistic Processes." In P. Heegaard and B. Helvik, editors, *Nordic Seminar on De-*

pendable Computing Systems, NSDCS '92, pp. 235–250, NTH, Trondheim, 1992.

[20] C. P. Lunau. "Encapsulation between Classes in CLOS." In *Proceedings of the Workshop on Object-Oriented Programming in Lisp: Languages and Applications Kaiserslautern*, Germany, July 1993.

[21] C. P. Lunau and J. K. Nielsen. "Emma: An Emergency Management System for use onboard Ships." In *Proceedings of IFAC Workshop on Control Applications in Marine Systems*, Trondheim, May 1995.

[22] M. Manthey. "The Combinatorial Hierarchy Recapitulated." In *Proceedings of the 1992 International ANPA Conference*, Cambridge, England, 1992.

[23] M. Manthey. "A Vector Semantics for Actions." In *Proceedings of the 1993 International ANPA Conference*, Cambridge, England.

[24] M. Manthey et al. "A Simple Vision Experiment Using Topsy." In *Proceedings of the 1994 International ANPA Conference*, Cambridge, England.

[25] M. Manthey, "Toward an Information Mechanics." In *Proceedings of the 2^{nd} IEEE Workshop on Physics and Computation*, pp. 95-109, Dallas, TX, November 1994.

[26] J. K. Nielsen and C. P. Lunau. "Emma: Emergency Management of Marine Accidents." In *The official Conference of Europort 1995*, Lloyds European Shipping Forum, Amsterdam, November 1995.

Technical Reports

[27] G. Agha and C. J. Callsen. "ActorSpaces: A Model for Scalable Hetergeneous Computing." TR UIUCDCS R-92-1766 and UILU-ENG-92-1746, Department of Computer Science, University of Illinois, Urbana-Champaign, November 1992.

[28] T. J. Hansen, B. Nielsen, J. F. D. Nielsen, and A. Skou. "Environments and Paradigms for Development of Reliable Distributed Real-Time Software: An Industrial Oriented Research Project." Technical report, Aalborg University, 1994.

[29] K. Henriksen, R. J. Hillemann, W. Pietraszek, A. Skou, and M. Aaen. "Experiments with TCP/IP in ATM High Speed Data Communications." Technical report, Aalborg University, 1995.

[30] H. D. Jensen and T. Rix. "AUC-Linda User's Manual." Technical report IR-95-2007, Aalborg University.

[31] C. P. Lunau, K. H. Christensen, and J. Sommer. "Design Specification for the Emergency Management System EMMA." Technical Note 2304.03.03.-064.001 ATOMOS Project.

[32] R. Sharp and A. Skou. "A Testing Framework." May 1992. (unpublished)

Ph.D. Theses

[33] K. K. Jensen. "Towards a Multiple Tuple Space Model." 1994.

[34] C. J. Callsen "Open Distributed Heterogeneous Computing." 1994.

Chapter 5

Formal Systems

5.1 Profile

The research activities of the Formal Systems Group are concerned with the development of formal semantic theories, supporting methodologies and automatic tools for specifying, designing, and formally verifying the correctness of so-called reactive systems, i.e., systems that evolve by reacting to stimuli from their environment. Traditionally, reactive systems aim at modeling the behavior of distributed and concurrent systems, but have recently been extended to also include real-time and hybrid systems (mixtures of discrete and continuous components and behavior).

The research approach of the Formal Systems Group has to a large extent been carried out as an integrated and iterative development of

- sound theoretically and mathematically based formalisms,
- design and implementation of tool support for semantic analysis, and
- application and validation of the formalism and tools based on practical case-studies.

We believe that each of these three directions, as well as their integration, is necessary.

The research of the Formal Systems Group has its roots in Process Algebra. In particular, the original work by Milner on CCS and by Hoare on CSP as well as later work by Bergstra (ACP), Boudol (MEIJE), Brinksma (LOTOS), Hennessy, and others is an important common scientific heritage for the Formal Systems Group. Common to all these algebras is that they understand the behavior of a system in terms of its interaction with a surrounding environment. Important questions that arise for each individual Process Algebra are the following:

- How can systems be combined and how is the semantics (i.e., behavior) of a combined system determined by the semantics of its components? In

particular, a choice of semantic presentation (operational, denotational or axiomatic) must be made.

- When should two systems be considered semantically equivalent? A number of abstracting behavioral equivalences has been proposed to answer this very question for a variety of abstraction levels. In particular, equivalences abstracting from non-determinism and/or internal computations of systems has been put forward.

- What are the relationships between ways of combining systems and the chosen behavioral equivalence? In particular, (complete) algebraic laws are sought.

In the 1990 research evaluation, a research plan was formulated to "continue research in all three current directions, i.e., development of specification formalisms, design and implementation of tools, and application in case-studies" and "to extend the process algebraic approach with non-classical concepts." In particular, we expressed a need for explicit integration of quantitative aspects such as (dense) real-time aspects and probabilistic non-determinism in order to enable appropriate modeling of real-time and dependable systems. Also, extensions of the classical Process Algebras from mere synchronization between process to actual communication of values and to systems with infinite state-spaces were planned.

In addition to these extensions, we have constantly aimed at obtaining a unified view of the various Process Algebras (and their extensions) though the establishment of algebra-independent results.

5.2 Activities and Results

The research of the Formal Systems Group in the period 1990–1995 has thus been driven by a desire to extend *all* classical process algebraic results in a number of directions, in particular to include real-time and probabilistic aspects, as well as value-passing and infinites state-space systems. Simultaneously, we have aimed at designing and implementing supporting software (verification) tools and carry out case-studies.

The above research plans have been successfully carried out in collaboration with other groups in the department (in particular the Computer Systems Group), with the Theoretical Computer Science Group at University of Aarhus (currently) within the BRICS center and (before) within the DART Programme. At a European level, our research plans have been carried out within (and are almost identical to) the successfully completed Esprit Basic Research Action CONCUR2. More recently, a new and very promising line of international collaboration with the Department of Computer Systems, Uppsala University, Sweden has been initiated.

In the remainder of this section, we describe first the most important of group's research results organized by research areas and then describe the projects that the group has taken part in.

5.2.1 Research Results

Process Algebra

The Formal Systems Group has contributed to the continued development of classical process algebraic results. On the theoretical side, the group has studied (variations on) standard process description languages and behavioral theories for these. (Cf., e.g., the references [69, 8, 12, 49, 17, 13, 5].) Complete axiomatizations of notions of behavioral equivalence for interesting classes of process specifications have also been obtained (e.g., [17, 69, 49, 13, 72, 73, 74]). These form the basis for (semi-)automatic tools based on term-rewriting and theorem-proving. The study of mathematical models for concurrent agents has also been pursued [31, 74].

Logics and Specification Formalisms

A specification formalism allows one to express in a precise formal manner the desired properties of an implementation to be subsequently developed. Specifications may either themselves be process algebraic (abstract) descriptions or they may be logical formulas of some (temporal or modal) logic. The choice of formalism depends on a number of (often conflicting) factors. On the one hand, we want the formalism to be as *expressive* as possible (in order to allow as accurate and concise specifications as possible). On the other hand, we want (ideally) correctness checking (model checking) to be (semi-)decidable in order make the construction of verification tools possible. In our research we have proposed, analysed, and compared a number of specification formalisms suited for classical process algebraic descriptions.

- *Graphical Specifications and Modal Transition Systems* [24, 61, 21]
  A number of *graphical* specification formalisms have been proposed. These formalisms may be seen as *logical* extensions of process algebra, and are thus somewhat more expressive. However, they still allow for efficient (i.e., polynomial) model checking and compositional verification since specifications may be combined with respect to the process algebraic operators. In particular, the theory of Modal Transition Systems has formed the basis of the verification tool TAV.

- *Compositionality* [44, 25]
  Ideally, we want a specification formalism that supports compositional verification, i.e., it should be possible to infer a correctness property of a

combined system from similar correctness properties of components of the systems. Formal results constructively providing this correspondence have been given.

- *Constraint Oriented Methodology* [30, 35]
  A constraint-oriented state-based proof methodology has been put forward and applied on a number of examples. The methodology is based on Modal Transition Systems and exploits compositionality and abstraction for the reduction of the verification problem under investigation. The methodology employs extremely fine-granular specifications: each aspect of a component is specified by a (large) number of independent (but similar) constraints.

- *Implicit Specifications and Equation Solving* [65, 18, 51, 68]
  It is sometimes more convenient to specify the behavior of a desired system implicitly rather than directly. That is, we specify the behavior of the system in a particular *context* rather than the system itself. A remarkable fact is that the expressive power of *implicit* process algebraic specifications is precisely that of temporal logic.[1] The synthesis problem for implicit process algebraic specifications has been shown to be PSPACE-hard [68] in the unary case (i.e., one component is specified in a certain context) whereas the synthesis problem for polyadic implicit specifications (i.e., several components are simultaneously specified in a certain context) is only semi-decidable [51]. In both cases, prototypes tools have been developed.

Real Time and Hybrid Systems

For numerous practical systems, the most important and critical aspect is that the services offered by the system are provided at the *right* moments in time (e.g., traffic control systems, plant process controllers, interactive audio- and video (remote) control, aircraft, and robot design). By their very nature, such real-time systems are not adequately described as classical finite-state systems, but rather requires additional information about timing-constraints. Hybrid systems are dynamical systems consisting of interacting discrete and continuous components. They are used to model the combined behavior of embedded real-time systems and their physical environments.

These two related research areas are becoming very active in Computer Science, due to both the increasing importance of embedded and real-time systems and the emergence of of promising techniques and tools for specification and verification.

During the entire period, the Formal Systems Group has been actively engaged in the theoretical development of a timed extension of Process Algebra suitable for

[1] More precisely, that of the modal ν-calculus.

the description of real-time and hybrid systems. Also, from the very beginning, we have been anxious to implement our theories in automatic verification tools.

The following summarizes our contributions in the period:

- *Timed Process Algebra and Logic*
  A number of behavioral equivalences between real-time systems have been introduced and their relationships completely determined [48]. In the search for complete axiomatization of these equivalences it has been proven that the technique traditionally used for classical Process Algebras (the so-called Expansion Law) is not applicable [57, 59]. The problem is essentially that even the simplest real-time system has infinite-state behavior due to the denseness of the timing-domain. However, development of new techniques has enabled complete axiomatizations to be obtained. Compositionality issues for real-time systems have also been completely determined: It has been shown how a real-time logical property of a combined real-time system may be decomposed into a necessary and sufficient property of the component of the system [34, 37]. Also, the synthesis problem (i.e., given a logical specification, construct an implementing real-time system) was shown to be decidable [34, 36]. Finally, it has been shown how various behavioral equivalences between real-time systems can be (effectively) transformed into (decidable) logical model-checking problems [36].

- *Symbolic Verification Techniques and Decidability* [67, 50, 38, 39]
  Due to their immediate infinite state nature classical model-checking techniques developed for finite-state systems are not directly applicable. However, during the last five years, the theoretical foundation for automatic verification has been provided by the development of a symbolic technique known as the region-graph technique (due to Alur, Courcourbetis and Dill). This technique allows the analysis of a timed system to be carried out based on an abstract and *finite* interpretation [67].

- *EPSILON*
  Only after three years, Karlis Cerans showed how the region-graph technique could be extended to deciding various time-sensitive equivalences between real-time systems. In reference [50], this technique is presented, and a (decidable) timed extension of the specification formalism Modal Transition Systems is given together with a number of smaller examples. This work together with [33] provides the theoretical basis for the automatic verification tool EPSILON (logically, a timed extension of the TAV system). A number of case-studies have been (automatically) verified using EPSILON [43, 45, 40].

- *UPPAAL*
  However, algorithms directly based on the region-graph technique are all

faced with a potential explosion in the state-space of the abstract interpretation occurring when considering networks of systems. During a half year visit spring 1995, a very fruitful collaboration with Department of Computing Systems at Uppsala University was initiated on the design and implementation of new and more efficient symbolic techniques for real-time systems. In particular, we have developed techniques allowing the verification to be reduced to that of solving simple linear constraint systems, as well as compositional techniques intended to avoid explicit construction of the complete (and exploding) state-space of the (abstracted) network [38, 37, 39]. These techniques have been implemented in C++ in the new tool UPPAAL and already at this stage experimentally demonstrated to perform superior to all other existing verification tools for real-time systems [28]. A distinguishing feature for UPPAAL is its graphical user interface (insuring that what you see is really what you verify) and the automatic generation of diagnostic debugging information [29] produced in case or erroneous systems. The case-studies considered so far include a Steam Generator provided by the Department of Process Control, Aalborg University, and the (actually installed) Interface Bus Protocol used in a Philips Audio System (with a verification time of 3.6 seconds). For more information on UPPAAL we refer to the World Wide Web address `http://www.docs.uu.se/docs/rtmv/uppaal`.

In conclusion, we feel that we have successfully accomplished our goal in extending all classical process calculi results and methods to the real-time setting.

Tools

As already indicated, the Formal Systems Group has been continuously involved in the development of tools for automatic verification of concurrent system development since 1984, in parallel and competition with the tools of other research groups.

The first tool in the series is TAV, which like to many other tools, provides algorithms for deciding various behavioral equivalences between processes, and for model checking (with respect to a recursive modal logic). However, as a unique feature of TAV, both the equivalence and model checking tools are based on a (polynomial time) *local* checking algorithm [55] exploiting only as necessary the state space of the process under consideration. Beside avoiding the state-explosion problem, this unique feature also enables the construction of concise explanations for the answers produced (e.g., a small distinguishing formula for non-equivalent processes). The TAV tool extends other (competing) tools by the use of modal transition systems, thus enabling partial specifications to be expressed; a feature of crucial importance for step-wise development of systems. During the period 1991-95, the work on the TAV tool was completed and demonstrated on a number of case studies [61, 35].

Later, recent decidability results on timed modal transitions and on timed logics have been implemented in a similar tool for real-time systems (EPSILON) [50]. EPSILON has also been successfully demonstrated on a number of case studies [43, 45, 40].

Most recently, an extremely efficient model checker for real-time and (certain) hybrid systems (UPPAAL) [28] was designed and implemented in intense collaboration with a research group (headed by Lecturer Wang Yi) at Department of Computer Systems at Uppsala University. This tool is based on novel symbolic techniques and significantly outperforms all other existing verification tools for real-time and hybrid systems. These results have created general international interest and we have already received a number of tutorial invitations (to the DIMACS Hybrid Systems Workshop 1995, IEEE Real Time Systems Symposium 1995, Foundations of Computation Theory 1995, and the International School and Symposium on Formal Techniques in Real-Time and Fault Tolerant Systems, 1996).

Probabilistic Processes

With respect to probabilistic behavior, we have studied various specification formalisms. In particular, a transition-based specification formalism for probabilistic behavior was introduced, together with a characterization of the refinement ordering between specifications [66]. The resulting theory can be seen as an extension of the theory for modal transition systems. Furthermore, a probabilistic calculus and a probabilistic modal logic was introduced [56]. As main results, it is shown how to decompose properties expressed in the logic with respect to operators of the calculus. Also a complete axiomatization of (validity of) the logic is given. Finally, various test theories for probabilistic processes were developed. A test theory that introduces a probabilistic version of bisimulation was developed [23, 53] and is by now widely recognized and accepted. It has been shown that a process will pass a test with a certain probability [60], refining the notions of a process *must* or *may* pass a test in the classical test theory of Hennessy.

Value Passing

We have extended existing semantic models for communicating processes with value-passing. We give two kinds of models: The first one are models based on the idea of testing [15, 16, 5], where processes are only to be distinguished if they can be distinguished by another process. The second is based on the more standard idea of bisimulation [74, 75]. In both cases, we give an operational, denotational, and axiomatic semantics, and we show that at least under some circumstances, they all are equivalent. We also investigate the difference between the semantics based on the late and the early approach [74, 75, 5].

A denotational model is fully abstract with respect to a behavioral model for

processes if two processes have the same denotational interpretation if and only if they are behaviorally equivalent. We have given such fully abstract models for several behaviorally based semantics [15, 16, 31, 32].

Infinite State Transition Graphs

In recent years, an important theme of research has been that of the decidability of verification problems for processes with infinite transition graphs and the new angles that the study of such problems gives us on familiar problems within the realm of formal language theory. A central class of problems here is that of *equivalence problems*—when do two processes exhibit the same behavior? For context-free grammars this problem is undecidable if one takes "same behavior" to mean language equivalence. However, if one considers the bisimulation equivalence of Park and Milner and considers the transition graph with vertices corresponding to sentential forms and edges given by leftmost derivation steps, the equivalence problem for context-free grammars is decidable. This result was established by Baeten, Bergstra, and Klop—a simpler proof was later found by Stirling and Hüttel [70]. Later, the result was generalized to all context-free transitions graphs, a result due to Stirling, Christensen, and Hüttel [70].

It turns out (as shown by Hüttel and Groote [14]) that no other known behavioral equivalences have a decidable equivalence problem for this class of transition graphs. The class of context-free grammars corresponds exactly to the process calculus BPA (first studied by Bergstra and Klop) which has sequential composition, nondeterministic choice and recursion.

The same problems can be studied for a process calculus with a notion of parallel composition instead of sequencing. Again, it turns out that bisimulation equivalence is undecidable (a result proved by Christensen, Hirshfeld, and Moller) while no other non-trivial equivalence is decidable for this class of processes, a result due to Hüttel [47].

Mobile Processes and Concurrent Functional Languages

Within the past few years, members of the group have developed an interest in functional languages, their relationship to process calculi with a notion of mobility, and the calculi themselves. In particular, there has been interest in the group in the notion of polymorphic type systems in the context of Concurrent ML-like languages with asynchronous (Linda-like) concurrency. Hüttel has, together with three of his students, provided such a language with an effect type system and a type reconstruction algorithm [41]. The effects are terms of an asynchronous higher-order process calculus.

There has also been work on mobile process calculi in the tradition of the π-calculus. Two M.S. students have studied the question of equivalences for both process-passing and label-passing calculi with asynchronous process calculi, and

they have obtained the surprising result that the three variations of bisimulation equivalence studied in the π-calculus coincide for an asynchronous calculus [42].

Meta Theory

The meta theory of process algebras aims at contributing to the systematic development of process theory by offering results that hold for classes of process description languages. As these languages are often equipped with a Plotkin-style Structural Operational Semantics (SOS), this way of giving semantics to processes provides a natural handle to establish results that hold for all languages whose semantics is given by means of inference rules that fit a certain format. Examples of the kind of meta-theoretic results that have been systematically derived from the form of the SOS rules may be found in the following contributions by members of the Formal Systems Group [10, 7, 46, 32]. So far, this line of research has produced a wealth of results which generalize and explain several of the most important theorems and constructions in process theory. For example, given a language with an SOS semantics, an examination of the SOS rules is often all that is needed to guarantee that a notion of behavioral equivalence or preorder will be preserved by the constructs in the language, that a process equivalence can be equationally characterized [10, 46], that a language generates only finite-state processes [7], or that a language has fully abstract denotational models [32].

5.2.2 Projects

BRICS, January 1994–January 1998

BRICS, a center for Basic Research in Computer Science, is a joint effort between the Formal Systems Group and the Theoretical Computer Science Group at the University of Aarhus. The Center is funded by the Danish National Research Foundation[2] for the period 1994–1998 (official start January 1, 1994). The Danish National Research Foundation was established in 1991 in order to enhance Denmark's research development capability. This objective was reached by funding "unique" Danish research at "the international level." The foundation presently funds 23 research centers, among them BRICS.

The aim of BRICS is to establish in Denmark important areas of basic research in the mathematical foundations of Computer Science, notably Algorithmics and Mathematical Logic. These are areas of great international significance. The objective is to attract mainly foreign research experts to contribute to the research efforts of the center, and to the training of young Danish researchers. The research plan is based on a commitment to develop Algorithmics and Logic integrated with existing strong activities in the Semantics of Computation, using a combination

[2]Additional information on the Danish National Research Foundation is available from `http://www.brics.aau.dk/BRICS/DG/index.html`.

of long-term efforts and a number of short-term, intensive programs with carefully chosen scientific themes. Organisationally, BRICS is an autonomous center with its own management, and yet with its activities strongly integrated in the existing infrastructure and student environments at the two universities.

During its first one and a half years, much effort has been invested in establishing BRICS as a visible center of expertise, locally as well as internationally. The staffing of BRICS consists today of the 8 university-employed kernel researchers (Kim Guldstrand Larsen and Arne Skou from Aalborg University) originally behind the application to the Research Foundation, 10 BRICS-employed researchers at the postdoctoral level, a further 17 associated researcher financed by alternative funding, and a total of 14 Ph.D. students, of which 5 are supported by grants from BRICS itself. So, BRICS is today a well-defined institution, and yet with all its activities integrated in the computer science departments at the two universities.

Scientifically, BRICS has spawned a tremendous number of activities already in its first year. This is probably best reflected by a list of 137 publications, almost all of which are published or accepted for publication in international journals or conference proceedings. Scientific publication is also supported by our BRICS Research Report and Notes Series.

During the first year a first research theme in Complexity Theory was successfully completed and a second theme, Logic in Semantics, initiated. For 1996 we plan a theme on Verification (to be organized as a summer-school).

One of the most visible effects of the existence of BRICS is the sizable group of talented Ph.D. students now associated with the center. After its first year, BRICS has managed to attract and finance a total of 17 Ph.D. students (three of whom got their degrees in 1994) all supervised by BRICS kernel researchers (from one to three students each). Not only do these students contribute scientifically to the aims of BRICS, but the quantity of students has resulted in an intensive program of associated courses and activities having been generated.

For more information on BRICS, refer to the World Wide Web address: `http://www.brics.aau.dk/BRICS/`.

CONCUR2, September 1992–September 1995

During the entire period, the Formal Systems Group has been engaged in the Esprit Basic Research Action no. 7166, CONCUR2. The aims of this project are almost identical to those of the Formal Systems Group in the period.

Concurrency theory is important for the specification and verification of concurrent and distributed systems. CONCUR2 aimed at extending process algebra and logical calculi to incorporate real-time aspects, probabilistic nondeterminism, value passing and infinite state spaces. CONCUR2 sought a unified view of process algebra (in particular, unifying the various semantical styles), and succeeded in designing, specifying, and implementing supporting software tools, and

common formats and interfaces for same.

The project delivered both a significant extension of theory and notation for process algebras and related calculi, exemplified by a substantial range of case studies as well as some de facto software standards concerning tool design and tool interfacing. To this end, we pursued three lines of research:

- Theoretical research, on the extension and unification of process algebras and calculi.
- Tool development, the design, specification, and implementation of prototype software tools, and common formats and interfaces for such tools.
- The application of our theory and tools in relevant case studies.

The CONCUR2 project brought together 8 partners and 4 associated partners, combining the largest academic teams working on process algebra as well as partners having long-standing experience in tool design for process algebra. The partners was: Aalborg University (Denmark); CWI (The Netherlands); University of Edinburgh (UK); Eindhoven University of Technology (The Netherlands); INRIA Sophia Antipolis (France); Oxford University (UK); University of Sussex (UK); Swedish Institute of Computer Science (Sweden). The associated partners ware: IMAG, Grenoble (France); European Computer-industry Research Centre (Germany); Sharp Laboratory of Europe, Ltd. (UK); Chalmers University of Technology (Sweden).

Kim Guldstrand Larsen was coordinator and editor for the original proposal and scientific manager during the first year. For more information on CONCUR2, we refer to the World Wide Web address: `http://www.cwi.nl/~fritsv/concur2/concur2.html`.

DART, 1991–1993

Before the start of BRICS, the Formal Systems Group participated in the national DART Programme (Design, Analysis, and Reasoning about Tools) during the period 1991–1993 together with the Theoretical Computer Science Groups in Copenhagen and Aarhus. The project was and is still funded by the Danish Research Programme on Informatics, PIFT. In Aalborg, the main activity has been on the semantics of concurrency, with special emphasis on the development of a theoretical basis together with supporting methodologies and automatic tools for designing provably correct distributed/concurrent systems. In particular, theories allowing modular design and compositional verification are sought; i.e., it should be possible to relate properties of a complex system to properties of its components.

Using the modal μ-calculus as a specification language for parallel systems, we have described methods for decomposing specifications of a combined process

into sufficient and necessary properties for its components. This theoretical basis for compositional verification has been established for classical process algebras such as CCS, CSP, and ACP. Furthermore, the expressive power required of a specification formalism in order that it support decomposition of properties has been characterized.

Related to these decomposition methods, we devloped a number of *local techniques* for checking various properties of parallel systems. In contrast to traditional *global* techniques, the local techniques are designed so that pre-computation of the global state-space (and hence state-explosion) may be avoided. A general description of the local technique underlying all the tools of the TAV system has been given in terms of Boolean Equation Systems. Local algorithms for model checking with respect to the modal μ-calculus have been given. In particular, the efficiency of these local techniques are comparable to the best known global algorithms, and we are currently working towards their practical implementation.

Using classical process algebras and classical modal and temporal logics, it is possible to specify concisely the desired observable behavior of a parallel system. However, such specifications will focus only on *qualitative* aspects of a system while leaving unspecified *quantitative* aspects which often are vital for several practical applications. Such quantitative aspects include: real-time (dense or discrete), probabilistic non-determinism, passing of values in communication, and priorities among processes and/or actions.

Within DART substantial research has been carried out in order to extend classical results in the above directions. More detailed accounts of the results obtained will be given as part of the activity descriptions.

Software Engineering, 1990–1992

From 1990 to 1992, this research project involved the Programming Systems, Information Systems, and Formal Systems Groups in the unit. I was supported by the Danish Natural Research Council, Grant no. 11–8394; see the related section under the Information Systems Group.

POTP, 1988–1991

The research project POTP (Programming Environments in Theory and Practice) was a joint project between the Programming Systems and Formal Systems Groups. The project was supported by an Aalborg University "Grundforskningsbevilling," project no. 714.

EXPRESS, 1993–1996

EXPRESS is a network under the European Human Capital and Mobility initiative. The Formal Systems Group participates in this network with Luca Aceto as representative.

One of the main research objectives of computer science is the development of formal methods for the design and implementation of programming languages. A most prominent feature of this research area is the proliferation of programming concepts which have been developed and studied: We mention programming paradigms such as imperative programming, logic programming, functional programming, concurrent programming, object-oriented programming, and data flow, and the various combinations thereof. The formalization of the relation between a specification and its implementation has been a main focus of systematic research. However the issue of the relative expressive power of the various programming concepts, an issue which is directly related to the implementation and use of programming languages, has hardly been addressed in a systematic manner.

Such methods will provide a tool to classify the variety of programming languages and will provide a formal basis for the design principles and implementation of programming languages. The research program of EXPRESS aims at providing a general framework for the comparison of the formal methods for specification and verification developed within the various programming paradigms.

The EXPRESS proposal brings together 15 prominent teams in the area of expressiveness. Besides the coordinator, the consortium consists of 6 proposers, 4 associated proposers, and 4 external proposers from non-Member States: Centre for Mathematics and Computer Science (CWI), The Netherlands (coordinator); Eindhoven University of Technology (TUE), The Netherlands; Swedish Institute of Computer Science (SICS), Sweden; University of Genova, Italy; University of Rome "La Sapienza", Italy University of Hildesheim, Germany; Gesellschaft für Mathematik und Datenverarbeitung (GMD), Bonn, Germany; University of Amsterdam, The Netherlands; INRIA Sophia-Antipolis, France; INRIA Rennes, France; University of Sussex, UK (Aalborg University, DK); Weizmann Institute, Israel; State University of New York at Stony Brook, USA; Cornell University, USA.

COST 247

The Formal Systems Group is one of the two Danish representatives in the EU COST workgroup no. 247 on Formal Methods for Verification (our particular interest is on the subgroup on real-time verification and specification). The other Danish representative is TeleDanmark Research.

5.3 Organization and Staff

The tables below give numerical information about our staffing. We show each year, with more detailed half-year information provided by left- or right-shifting of numbers.

Teaching Positions					
	1991	1992	1993	1994	1995
Full Professors			1	1	1
Associate Professors	1.5	1.5	0.5	0.5	0.5 1.5
Assistant Professors	2	2	2	2	2
Research Positions					
Visiting Professors					1
Visiting Researchers	1	1	2	1 2	2
Ph.D. Students	1 2	2 2	2 2	1 1	1 4

Kim Guldstrand Larsen

Academic Degrees

1986 Ph.D. in Computer Science, Edinburgh University, Scotland
1982 Cand.scient. (M.S.) in Mathematics, Aalborg University

Positions

1993- Full Professor in Computer Science, Aalborg University, Institute of Electronic Systems
1992–93 Associate Professor in Computer Science, Aalborg University, Institute of Electronic Systems
1989-92 Senior Researcher (Seniorstipendiat) in Computer Science, Aalborg University, Institute of Electronic Systems
1986–89 Associate Professor in Computer Science, Aalborg University, Institute of Electronic Systems
1985–86 Assistant Professor in Computer Science, Aalborg University, Institute of Electronic Systems
1987–89 Co-chairman of the Department of Mathematics and Computer Science, Aalborg University
1991– Member of the Danish Natural Science Research Council

Kim Guldstrand Larsen has, in this period, served as Program Committee member of a number of conferences (LICS 1993, NWPC, 1989–1995, CONCUR 1990–1992 and 1994, CAV 1991–1994 and 1996, FTFTFR 1996, PARLE 1992), served as reviewer on numerous international journals and conferences. He is presently Steering Committee member of the conferences NWPC, TACAS and CONCUR and is member of the editorial and executive board of the Nordic Journal of Computing. Founding Coordinator and First Year Scientific Manager of CONCUR2. He was Key Note Speaker at CONCUR 1990, Formal Methods Europe 1993, Foundation of Computation Theory 1995, IEEE Real Time Systems Symposium 1995. Co-organizer of 4 international conferences in the period. Reviewer

for European 4th Framework LTR projects. Finally, he has been a member of the Danish Natural Science Research Council since 1991 and serve(d) on several Information Technology Committees under the Ministry of Research.

In 1994, Aalborg University awarded Kim Guldstrand Larsen a bonus (DKK 25,000) for his contributions to research and research administration.

Arne Skou

Academic Degrees

1990	Ph.D. in Computer Science, Aalborg University
1975	Cand.scient. (M.S.) in Computer Science, Copenhagen University

Positions

1989–	Associate Professor in Computer Science, Aalborg University, Institute of Electronic Systems
1986–89	Ph.D. Scholarship, Århus University, Computer Science Department. Visiting University of Sussex, UK, for 7 months during the Ph.D. study.
1986	Research assistant, University of Copenhagen.
1975–86	Member of the senior scientific staff at the computing center of Aalborg University. Head of the computer networking group.

Arne Skou has, in this period, served as reviewer for a number of international journals and conferences. He has co-organized two international conferences. He has been head of department for the period 1993–1995.

Hans Hüttel

Academic Degrees

1991	Ph.D. in Computer Science, University of Edinburgh, Scotland
1988	Cand.scient. (M.S.) in Computer Science, Aalborg University

Positions

1995–	Associate Professor in Computer Science, Aalborg University, Institute of Electronic Systems
1991–95	Assistant Professor in Computer Science, Aalborg University, Institute of Electronic Systems

Hans Hüttel was an invited speaker for the seminar "Automata theory and algorithms," held February 5–12, 1994 at Schloss Dagstuhl, Saarbrücken, Germany. He was an invited speaker for sessions on process calculi at the three-week seminar on the foundations of probability theory at Aalborg University, June 1994.

He also organized the DART workshop on concurrent functional languages in at Aalborg University, November 1993.

In January 1995, he received the award of the board of studies in mathematics, computer science and physics as Teacher of the Year 1994 ("Årets underviser 1994").

Anna Ingolfsdottir

Academic Degrees

1994	Ph.D. in Computer Science, University of Sussex
1987	Cand.scient. (M.S.) in Mathmatics/Computer Science, Aalborg University
1976	B.S. in Mathematics/Physics, The University of Iceland

Positions

1995–	Research Assistant in Computer Science, Aalborg University
1991–95	Assistant Professor in Computer Science, Aalborg University
1990	Research Fellow, The University of Sussex

Luca Aceto

Academic Degrees

1991	Ph.D. in Computer Science, University of Sussex
1986	Laurea (M.S.) in Computer Science, University of Pisa

Positions

1994–	BRICS Research Fellow, Aalborg University, Institute of Electronic Systems
1995	Visiting Research Professor (half a year), Aalborg University, Institute of Electronic Systems
1993	Visiting Researcher (half a year), Aalborg University, Institute of Electronic Systems
1992–	Permanent Lecturer in Computer Science, School of Cognitive and Computing Sciences, University of Sussex
1991–92	Researcher, Hewlett-Packard Lab., Pisa Science Center
1991	Professeur Invité (half a year), Centre de Math., INRIA Sophia Antipolis
1987–90	Research Fellow, Computer Science, University of Sussex

During the period of this evaluation, Luca Aceto has been an invited speaker in four international workshops on concurrency theory, and an invited lecturer at the *Sixth European Summer School in Logic, Language and Information*, 8–19

August, Copenhagen, Denmark. He is presently a member of the steering committee for the EEC Human Capital and Mobility project EXPRESS. He has also participated in the Syllabus Review Committee which has redesigned the degree courses in Computer Science and Artificial Intelligence at the University of Sussex in the period June–July 1993. Finally, in the period January 1994–September 1994, he was CSEURO organizer of the School of Cognitive and Computing Sciences at the University of Sussex, i.e., the member of staff responsible for the connections with other European countries.

Luca Aceto's doctoral dissertation *Action Refinement in Process Algebras* was selected by the British Association of Professors in Computer Science as one of the two Distinguished Dissertations in Computer Science in 1990/1991 and was published by Cambridge University Press in September 1992.

5.4 Collaboration

The Formal Systems Group has put strong emphasis on collaboration throughout the period both on departmental, national as well as international level. In particular, the national projects DART and (later) BRICS as well as the European project CONCUR2 have allowed us to constantly employ project researchers, yielding an important contribution to the group's research environment.

Project Researchers

- Liu Xinxin, –August 1992. Funded by DART.
- Wang Yi, August 1992–May 1993. Funded by DART.
- Luca Aceto, various periods from 1993. Funded by DART and BRICS.
- Ramakrishna, August 1993–January 1994. Funded by CONCUR2.
- Francois Laroussinie, August 1994–July 1995. Funded by CONCUR2.
- Anna Ingolfsdottir, August 1995–. Funded by BRICS.

Short Term Visitors

Throughout the period and with increasing frequency, a number of short term guests (visits longer than one week) have visited the group. In all cases but one (but there is still hope) the visits have resulted in subsequent publications, as can be seen from the publication list.

- Wan Fokkink, Utrecht University, December 1995.
- Paul Pettersson, DoCS, Uppsala University, September–December 1995.
- David Griffeon, CWI, Amsterdam, November 1995.

- Fredrik Larsson, DoCS, Uppsala University, November 1995.
- Johan Bengtsson, DoCS, Uppsala University, November 1995.
- Wang Yi, DoCS, Uppsala University, November 1995.
- Jans Friso Groote, Utrecht University, March 1995.
- Carsten Weise, Aachen University, September, October 1994.
- Karlis Cerans, Riga University, Latvia, October 1992.

Sabbaticals

Three carefully planed sabbaticals have played an important role in maintaining a healthy research group with consolidated and strengthened international contacts.

- Hans Hüttel, Edinburgh University, spring 1993.
- Anna Ingolfsdottir, University of Sussex, spring 1994.
- Kim Guldstrand Larsen, DoCS, Uppsala University, spring 1995.

Individual collaborations

Besides the above, a number of more informal and individual collaborations have taken place during the period. In particular, the Formal Systems Group has constantly been collaborating with member(s) of the Computer Systems Group based on the shared interest in concurrency (though from different perspectives) and in programming languages based on this paradigm. Also, the contacts made via the national projects first DART and now BRICS as well the European project CONCUR2 have been a constant driving force for the research of the group.

5.5 Ph.D. Projects

5.5.1 Overview

During the period 1991–1995, ten Ph.D. students have been associated with the Formal Systems Group. Five of these have successfully completed their Ph.D. studies during the period, one will finish 1996, one in 1997, and the remaining three have just started. Of the ten associated Ph.D. students, three have graduated or will graduate from foreign universities, whereas the remaining seven Ph.D. students have been (or are) advised by members of the Formal Systems Group.

Previously advised Ph.D. students include Klaus Havelund, Jens Christian Godskesen, Liu Xinxin, Anna Ingolfsdottir, and Hans Hüttel. The currently advised Ph.D. students are Josva Kleist, Jørgen Andersen, Kåre Kristoffersen, Ole Høgh Jensen, and Henrik Ejersbo Jensen.

5.5.2 Project Descriptions

The Fork Calculus

Name:	Klaus Havelund
Education:	Cand.scient. (M.S.) Computer Science
Duration:	January, 1991–December, 1993
Status:	Degree awarded March, 1994
Funding:	University of Copenhagen
Advisor:	Kim Guldstrand Larsen

Timed Modal Specifications—A Theory for Verification of Real-Time Concurrent Systems

Name:	Jens Christian Godskesen
Education:	Cand.scient. (M.S.) Computer Science and Mathematics
Duration:	August, 1991–August, 1994
Status:	Degree awarded December, 1994
Funding:	Aalborg University
Advisor:	Kim Guldstrand Larsen

This thesis provides contributions in two main areas: the analysis of behavioral equivalences of real-time processes in the process algebra TCCS with respect to the existence or non-existence of expansion theorems, and the definition of a new specification theory, TMS (Timed Modal Specification, for real-time systems. This theory is a conservative extension of TCCS and has been implemented in the automatic verification tool EPSILON.

Specification and Decomposition in Concurrency

Name:	Liu Xinxin
Education:	B.S. in Computer Science
Duration:	August, 1989–July, 1992
Status:	Degree awarded May, 1992
Funding:	Aalborg University and DART
Advisor:	Kim Guldstrand Larsen

A fundamental issue in program (system) development is that of program correctness. For concurrent systems, the correctness issue becomes a vital one simply due to the behavioral complexity which such systems by exhibit. A main stream of research concerning this problem is to develop formally based methodologies, so that the correctness of such systems can be formally verified. This line of research has been a main focus point for the last ten years within the area of formal semantic theories for concurrency. In order to support verification of large scale systems, it is important that the various semantic theories allow the verification

to be decomposed into verification problems of the components of the systems. This thesis is concerned with the study of formal theories for specifying and verifying concurrent systems, with special emphasis on the decomposition problems associated with these theories.

Semantic Models for Communicating Processes with Value-passing

Name:	Anna Ingolfsdottir
Education:	Cand.scient. (M.S.) in Mathematics and Computer Science
Duration:	1988–1994
Status:	Degree awarded 1994
Funding:	Århus University
Advisor:	Professor Mathew Hennessy, Sussex University.

This Ph.D. project presents semantic models for several process algebras with value passing. First it gives semantic description based on the notion of testing and then based on the notion of bisimulation. In both cases three kinds of semantics are given: behavioral, denotational, and axiomatic semantics. All three approaches are compared.

Decidability, Behavioral Equivalences and Infinite Transition Graph

Name:	Hans Hüttel
Education:	Cand.scient. (M.S.) in Computer Science and Mathematics
Duration:	August, 1988–July, 1991
Status:	Degree awarded 1992
Funding:	Århus University
Study Plan:	At Edinburgh University
Advisor:	Professor Colin Stirling, Edinburgh University

The theme of my Ph.D. was that of verification problems for processes with infinite transition graphs and the new angles that the study of such problems gives us on familiar problems in formal language theory. A problem is *equivalence*—when do two processes exhibit the same behavior? For context-free grammars, this problem is undecidable if one takes "same behavior" to mean language equivalence. However, if one considers the bisimulation equivalence of Park and Milner and considers the transition graph with vertices corresponding to sentential forms and edges given by leftmost derivation steps, the equivalence problem for irredundant context-free grammars is decidable. This result was proved in my thesis. As a matter of fact, it turns out, as was also shown in my thesis, that no other known behavioral equivalences have a decidable equivalence problem for this class of transition graphs. The class of context-free grammars corresponds

exactly to the process calculus BPA (first studied by Bergstra and Klop) which has sequential composition, nondeterministic choice, and recursion.

Models for Object Orientation

Name:	Josva Kleist
Education:	Cand.scient. (M.S.) in Computer Science
Duration:	August, 1995–August, 1998 (expected)
Status:	In progress
Funding:	BRICS
Advisor:	Kim Guldstrand Larsen and Hans Hüttel

Whereas the use of objects has turned out to be a natural way to structure programs, the theoretical foundation for object orientation is still in its infancy. The goal of this Ph.D. project is to develop a semantic model for object orientation explicitly and naturally dealing with inheritance, polymorphism, and encapsulation. Furthermore a better understanding of typing and type inference for object oriented languages i aimed at.

Parameterized Process Algebras and Logic

Name:	Jørgen H. Andersen
Education:	Cand.scient. (M.S.) in Computer Science
Duration:	August, 1994–August, 1997 (expected)
Status:	In progress
Funding:	BRICS and Aalborg University.
Advisor:	Kim Guldstrand Larsen

My thesis work will include a study of general extensions of the classical pure calculi and modal logics to take into account the variety of parameters of extensions such as value-passing, real-time, and hybrid systems. The focus will be how to extend existing and/or provide new algorithms for automatic verification in the more general settings. This is, however, quite an ambitious goal which in most cases will be impossible to reach as problems become undecidable with further expressiveness of the modeling languages. When this becomes the case we are confronted with the choices of

- either give up expressive power of the formalisms to obtain decidability, or
- design verification strategies which can be used for semi-automatic verification.

Tools like EPSILON and TAV are products of the first choice. A tool like ALF is product of the second choice. These are in a sense extremes as the first often are too restrictive and the latter is too general and therefore requires too much

interaction. As my focus is on generality of specific kinds of calculi and logics my aim is somewhere in between the two. In parallel with the evolution of theory I will provide prototype tools implementing the theory.

Compositional Specification and Analysis of Real-Time and Hybrid Systems

Name:	Kaare Kristoffersen
Education:	Cand.scient. (M.S.) in Computer Science
Duration:	August, 1995–July, 1998 (expected)
Status:	In progress
Funding:	The Danish Technical Science Research Council
Advisor:	Kim Guldstrand Larsen and Arne Skou

We are interested in identification of efficient techniques for representation and analysis of Hybrid Systems, that is complex systems such as real-time communicating systems and Process Control Systems. Recent results point in the direction of symbolic and compositional techniques, and we are currently investigating the effect of using BDD-like techniques in this area.

Hybrid Systems

Name:	Henrik Ejersbo Jensen
Education:	Cand.scient. (M.S.) in Computer Science
Duration:	August, 1995–July, 1998 (expected)
Status:	In progress
Funding:	Aalborg University
Study plan:	June, 1996–January, 1998 at M.I.T.
Advisor:	Kim Guldstrand Larsen

The topic characterizing my field of study is the use of formal methods in the development of hybrid systems. The term "hybrid system" denotes a system consisting of computer units embedded in continuous environments. Examples of hybrid systems include control processes in planes, nuclear reactors, hospital systems etc.

My main fields of interest include development of formal modeling and specification languages for hybrid systems. Special attention will be given to the development of automatic tools to support verification between specifications and design of hybrid systems.

An important issue in the development of formal hybrid models is to verify their use in real applications, and hence it is essential to analyse "real-world" problems by the developed methods.

Finally, I am also interested in the construction of implementation languages for programming hybrid systems (i.e., to establish some kind of refinement from

the formal model to an executable implementation).

The main part of Ph.D. study (two years) will be spent working with Professor Tom Henzinger who is *the* expert on formalisms for and automatic verification of hybrid systems.

Action Structures

Name:	Ole Høgh Jensen
Education:	Cand.scient. (M.S.) in Computer Science
Duration:	August, 1993–July, 1996 (expected)
Status:	In progress
Funding:	Danish Natural Science Research Council
Study plan:	At Edinburgh University.
Advisor:	Professor Robin Milner, Cambridge University

5.6 Services and Research-related Activities

Nordic Workshop on Programming Theory Series

Members of the Formal Systems Group have been the initiators of this Nordic workshop series (started 1989) and have acted as Steering Committee members over the entire period. The intention of this annual workshop is to function as an informal meeting place for Nordic (junior) researcher in the area of programming correctness and theory.

Nordic Journal of Computing

In 1993 the Scandinavian Journal BIT was divided into two, resulting in the emergence of the new Nordic Journal of Computing. Members of the Formal Systems Group have played an active role in the creation of this new journal and serve as members of the Editorial and Executive Boards of the journal.

CONCUR Conference Series

The CONCUR conferences were initiated during the CONCUR1 project (the predecessor to CONCUR2) but have now become established as the main conference on concurrency theory, with no particular connection to the CONCUR2 project. Members of the Formal Systems Group serve as members of the Steering Committee of the conference series.

CAV'91 - Computer Aided Verification '91

The third conference of this international series was organized by the Formal Systems Group at Aalborg University. It was attended by 110 researchers from all continents, and the presentations included 42 refereed papers plus 2 invited talks by Joseph Sifakis, IMAG Grenoble, France, and Colin Stirling, University of Edinburgh, Scotland. The proceedings were published in the LNCS series. Special sessions for demonstration of verification tools were scheduled.

The objective of this series of workshops is to bring together researchers and practitioners interested in the development and use of methods, tools, and theories for automatic verification of (finite) state systems. In particular, the workshop provides a unique opportunity for comparing the numerous verification methods and associated verification tools, and the extent to which they may be utilized in application design. The emphasis is not only on new research results but also on the applications of existing results to real verification problems.

The workshop was sponsored by the Danish National Research Council and by Aalborg University through the basic research project Programming Environments in Theory and Practice. The workshop series has official recognition from IFIP WG10.2 (Hardware Description Languages).

TACAS '95 - Tools and Algorithms for The Construction and Analysis of Systems '95

This workshop was the first in a new international series. It was organized by BRICS (and, in particular, by the Formal Systems Group at Aalborg). The workshop was attended by 50 researchers from all continents, and the program contained 24 refereed papers. The aim of the workshop series is to bring together researchers and practitioners interested in the development and application of tools and algorithms for the specification, verification, analysis, and construction of distributed systems. The overall goal of the workshop is to compare the various methods and the degree to which they are supported by interactive or fully automatic tools. Special sessions for the demonstration of verification tools were arranged.

Program Committees

- NWPT 95: *Nordic Workshop on Programming Theory*, Göteborg, Sweden, 1995 (KGL)
- CONCUR 95: *International Conference on Concurrency Theory*, Philadelphia, 1995 (ASK)
- NWPT 94: *Nordic Workshop on Programming Theory*, Århus, Denmark, 1994 (KGL)
- CAV 94: *Sixth International Conference Computer Aided Verification*, Stanford University, U.S.A., July 1993 (KGL)
- CONCUR 94: *International Conference on Concurrency Theory*, Uppsala, Sweden, 1994 (KGL)
- NWPC 93: *Nordic Workshop on Program Correctness*, Turku, Finland, 1993 (KGL)
- LICS 93: *8th Annual Symposium on Logic in Computer Science*, Philadelphia, USA (KGL)
- CAV 93: *Fifth International Conference Computer Aided Verification*, Elounda, Greece, July 1993 (KGL)
- CAV 92: *Forth International Conference Computer Aided Verification*, Montreal, Canada, July 1993 (KGL)
- NWPC 92: *Nordic Workshop on Program Correctness*, Oslo, Norway, 1992 (KGL)
- CONCUR 92: *International Conference on Concurrency Theory*, Stony Brook, NY, USA, 1992 (KGL)
- NWPC 91: *Nordic Workshop on Program Correctness*, Göteborg, Sweden, 1991 (KGL)
- CONCUR 91: *International Conference on Concurrency Theory*, Amsterdam, The Netherlands, 1991 (KGL)
- CAV 91: *Third Workshop on Computer Aided Verification*, Aalborg, Denmark, July 1991 (KGL)

Invited Presentations

- IEEE Real Time Systems Symposium 95, Pisa, Italy, December 1995 (KGL)
- Foundation of Computation Theory 95, Dresden, Germany, 1995 (KGL)
- Sixth European Summer School in Logic, Language and Information, 8–19 August, Copenhagen, Denmark (LA)
- First Express Workshop, Amsterdam, The Netherlands, 1994 (KGL)
- Formal Methods Europe 93, Odense, Denmark, 1993 (KGL)
- CEDISYS Workshop in Sophia Antipolis, Antibes, France, September 1991 (KGL)
- Chalmers Workshop on Concurrency, Båstad, Sweden, May 1991 (AI, LA, ASK, KGL)
- 3rd Workshop on Concurrency and Compositionality, Goslar, Germany, March 1991 (LA, KGL)

Refereeing Services

Members of the group have acted as referees for the following journals and international conferences.

Journal Refereeing

- *Acta Informatica*, 1990–1995
- *Advances in Petri Nets*, 1990–1995
- *Distributed Computing*, 1990
- *Formal Aspects of Computing*, 1991–1995
- *Information and Computation*, 1991–1995
- *Journal of the ACM*, 1991–1995
- *Journal of Mathematical Structures in Computer Science*, 1991–1995
- *Mathematical Systems Theory*, 1994–1995
- *Nordic Journal on Computing*, 1994–1995
- *Theoretical Computer Science*, 1990–1995
- *Transactions on Programming Languages and Systems*, 1994–1995

Conference Refereeing

- *Logic in Computer Science* (LICS), 1990–1995

- *International Conference on Concurrency Theory* (CONCUR), 1990–1995
- *International Conference on Automata, Languages and Programming* (ICALP), 1990–1995
- *Theory and Practice of Software Technology* (TAPSOFT), 1990–1995
- *Computer Aided Verification* (CAV), 1991–1995
- *Algebraic Methodology and Software Technology* (AMAST), 1994–1995
- *Principles of Programming Languages* (POPL), 1994–1995
- *Mathematical Foundations of Computer Science* (MFCS), 1990–1995
- *Mathematical Foundations of Programming Semantics* (MFPS), 1990–1994
- *Conference on Foundations of Software Technology and Theoretical Computer Science* (FST/TCS), 1995
- *Conference on Formal Description Techniques* (FORTE), 1990–1995
- *Protocol Specification, Testing, and Verification* (PSTV), 1991–1995
- *PARLE Conference*, 1991–1994

Evaluation Committees

- Anders P. Ravn, Danish Doctor Degree, Danish Technical University, 1995 (KGL)
- Martin Weichert, Lic., Chalmers Technical University, 1995 (LA)
- Claus T. Jensen, Ph.D., Aarhus University, 1994 (KGL)
- Janna Elroanta, Lic. and Ph.D., University of Helsinki, 1994 (KGL)
- Hakan Erdogmus, Ph.D., Quebec University, Canada, 1993 (KGL)
- Steven Klusener, Ph.D., Centrum for Wiskunde et Informatice, Amsterdam, 1993 (KGL)
- Jens Nordal, Ph.D., DTU, 1993 (KGL)
- Madhavan Mukund, Aarhus University, 1992 (KGL)
- Liang Chen, Ph.D., University of Edinburgh, 1992 (KGL)
- Anders Gammelgård, Ph.D., Aarhus University, 1992 (KGL)
- Hans Hansson, Ph.D., Uppsala University, 1991 (KGL)
- Wang Yi, Ph.D., Chalmers Technical University, 1991 (KGL)
- Ivan Christoff, Ph.D., Uppsala University, 1991 (KGL)

5.7 The Group's Own Evaluation

To our mind, the research activities carried out by the Formal Systems Group are very satisfactory, in terms both of their quality and quantity. In particular, the group has consolidated its international standing within the research area of formal models for reactive, real-time, and hybrid systems. The increased visibility of the group at a national and international level is witnessed by the high degree of collaborative work carried out by the members of the group, the invited presentations given by its members at international events, and the projects that the group has contributed, and is contributing, to. The group has been an active member of the Esprit project CONCUR2, and is currently one of the two sites of BRICS (Basic Research in Computer Science), a center of the Danish national research foundation. The number of the publications by the members of the group has remained very high. In fact, we think that both the number and quality of the group's publications have benefited from the high level of international cooperation of the group's members during the evaluation period. The high quality of the group's research output is witnessed by its appearance in the top-ranking archival journals in the field of theoretical computer science (e.g., Journal of the ACM, Information and Computation, Theoretical Computer Science), and in the proceedings of very competitive conferences (e.g., LICS and ICALP). Because of the speed at which results and papers emerge in our particular field, the general trend in the research community is to use international conferences as a means of quick dissemination of new developments. For this reason, the most of the group's research publications consist of papers published in the proceedings of international conferences. We are, however, very pleased to note that the number of papers published by the members of the group in archival journals of high quality has considerably increased with respect to the previous evaluation period. To our mind, this is proof of the fact that many of the ideas and results presented by the group at international conferences in the past have been considered by our research community to have reached a degree of maturity that makes them worthy of journal publication. This is particularly satisfying because we still consider journal publication as the acid test of the validity of our papers.

In comparison with the previous evaluation period, many more people have contributed to the group's research. However, we note that all but two of the members of the group have been on temporary (research) positions. The group has been able to attract researchers from other research institutions, and these have given their relevant contributions to the group's research. This is a very positive development, as these people bring new ideas and approaches to aid the group's scientific development. However, the research strength of the group cannot rely on the assumption that the flow of visiting researchers will continue. So far, the group has enjoyed good and consistent funding from national and European funding agencies, and most of these funds have been invested in man-power for research—an approach which has paid handsome dividends. However, the

number of teaching staff in the group has actually decreased, and, in conjunction with the unreliability of the intake of funding for the future, we are led to believe that the group will be under-manned in the near future.

Another positive consequence of the high level of funding enjoyed by the group is that the current number of Ph.D. students is high. This is a very good development and bodes well for the group's scientific future.

The main weakness that we perceive in the group's research during the evaluation period is the comparatively low number of papers documenting applications of the group's techniques and tools to industrial case studies. We perceive a tendency in our research area towards industrial application of formal methods for concurrency, and the importance of these case studies is destined to grow. For this reason, we believe that the group will have to improve its connections with industrial enterprises both nationally and on a European scale. Present experiences strongly suggest that this is also the only way to continue to secure the consistently high level of funding that has been so vital for the success of the group during the last five years.

5.8 Research Plan 1996–2000

As previously mentioned, a consistent, high level of funding and collaborations on an national and international scale are vital to the success of the group's research. For this reason, on a national level we shall endeavour to continue our participation in the BRICS project. Moreover, we hope to play an active r ole in the national IT center that is being set up. As part of our involvement in the BRICS project, we have concrete plans to host one of the major European conferences in theoretical computer science. We expect that the organization of this conference will further strengthen the group's position in the European theoretical computer science community. At a European level, we are currently examining several ideas to establish successors of the successful CONCUR2 project. In particular, we plan to give more emphasis to the development of industrially viable approaches driven by problems arising in real-life case studies. To this end, we are developing links with Bang & Olufsen, whom we expect will be an ideal provider of industrially relevant case studies on which to test our verification techniques and tools.

The research carried out by the group is traditionally focused on three main areas: the development of specification formalisms and verification methods, design and implementations of tools, and applications to case studies. We will continue our current efforts in all these areas, but more emphasis than in the past will be placed on industrial case studies.

Concerning the development of specification formalisms and verification methods, we plan to complete our understanding of process algebraic languages and automata-based approaches extended with "non-classical" features. In particu-

lar, we plan to focus on the study of hybrid systems, i.e., systems in which computer programs interact with continuous components. This appears to be a very promising area of research whose results are already been applied to industrially relevant case studies from the fields of consumer electronics, embedded systems, and data-communication protocols. At the same time, we will aim at consolidating our understanding of the theory of classical process description languages. The theory of such languages is now fully mature, and we are seeing a trend towards the development of, on the one hand, very specific techniques which only apply to particular classes of systems, and, on the other, of very general results about whole classes of programming and specification languages. We expect that the group will contribute to both of these lines of research. More specifically, we shall

- study the theory of pushdown processes,
- complete our understanding of classes of infinite state processes,
- consolidate our understanding of axiomatic characterizations of process equivalences over specific classes of processes,
- identify classes of languages that are guaranteed to generate context-free processes, and
- relate specification formalisms based on mobility to the calculi of objects that are rapidly emerging in the research literature on the foundations of object-oriented programming.

Regarding tool development, in a joint undertaking with the Department of Computing Sciences at Uppsala University, we are currently developing a tool suite for the verification of real-time and hybrid systems, UPPAAL. The tool UPPAAL is showing considerable promise in that its prototype versions have been able to efficiently handle fairly complex benchmark problems. One of our primary aims will be to pursue the development of UPPAAL to the limit, and to apply it to industrially relevant case studies.

The plans outlined above are indeed ambitious and far-reaching. In achieving them, the issue of the human resources of the group will play an important part. We hope that the flow of visiting researchers to the Formal Systems Group will continue in the future, but the achievement of our objectives can only be reached by strengthening the group with new permanent members of staff. For this reason we expect to increase the number of teaching faculty members in the group.

Bibliography

Edited Books

[1] E. Brinksma, W. R. Cleaveland, K. G. Larsen, T. Margari, A. Skou, and B. Steffen, editors. *Tools and Algorithms for the Construction and Analysis of Systems*, Volume 1019 of *Lecture Notes in Computer Science*. Springer Verlag, 1995. Appeared earlier as BRICS Notes Series NS–95–2.

[2] U. H. Engberg, K. G. Larsen, and P. D. Mosses, editors. *Proceedings of the 6th Nordic Workshop on Programming Theory*, number NS–94–6 in BRICS Notes Series, 1994.

[3] J. Stage, K. Nørmark, and K. G. Larsen, editors. *Quality Software: Concepts and Tools*, 1993.

[4] K. G. Larsen and A. Skou, editors. *Proceedings of Computer Aided Verification 1991*, Volume 575 of *Lecture Notes in Computer Science*. Springer Verlag, 1992.

Refereed Journal Articles

[5] A. Ingólfsdóttir. Late and early semantics coincide for testing. *Theoretical Computer Science*, 146(1–2):341–349, 1995.

[6] A. Børjesson, K. G. Larsen, and A. Skou. Generality in Design and Compositional Verification using TAV. *Journal of Formal Methods of Systems Design*, 6:239–258, 1995.

[7] L. Aceto. GSOS and finite labelled transition systems. *Theoretical Computer Science*, 131:181–195, August 1994.

[8] L. Aceto. A static view of localities. *Journal of Formal Aspects of Computing Science*, 6(2):202–222, August 1994.

[9] L. Aceto. On "Axiomatising Finite Concurrent Processes". *SIAM Journal on Computing*, (4):852–863, August 1994.

[10] L. Aceto, B. Bloom, and F. W. Vaandrager. Turning SOS rules into equations. *Information and Computation*, 111(1):1–52, May 1994.

[11] B. Steffen and A. Ingólfsdóttir. Characteristic formulae for processes with divergence. *Information and Computation*, 110(1):149–163, April 1994.

[12] L. Aceto and M. Hennessy. Adding action refinement to a finite process algebra. *Information and Computation*, 115(2):179–247, December 1994.

[13] K. Havelund and K. G. Larsen. The Fork Calculus. *Nordic Journal of Computing*, 1:346–363, 1994.

[14] J. F. Groote and H. Hüttel. Undecidable equivalences for basic process algebra. *Information and Computation*, 115:354–371, 1994.

[15] M. Hennessy and A. Ingólfsdóttir. Communicating processes with value-passing and assignment. *Journal of Formal Aspects of Computing Science*, 5:432–466, 1993.

[16] M. Hennessy and A. Ingólfsdóttir. A theory of communicating processes with value-passing. *Information and Computation*, 107(2):202–236, 1993.

[17] L. Aceto and M. Hennessy. Towards action refinement in process algebras. *Information and Computation*, 103(2):204–269, April 1993.

[18] K. G. Larsen. The expressive power of implicit specifications. *Theoretical Computer Science*, 114:119–147, 1993.

[19] S. Christensen and H. Hüttel. Decidability issues for infinite-state processes – a survey. *Bulletin of the EATCS*, 51:156–166, October 1993.

[20] L. Aceto and M. Hennessy. Termination, deadlock and divergence. *Journal of the ACM*, 39(1):147–187, January 1992.

[21] K. G. Larsen and R. Milner. A complete protocol verification using relativized bisimulation. *Information and Computation*, 99(1), 1992.

[22] K. G. Larsen and G. Winskel. Using information systems to solve recursive domain equations. *Information and Computation*, 91(2), 1991.

[23] K. G. Larsen and A. Skou. Bisimulation Through Probabilistic Testing. *Infomation and Control*, 94(1), 1991.

[24] K. G.Larsen and B. Thomsen. Partial Specifications and Compositional Verification. *Theoretical Computer Science*, 88:15–32, 1991.

[25] K. G. Larsen and L. Xinxin. Compositionality Through an Operational Semantics of Contexts. *Journal of Logic Computation*, 1(6):761–795, 1991.

[26] H. Hüttel. *SnS* can be modally characterized. *Theoretical Computer Science*, 74:239–248, 1990.

[27] S. Christensen, H. Hüttel, and C. Stirling. Bisimulation equivalence is decidable for all context-free processes. *Information and Computation*, 121:143–148.

Refereed Conference Articles

[28] K. G. Larsen. UPPAAL — A Tool for Verifying Real-Time Systems. *Lecture Notes in Computer Science*, 1995. In Proceedings of Workshop on Verification and Control of Hybrid Systems, DIMACS.

[29] K. G. Larsen, P. Pettersson, and W. Yi. Diagnostic Model-Checking for Real-Time Systems. *Lecture Notes in Computer Science*, 1995. In Proceedings of Workshop on Verification and Control of Hybrid Systems, DIMACS.

[30] K. G. Larsen, B. Steffen, and C. Weise. Fischer's Protocol Revisited: A Simple Proof Using Modal constraints. *Lecture Notes in Computer Science*, 1995. In Proceedings on Verification and Control of Hybrid Systems, DIMACS.

[31] A. Ingólfsdóttir and A. Schalk. A fully abstract denotational model for observational congruence. To appear in the *Proceedings of CSL 95*, Springer-Verlag.

[32] L. Aceto and A. Ingólfsdóttir. CPO models for a class of GSOS languages. *Lecture Notes in Computer Science*, Vol. 915, pp. 439–453, Springer-Verlag, 1995.

[33] J. C. Godskesen and K. G. Larsen. Synthesizing Distinguishing Formulae for Real-Time Systems. *Lecture Notes in Computer Science*, Vol. 969, 1995. In Proceedings of Mathematical Foundations of Computer Science.

[34] J. H. Andersen, K. J. Kristoffersen, K. G. Larsen, and J. Niedermann. Automatic Synthesis of Real Time Systems. *Lecture Notes of Computer Science*, Vol. 944, 1995. In Proceedings of 22nd International Colloquium on Automata, Languages and Programming (ICALP).

[35] K. G. Larsen, B. Steffen, and C. Weise. A Constraint Oriented Proof Methodology. *Lecture Notes in Computer Science*, Vol. 1019, 1995. In Proceedings of the Workshop on Tools and Algorithms for the Construction and Analysys of Systems (TACAS).

[36] K. G. Larsen, F. Laroussinie, and C. Weise. From Timed Automata to Logic — and Back. *Lecture Notes in Computer Science*, Vol. 969, 1995. In Proceedings of Mathematical Foundations of Computer Science 1995.

[37] K. G. Larsen and F. Laroussinie. Compositional Model Checking Real-Time Systems. *Lecture Notes in Computer Science*, Vol. 962, 1995. In Proceedings of CONCUR95.

[38] K. G. Larsen, P. Pettersson, and W. Yi. Model-Checking for Real-Time Systems. *Lecture Notes in Computer Science*, Vol. 965, 1995. Invited talk at Fundamentals of Computation Theory.

[39] K. G. Larsen, P. Pettersson, and W. Yi. Compositional and Symbolic Model-Checking of Real-Time Systems. To appear in *Proceedings of the 16th IEEE Real-Time Systems Symposium*, Pisa, Italy, December 1995.

[40] C. H. Kristensen, J. H. Andersen, and A. Skou. Specification and automated verification of real-time behaviour – a case study. In *3rd IFAC/IFIP workshop on Algoritms and Architectures for Real-Time Control (AARTC '95)*, pp. 613–628, Belgian Federation of Automatic Control, June 1995.

[41] M. Hansen, H. Hüttel, B. Jensen, and J. Kleist. Inferring effect types in an applicative language with asynchronous concurrency. In U. S. Reddy, editor, *Proceedings of Second ACM SIGPLAN Workshop on State in Programming Languages*, number UILI-ENG-95-1702 in Department of Computer Science, University of Illinois at Urbana-Champaign Tech. Rep., pp. 49–64, ACM, January 1995.

[42] M. Hansen, H. Hüttel, and J. Kleist. Bisimulations for asynchronous mobile processes. In *Tbilisi Symposium on Language, Logic, and Computation*, 1995.

[43] J. C. Godskesen, K. G. Larsen, and A. Skou. Verification of Real-Time Applications Using The Epsilon System. In *Nordic Seminar on Dependable Computing Systems – NSDCS '94*, pp. 41–52, Technical University of Denmark, August 1994.

[44] K. Havelund and K. G. Larsen. A refinement logic for the fork calculus. In *International IFIP Symposium on Protocol Specification, Testing and Verification*, 1994.

[45] J. C. Godskesen, K. G. Larsen, and A. Skou. Automatic Verification of Real-Time Systems Using Epsilon. In S. T. Voung and S. T. Chanson, editors, *Fourteenth International IFIP Symposium on Protocol Specification, Testing and Verification – PSTV '94*, pp. 232–330, Chapman and Hall, June 1994.

[46] L. Aceto. Deriving complete inference systems for a class of GSOS languages generating regular behaviours. In Jonsson and Parrow [81], pp. 449–464.

[47] H. Hüttel. Undecidable equivalences for basic parallel processes. In M. Hagiya and J. C. Mitchell, editors, *Proceedings of TACS '94, LNCS 789*, pp. 454–464, 1994.

[48] K. G. Larsen and W. Yi. Time abstracted bisimulation: Implicit specifications and decidability. *Lecture Notes in Computer Science*, Vol. 802, 1993. In Proceedings of Mathematical Foundations of Programming Systems.

[49] K. Havelund and K. G. Larsen. The fork calculus. *Lecture Notes in Computer Science*, Vol. 700, 1993. In Proceedings of ICALP 1993. Selected for special issue of Nordic Journal of Computing.

[50] K. Cerans, J. C. Godskesen, and K. G. Larsen. Timed modal specifications — theory and tools. *Lecture Notes in Computer Science*, Vol. 697, 1993. In Proceedings of Computer Aided Verification 1993.

[51] O. H. Jensen, C. Jeppesen, J. Lang, and K. G. Larsen. Model Construction for Implicit Specifications in Modal Logic. *Lecture Notes in Computer Science*, Vol. 715, 1993. In Proceedings of CONCUR 1993.

[52] L. Aceto and D. V. J. Murphy. On the ill-timed but well-caused. In Best [82], pp. 97–111.

[53] K. G. Larsen and A. Skou. Testing and Verification of Probabilistic Processes. In P. Heegaard and B. Helvik, editors, *Nordic Seminar on Dependable Computing Systems, NSDCS '92*, pp. 235–250, NTH, Trondheim, 1992.

[54] A. Ingólfsdóttir and B. Thomsen. Semantic models for CCS with values. In *Proceedings of the Workshop on Concurrency, Båstad, Sweden.* Programming Methodology Group, Chalmers University of Technology, 1992. Technical Report 63.

[55] K. G. Larsen. Efficient local correctness checking. *Lecture Notes in Computer Science*, Vol. 663, 1992.

[56] K. G. Larsen and A. Skou. Compositional Verification of Probabilistic Processes. *Lecture Notes in Computer Science*, Vol. 630, 1992. In Proceedings of CONCUR 1992.

[57] J. C. Godskesen and K. G. Larsen. Real-Time Calculi and Expansion Theorems — Extended Abstract. *Springer Verlag Series on Workshops in Computing*, 1992. In Proceedings of NAPAW'92.

[58] K. G. Larsen. Efficient Local Correctness Checking. *Lecture Notes in Computer Science*, Vol. 663, 1992. In Proceedings of Computer Aided Verification 1992.

[59] J. C. Godskesen and K. G. Larsen. Real-time calculi and expansion theorems. *Lecture Notes in Computer Science*, Vol. 652, 1992. In Proceedings of FST/TCS 1992.

[60] W. Yi and K. G. Larsen. Testing probabilistic and nondeterministic processes. In *International IFIP Symposium on Protocol Specification, Testing and Verification — PSTV92*, 1992.

[61] A. Børjesson, K. G. Larsen, and A. Skou. Generality in Design and Compositional Verification Using TAV. In M. Diaz and R. Groz, editors, *Formal Description Techniques, V – FORTE '92*, pp. 449–464, North-Holland, October 1992.

[62] L. Aceto. On relating concurrency and nondeterminism. In Brookes et al. [85], pp. 376–402. A full version appeared as Computer Science Technical Report 6/89, University of Sussex, 1989.

[63] L. Aceto, B. Bloom, and F. W. Vaandrager. Turning SOS rules into equations. In LICS92 [84], pp. 113–124.

[64] S. Christensen, H. Hüttel, and C. Stirling. Bisimulation equivalence is decidable for all context-free processes. In W. R. Cleaveland, editor, *Proceedings of CONCUR '92, LNCS 630*, pp. 138–147, Springer-Verlag, August 1992.

[65] K. G. Larsen. The Expressive Power of Implicit Specifications. *Lecture Notes in Computer Science*, Vol. 510, 1991. In Proceedings of the 18th Colloquium on Automata, Languages and Programming (ICALP), 1991. Full version selected for special TCS issue.

[66] B. Jonsson and K. G. Larsen. Specification and Refinement of Probabilistic Processes. *In Proceedings of LICS'91*, 1991.

[67] U. Holmer, K. G. Larsen, and W. Yi. Decidability of Bisimulation Equivalence between Regular Timed Processes. *Lecture Notes in Computer Science*, Vol. 575, 1991. In Proceedings of Computer Aided Verification 1991.

[68] B. Jonsson and K. G. Larsen. On the Complexity of Equation Solving in Process Algebra. *Lecture Notes in Computer Science*, Vol. 493, 1991. Presented at TAPSOFT'91.

[69] L. Aceto and A. Ingólfsdóttir. A theory of testing for ACP. In Baeten and Groote [83], pp. 78–95.

[70] H. Hüttel and C. Stirling. Actions speak louder than words: Proving bisimilarity for context-free processes. In *Proceedings of 6th Annual Symposium on Logic in Computer Science (LICS 91)*, pp. 376–386, IEEE Computer Society Press, 1991.

[71] H. Hüttel. Silence is golden: Branching bisimilarity is decidable for context-free processes. In K. G. Larsen and A. Skou, editors, *Proceedings of CAV 91, LNCS 575*, pp. 2–12, 1991. The full version is available as Report ECS-LFCS-91-173, Department of Computer Science, University of Edinburgh.

Technical Reports

[72] L. Aceto and J. F. Groote. A complete equational axiomatization for MPA with string iteration. Research Report RS–95–28, BRICS, Aalborg University, May 1995. This and the following report are available through the WWW from `http://www.daimi.aau.dk/BRICS/` and via anonymous ftp from `ftp.daimi.aau.dk` in the directory `pub/BRICS`.

[73] L. Aceto and A. Ingólfsdóttir. A complete equational axiomatization for prefix iteration with silent steps. Research Report RS–95–5, BRICS, Aalborg University, January 1995.

[74] A. Ingólfsdóttir. A semantic theory for value-passing processes late approach—Part I: A denotational model and its complete axiomatization. Report RS–95–3, BRICS, Aalborg University, 1995.

[75] A. Ingólfsdóttir. A semantic theory for value-passing processes late approach—Part II: A behavioural semantics and full abstractness. Report RS–95–22, BRICS, Aalborg University, April 1995.

[76] T. J. Hansen, B. Nielsen, J. F. D. Nielsen and A. Skou. Environments and Paradigms for Development of Reliable Distributed Real-Time Software: An Industrial Oriented Research Project. Technical report, Aalborg University, 1994.

[77] A. Ingólfsdóttir. Semantic Models for Communicating Process with Value-Passing. Ph.D. thesis, Computer Science Report 8/94, School of Cognitive and Computing Sciences, University of Sussex, June 1994. Also available as report R–94–2044, Aalborg University.

[78] H. Hüttel and K. G. Larsen. A Dynamic Type System for Higher-Order Processes. Technical Report IR-93–2001, Aalborg University, 1993.

[79] L. Aceto. Action-refinement in process algebras. Ph.D. thesis, Report 3/91, Department of Computer Science, University of Sussex, 1991.

[80] H. Hüttel. Decidability, Behavioural Equivalences and Infinite Transition Graphs. Ph.D. thesis, University of Edinburgh, December 1991. Published as report CST-86–91/ECS-LFCS-91–191.

Cross References

[81] B. Jonsson and J. Parrow, editors. *Proceedings CONCUR 94,* Uppsala, Sweden, Volume 836 of *Lecture Notes in Computer Science*, Springer-Verlag, 1994.

[82] E. Best, editor. *Proceedings CONCUR 93,* Hildesheim, Germany, Volume 715 of *Lecture Notes in Computer Science*, Springer-Verlag, 1993.

[83] J.C.M. Baeten and J.F. Groote, editors. *Proceedings CONCUR 91,* Amsterdam, Volume 527 of *Lecture Notes in Computer Science*, Springer-Verlag, 1991.

[84] *Proceedings* 7^{th} *Annual Symposium on Logic in Computer Science,* Santa Cruz, California, IEEE Computer Society Press, 1992.

[85] S. Brookes, M. Main, A. Melton, M. Mislove, and D. Schmidt, editors. *Mathematical Foundations of Programming Semantics,* 7^{th} *International Conference*, Pittsburgh, PA, USA, March 1991, Volume 598 of *Lecture Notes in Computer Science*, Springer-Verlag, 1992.

[86] J. Leach Albert, B. Monien, and M. Rodríguez, editors. *Proceedings* 18^{th} *ICALP,* Madrid, Volume 510 of *Lecture Notes in Computer Science*, Springer-Verlag, 1991.

Software

[87] TAV (K. G. Larsen, A. Skou). A tool for automatically verifying equivalences and logical properties of finite-state transition systems expressed in CCS.

[88] EPSILON (J. Andersen, J. C. Godskesen, K. Kristoffersen, K. G. Larsen, A. Skou). A tool for automatically verifying refinements and logical properties of real-time systems expressed in Wang Yi's calculus TCCS.

[89] UPPAAL (K. G. Larsen, K. Kristoffersen). A high performance tool for reachability analysing of real-time systems expressed as timed automata.

Chapter 6

Programming Systems

6.1 Profile

The Programming Systems Group covers areas within the subjects of programming languages, programming environments, and hypermedia systems.

In *programming languages* the group has conducted research on

1. The foundations of conceptual programming.
2. A notation to be used in program design and programming language mechanisms for specific examples from conceptual programming, such as transverse activities, complex associations, and roles.
3. A model for active and distributed object-oriented systems together with experimental implementations of the model.
4. A model of the dynamics in object-oriented systems, and the handling of such a model via use of a tool instead of descriptions on paper.

With respect to *programming environments*, the group has conducted research on

1. General object-based environments, with special emphasis on the tasks of configuration management and version control.
2. Hypertext as the basis for both storage, internal representation, and interaction of programs and documentation.
3. Consistency in software system development.

With respect to *hypermedia systems*, the group has conducted research on

1. Hyperbase management systems.
2. Open hypermedia systems.

3. Collaborative authoring environments.

4. Integrated computing environments (e.g., program development environments and digital libraries).

The researchers in the Programming Systems Group have a common background in development of programming languages and tools. During the eighties, Bent Bruun Kristensen took part in the development of the Beta language and the Mjølner programming environment. Kurt Nørmark's original interest was within generic, syntax-directed editing environments. In addition, he has worked with dynamic languages and tools in the Lisp family. Kasper Østerbye's background is object-oriented languages and systems, and specification languages. Uffe Kock Wiil's background is dynamic programming environments, languages and tools, as well as the use of object-oriented tools and techniques for system development. Lars Bendix has a background in programming environments and object-oriented programming languages. His special interest is configuration management and version control.

6.2 Activities and Results

The research activities of the Programming Systems Group have in this period been organized in subgroups, or they have been carried out on an individual basis. The following presents the activities and results of the subgroups and individual researchers.

Conceptual Modeling

Kasper Østerbye and Bent Bruun Kristensen have throughout the period worked on a new approach to programming, called conceptual programming. This activity has two parts. The first deals with the establishment of the research area as such by defining the objectives and its foundation [25]. The foundation includes the definition and interpretation of the notions of concept and phenomena, the abstraction processes classification/exemplification, specialization/generalization, and aggregation/decomposition, as well as the various kinds of relations between the properties of concepts and phenomena that result from these processes. The second part applies the foundation to concrete examples to develop the notations and programming language mechanisms to be used in the design and programming processes. The work has included complex associations, transverse activities, and roles. Complex associations support the organization of object-oriented systems [8, 32]. Here, associations are supported explicitly in object-oriented programming languages, and in particular, complex associations between nested concepts were developed. Transverse activities model the cooperation of objects [24, 7]. The activities in the usual object-oriented models

are object-centric, whereas transverse activities support the description of the interactions among objects as abstractions themselves. Roles support the subjective understanding of the behavior of an object from other accessing objects [41, 33, 10]. Roles support the description of the dynamic change of an object towards the objects accessing it, by allowing the object to play different roles at different times. An experimental implementation of the role concept was carried out in Smalltalk [53].

Programming Environments

During the period covered by this report, Kurt Nørmark and Kasper Østerbye have conducted research on generic, structure-oriented programming environments based on hypertext. This work is a continuation of work in the 1986–1990 period, which is in part documented in [11]. The main vehicle for this research has been the development of a prototype environment called HyperPro [51], of which some early ideas are described in [49]. In the beginning of the period, a substantial amount of effort went into the design and implementation of the first prototype. This work is primarily documented in a design report [46] and in the report [50] about the visions of the project. This project coined the concept of rich hypertext, which is seen as an attractive representation of documents which include both programs and documentation (in a broad sense). The main focus of this research has been a development of interaction techniques for rich hypertext, documented in [18, 40, 2]. In addition, representational issues have been investigated, involving the borderlines between textual and structural representations [12]. One problem with hypertext is that it is graph structured, as opposed to programs, which are tree structured. It is therefore a problem how to specify and check the permitted structures of hypertext. This problem is addressed in [19].

One of the goals of the HyperPro system is to investigate the relationship between documentation and programs. In the HyperPro system, we have investigated several advanced techniques for doing this. Before HyperPro was completed, a hypertext system [52] was implemented which allowed us to investigate the problems using a hypertext system dedicated to documentation of Smalltalk programs [4]. In this system, version control was also addressed [17].

Lars Bendix's contribution to the programming environment research concerns the integration and automation of the tasks in the programming process. The objective of this work is to minimize the effort the programmer must spend on administration, so as to maximize the effort and creativity he can dedicate to the actual programming.

The potential of an object-based approach was investigated in contrast to traditional file-based approaches. Objects are used to encapsulate files and it is thus possible to add attributes, relations and actions to these files. It turns out that integration is primarily facilitated by the presence of attributes and relations. Automation, on the other hand, is obtained through the actions that objects can

carry out. The action mechanism is augmented by triggers, which are well-known from object-oriented programming languages. This way the possibility to react on specific events in the programming process is obtained.

In the research carried out so far, the primary focus has been on how to divide the different activities in the programming process into four categories: configuration management, version control, personnel management, and resource control [1]. Within these categories, work has in particular focused on finding solutions to problems within configuration management and version control [43, 5]. We have also investigated how the basic mechanisms can be used to support cooperative work [23].

We implemented a simple kernel to support an object-based approach to programming environments. On top of this, we implemented services to support the tasks of configuration management and version control. Several prototypes have been built and experiments carried out to gain more experience with the concepts and mechanisms.

Hypermedia Systems

Uffe Kock Wiil and Kasper Østerbye have developed a taxonomy for open hypermedia systems. The purpose of the Flag taxonomy [48] is manifold: (1) to provide a framework to classify and concisely describe individual systems; (2) to characterize what an *open* hypermedia system is; (3) to provide a framework for comparing different systems in a system independent way; and (4) to provide an overview of the design space of open hypermedia systems.

A series of hypermedia system prototypes have been developed by Uffe Kock Wiil to experiment with integration, collaboration, modeling, storage and versioning issues in complex computing environments [55, 56, 57]. Part of the research on Hyperform and HyperDisco have been conducted in collaboration with John Leggett, Texas A&M University.

The Emacs HyperText System (EHTS) [13] is a multiuser hypertext system enabling simultaneous sharing of a hypertext network of nodes and links by providing real-time monitoring and real-time communication and helping users to deal with access contentions. EHTS is comprised of three tools: a text editor, a graphical browser and a hypermedia database called HyperBase. HyperBase [27] is a multiuser database for hypertext systems designed especially to support collaboration among its users by providing fine grained event (notification) and lock mechanisms [22].

In order to investigate the relationship between HyperBase and other database systems for hypertext, a joint paper was written together with the designers of other hyperbase systems [34]. The paper puts forward seven critical issues that must be dealt with in any hyperbase system.

Hyperform [21, 3, 14, 15, 47, 26, 36, 37] is a dynamic, open and distributed multiuser hypermedia application development environment based on the con-

cepts of extensibility, tailorability and rapid prototyping of hypermedia services: data models, hyperbase management systems and system architectures. Hyperform provides a framework of general building blocks that can be extended and tailored at runtime, thus permitting hypermedia application developers to tailor functionality for specific applications. Experiments have shown that Hyperform greatly reduces the effort required to provide customized hypermedia services for distributed multiuser hypermedia applications.

HyperDisco [16] is an object-oriented hypermedia framework for flexible software system integration. The framework provides hypermedia linking services as a means for integration in heterogeneous software systems composed of diverse information repositories, user interfaces, services and tools. The basic idea in the HyperDisco approach is to allow different tools to be integrated in the hypermedia framework at different tool-dependent levels. Instead of providing a single model of integration that all tools must adhere to, we allow each tool its own specialized model of integration and its own specialized protocol for accessing the hypermedia services.

External Activities

The researchers of the Programming Systems Group have participated in the Software Engineering Programme [29] as well as in the Software Factory Programme. These programs have involved researchers from three other research groups in the department.

The results produced by the Programming Systems Group within the Software Engineering Programme include the following issues:

- *Language mechanisms for object-oriented analysis, design, and programming of computer systems*: The development of complex associations, transverse activities, and roles are seen as contributions to this issue.

- *Literate software development in hypertext*: A number of experiments with literate programming in Smalltalk have been carried out. In extension of this, the HyperPro prototype environment is seen as a more general contribution which, through its user interface mechanisms, supports the development of literate programs.

Within the Software Factory Programme, the Programming Systems Group works with object-oriented modeling, with special emphasis on frameworks. In addition, the group contributes with prototypes of software factory environments, with emphasis on structural hypertext representation, open hypertext system architecture, and co-operative work among software developers.

In the LUKAS Network, Bent Bruun Kristensen gave a seminar on "The Programming Language of the Future" and has been involved as a consultant for some of the participating organizations. Kasper Østerbye and Lars Bendix have

also been involved in several LUKAS activities, giving single as well as series of lectures on object oriented programming and version control.

Other Activities

In the beginning of the period, Bent Bruun Kristensen worked with Per Abrahamsen, Torben M. Hagensen and Ulla Villadsen on models, techniques and tools for describing and maintaining consistency during software system development [6].

During the years 1991–1993 Bent Bruun Kristensen worked with Jeppe Sommer on a project on active, persistent object-oriented systems. The project was supported by the The Danish Technical Research Council with a total of DKK 1,100,000. Bent Bruun Kristensen was also the advisor of Jeppe Sommer's Ph.D. project [20].

A year in 1993–1994, Bent Bruun Kristensen was on leave at Monash University, Melbourne, Australia. This activity was in part supported by The Danish Natural Science Research Council with DKK 142,000. In the project, language mechanisms were developed to support the organization of object-oriented systems [8, 32].

In the period Bent Bruun Kristensen also worked with Bo Bai, Kim H. Christensen, Peter Rasmussen and Karsten Thygesen on the design of an experimental programming environment, G–RAPH [45, 42].

In the last year of the period, Kurt Nørmark conducted research on dynamic, object-oriented models. The focus of this research has been the development of an internal representation of a dynamic model, which may be explored by a dynamic exploration tool. The approach taken in this research should be seen as a contrast to the diagrammatic approaches that dominate in the leading OOA&D literature, and in which dynamic models are projected onto static paper presentations.

Late in the period, Bent Bruun Kristensen and Kasper Østerbye worked with Liam Peyton during his stay at Aalborg University on the completion of his Ph.D. thesis.

In a two year period starting from March 1993, Kurt Nørmark was the coordinator of an EEC funded project by the ATOMOS Programme. The result of the project was a prototype, EMMA, of an Emergency and Damage Control System for ships. The prototype was designed in cooperation with the Danish Maritime Institute in Lyngby and Lloyds Register in London. A total 48 man months were spent on the project, divided over three different project participants.

6.3 Organization and Staff

The following table gives an overview of the group's staffing.

Teaching Positions					
	1991	1992	1993	1994	1995
Full Professors					
Associate Professors	2	2	2 3	3	3
Assistant Professors	1	1	1 1	1	1 2
Teaching Assistants				1 1	1
Research Positions					
Visiting Professors					
Visiting Researchers					
Research Assistants					
Ph.D. Students	3 3	3 4	3 1	1 1	1 1

Bent Bruun Kristensen

Academic Degrees

1974 Cand.scient. (M.S.) in Computer Science, Aarhus University

Positions

1993–94 Visiting Professor, Monash University, Melbourne, Department of Computer Science (1 year)

1989– Reading Professor in Computer Science, Aalborg University, Institute of Electronic Systems

1989 Researcher at the Database Systems Group at University of Maryland (3 months)

1986–88 Researcher (part-time) in the Mjølner project at Sysware ApS, Aarhus (3 years)

1984–85 Visiting Associate Professor in Computer Science, University of California, Santa Barbara, Department of Computer Science (1 year)

1980–89 Associate Professor in Computer Science, Aalborg University, Institute of Electronic Systems

1976–80 Assistant Professor in Computer Science, Aalborg University, Institute of Electronic Systems

1974–76 Lecturer in Computer Science, Aarhus University, Computer Science Department

Kurt Nørmark

Academic Degrees

1987	Lic.scient. (Ph.D.) in Computer Science, Aarhus University
1983	Cand.scient. (M.S.) in Computer Science, Aarhus University

Positions

1991–	Associate Professor in Computer Science, Aalborg University, Institute of Electronic Systems
1987–91	Assistant Professor in Computer Science, Aalborg University, Institute of Electronic Systems
1984–86	Visiting Scholar at the Computer Science Department, Stanford University
1984–87	Ph.D. Student, Aarhus University

Kasper Østerbye

Academic Degrees

1989	Lic.scient. (Ph.D.) in Computer Science, Aalborg University
1986	Cand.scient. (M.S.) in Computer Science, Aarhus University

Positions

1993–	Associate Professor in Computer Science, Aalborg University, Institute of Electronic Systems
1989–93	Assistant Professor in Computer Science, Aalborg University, Institute of Electronic Systems
1986–88	Visiting Scholar at the Computer Science Department, Stanford University
1986–89	Ph.D. Student, Aalborg University

Uffe Kock Wiil

Academic Degrees

1993	Ph.D. in Computer Science, Aalborg University
1990	Cand.polyt. (M.E.) in Computer Engineering, Aalborg University

Positions

1993–	Assistant Professor in Computer Science, Aalborg University
1991–92	Visiting Research Associate, Texas A&M University
1990–93	Ph.D. Scholarship, Aalborg University

Lars Bendix

Academic Degrees

1986	Cand.scient. (M.S.) in computer science, Aarhus University

Positions

1995–	Assistant professor in computer science, Aalborg University
1994–95	Teaching assistant in computer science, Aalborg University
1993	Visiting lecturer in computer science, Siegen University
1991–92	Lecturer in information technology, European Business School, Parma, Italy
1989–91	Ph.D. Scholarship, Computer Science Department, Aarhus University (the period was spent in the Computer Science Department, University of Pisa)
1987–89	Consultant in computer science, BIT Consulting, Struer, Denmark
1986–87	Research assistant in computer science, University of Pisa

6.4 Collaboration

Lars Bendix, Kurt Nørmark and Kasper Østerbye collaborate with Scandinavian groups in Linköping and Lund on programming environment topics. As an integrated part of this collaboration bi-annual Nordic Workshops on Programming Environment Research are held. The 1996 workshop in this series will take place in Aalborg with the above mentioned persons as the primary organizers.

During his leave at Monash University Bent Bruun Kristensen collaborated with Daniel C. May on the modeling of transverse activities in concrete object-oriented languages such as C++ [9].

Kasper Østerbye worked with Danny Lange (University of Tokyo, Japan) and Helge Schütt (GMD-IPSI, Darmstadt, Germany) on requirements for hypermedia storage [34].

Uffe Kock Wiil collaborates with John Leggett (Texas A&M University, USA) on the design and development of generic hypermedia system platforms and the deployment of these platforms in various application areas, such as collaborative authoring environments and digital libraries [15, 14, 36, 37].

Uffe Kock Wiil and Kasper Østerbye collaborate with Serge Demeyer (Brussels Free University, Belgium) in the area of open hypermedia systems (OHSs). This collaboration has resulted in the organization of two workshops on OHSs. The 1st OHS workshop took place at the ACM European Hypertext Conference in 1994 [30] and the 2nd OHS workshop will take place at the upcoming ACM Hypertext Conference in March 1996.

Uffe Kock Wiil has collaborated with Jörg Haake (GMD-IPSI, Darmstadt, Germany) and Cathy Marshall (Xerox Parc, USA) in the area of hypermedia

and CSCW. As a part of this, a workshop at the ACM CSCW Conference (1994) on collaborative hypermedia systems was organized [44].

6.5 Ph.D. Projects

Extensibility in Open, Distributed Hypertext Systems

Name:	Uffe Kock Wiil
Education:	Cand.polyt. (M.E.) in Computer Engineering, 1990
Duration:	1990–1993
Status:	Degree awarded April, 1993
Funding:	Faculty of Engineering and Science, Aalborg University
Advisor:	Bent Bruun Kristensen

The purpose of the Hyperform project was to design, develop and experiment with a dynamic, open and distributed hypermedia platform. Hyperform has two major objectives: (1) to serve as a development environment for hypermedia application developers and (2) to serve as a platform for research on important hypermedia issues. The Hyperform prototype is based on the concepts of extensibility, tailorability and rapid prototyping of the following hypermedia services: data models, hyperbase management systems and system architectures. Experiments have shown that Hyperform greatly reduces the effort required to provide customized hypermedia services for distributed multiuser hypermedia applications.

Active, Distributed and Persistent Object-Oriented Systems

Name:	Jeppe Sommer
Education:	Cand.polyt. (M.E.) in Computer Engineering, 1990
Duration:	1991–1993
Status:	Degree awarded June, 1993
Funding:	The Danish Technical Research Council
Advisor:	Bent Bruun Kristensen

The objective of this project was to elucidate the implications of combining active objects with persistence in an object-oriented system. One aspect of this is the demands on programming environments imposed by persistence and distribution. This includes the management of a distributed, persistent scope and a possibly complex topology of communication. The claim is that the problems related to both distribution and persistence are very similar in nature, and that a uniform solution of these problems is feasible.

Configuration Management and Version Control Revisited

Name:	Lars Bendix
Education:	Cand.scient. (M.S.) in Computer Science, 1986
Duration:	–1996
Status:	In progress
Funding:	Faculty of Natural Science, Aarhus University
Advisor:	Bent Bruun Kristensen

The dissertation addresses problems within configuration management and version control. An analysis shows shortcomings in present systems with respect to version control support in workspaces and proper integration of configuration management and version control. We provide full version control support in workspaces by moving version control from the level of whole repositories to the individual version groups, and by changing the checkout operation such that it creates a new version group in the workspace instead of a single component. Proper integration of configuration management and version control is done by considering dependency information as local to each single version of a component instead of as a global property of a version group, as is traditionally done. Therefore, we can model dynamically changing software architectures within one and the same model without suffering the traditional added complexity of performing explicit version control of configuration descriptions.

Understanding Software Automation

Name:	Liam Peyton
Education:	M.S. in Computer Science, 1989
Duration:	–1996
Status:	In progress
Funding:	Danish Research Academy
Advisor:	Bent Bruun Kristensen

The initial insight was that software development could be automated by building explicit models of software in terms of its components and by applying artificial intelligence techniques that incorporated those models. While building several systems based on this idea, a new understanding of software began. Developers build software by building documents that describe how software will behave. It was also noticed that in many organizations the tasks performed with documents were now being performed electronically. This work that centered around documents was often referred to as knowledge work. The thesis is that knowledge work, including software development, can be automated by building explicit models of electronic documents and by applying artificial intelligence techniques that incorporate those models. The result of this research is an architecture for software automation based on document models underlying a domain of knowl-

edge work and the use of templates to support and automate tasks that knowledge workers perform with documents.

6.6 Service and Research-related Activities

The members of the Programming Systems Group serve as reviewers for the following journals and conferences: ACM Transactions on Information Systems; Journal of Management Information Systems; IEEE Computer; Software–Practice and Experience; IEEE Parallel and Distributed Technology. Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA); European Conference on Object-Orented Programming (ECOOP); East-West International Conference on Multimedia, Hypermedia, and Virtual Reality (MHVR); ACM Conference on Hypertext; ACM European Conference on Hypermedia Technology (ECHT); Nordic Workshop on Programming Environment Research; ACM SIGPLAN Symposium on Partial Evaluation and Semantics-Based Program Manipulation; International Workshop on Hypermedia Design; IEEE-CS Hawaii International Conference on System Sciences (HICSS); and ACM/IEEE-CS Symposium on Applied Computing (SAC).

The members of the Programming Systems Group have participated in the organization of a number of conferences and workshops: Uffe Kock Wiil was publicity chair for the International Workshop on Temporal Databases at the 21st International Conference on Very Large Data Bases (VLDB'95); co-organizer for the Workshop on Open Hypermedia Systems [30]; and co-organizer of the Workshop on Collaborative Hypermedia Systems [44]. Kurt Nørmark was on the organizing committee of the European Conference on Object-Oriented Programming, 1995 (ECOOP). Kasper Østerbye was the workshop coordinator for the last three ACM conferences on hypertext; organizer of a workshop on hypertext in engineering [31]; and co-organizer of the open hypermedia systems workshop [30] .

In addition Uffe Kock Wiil and Kasper Østerbye were invited to give talks at the NSF (National Science Foundation, USA) sponsored Workshop on Hyperbase Systems in 1992.

6.7 The Group's Own Evaluation

According to the plans made at the beginning of the period, we intended to relate all major research activities to a common experimental programming environment laboratory. The organization of the laboratory should allow freedom for individual choice of research subjects within this very broad area. We wanted to plan, conduct, and evaluate our work explicitly as experimental work, including both design, implementation, and applications of the system. The laboratory would

serve as a common basis for international collaboration with similar research groups, educational activities, as well as a growing interaction with industry. Most of these objectives have been met at the end of the period.

For a number of reasons, we have not organized our work around a single, common laboratory. One reason is that there is a lot of overhead involved in building and maintaining the common basis. Also with the number of persons involved, at various levels of education and points in career, it is difficult to keep it all together, especially when a number of Ph.D. projects are already well under way,. Finally we will probably not be willing to give up our freedom to choose subjects and to organize our work—at least partially—to obtain the benefits from a common laboratory. The daily activities, varying in number, nature, and intensity, do not fit very well into organized projects with discipline, time limits, etc. On the other hand, we have no problem in seeing our research during the period as belonging to the programming environment laboratory. All activities fit very well into this idea.

The remaining elements of the plan have all been met. We have conducted research on the components and aspects of the laboratory, namely database, user interface, integration, extensibility, and language component. We have to a natural extent organized our work experimentally, and have worked with design, implementation, and applications. The research has been successfully integrated into the curriculum, at the level of M.S. and Ph.D. students. We have had fruitful exchange of ideas internationally—mostly via individual contacts. We have industrial contacts, but we have only to a limited extent succeeded in actual collaboration.

Our primary targets for publication of our research results are the international conferences in the areas of programming environments, object-oriented design and programming, and hypertext. It is our position that the quality of these conferences is at least as high as for periodicals in our area, and that the conferences are the most important medium for international interaction and exchange of ideas.

6.8 Research Plan 1996–2000

Our work in the period from 1996 to 2000 will be organized in groups of 1–2 persons. There will be no overall system to which the activities of such groups will belong. The specific subjects will be chosen on an individual basis according to interests and possibilities for collaboration.

We will continue to publish our research results in high-quality international conferences and (to some extent) international periodicals covering our research areas. We will also continue the international interaction and exchange of ideas with researchers working with similar subjects, and hopefully extend this further in the direction of cooperation on experiments, joint articles, and prolonged visits

to international research groups.

We aim to establish better contacts with the industry. We need a better interaction with relevant settings in the industry in order to promote our theoretical results and to get inspiration and feedback from practical reality. The objective of improved interaction with the industry is still to find inspiration and obtain relevance criteria for our ongoing research.

Here follows a list of planned activities in the period.

Objects, Patterns, and Architectures

Bent Bruun Kristensen and Kasper Østerbye plan to carry out research on objects, patterns, and architectures. Object-oriented methods, patterns, and architectures have a lot of attention because each holds out the promise of addressing the chronic problems of software development: high development costs, even higher maintenance costs, and low levels of reuse. Our research is intended to investigate each of these subjects further and also to examine the relationships between these. A further understanding and further development of object-based conceptual understanding may result in additional different kinds of objects, similar to association objects and subject/role objects. An understanding of the dimensions in the pattern universe may result in different types of patterns for different purposes in object-oriented analysis, design, and implementation. An understanding of software architecture may result in new organizational structures as well as new notational constructs for expressing the organization, and a clarification of the relation of these to objects and patterns.

Dynamic Models in Object-oriented Design

Kurt Nørmark wants to continue the research on tool support of object-oriented design with dynamic models. The thesis behind this work is that programmers think in terms of objects, object relations, and object interactions during the creative phases of the design process. As a consequence of this, we want to experiment with a design technique, in which dynamic design models are worked out prior to the construction of static design models. Our planned approach is to develop a dynamic medium, in terms of a number of tools, via which dynamic models can be constructed and explored.

Hypertext and Software Engineering

Kurt Nørmark and Kasper Østerbye want to continue the ongoing research on hypertext based programming environments. This research will have a theoretical dimension as well as a practical dimension. In the theoretical dimension we want to survey the existing environments which in some way use hypertext technology.

In the practical dimension we wish to continue the experiments with deployment of hypertext concepts in relation to solving practical software engineering problems.

Hypermedia Systems

Uffe Kock Wiil will continue the ongoing research on hypermedia systems. The work is planned to take two major directions: (1) design, development and deployment of hypermedia system prototypes and (2) development of standards and references models for open hypermedia systems. The latest hypermedia system prototype, HyperDisco, will be further developed and deployed in different application areas such as digital libraries. The Flag taxonomy will be expanded and developed into a reference model for open hypermedia systems and we will continue to encourage the open hypermedia system community to meet (by organizing workshops) and work towards standards for open hypermedia systems.

Bibliography

Refereed Journal Articles

[1] Lars Bendix. Fundamental Tasks in Software Development Environments, *Informatica*, 19(3), 1995.

[2] Kurt Nørmark, Kasper Østerbye. Rich Hypertext: A Foundation for Improved Interaction Techniques. *International Journal of Human-Computer Studies*, 43:301–321, 1995.

[3] Uffe K. Wiil. Hyperform: Rapid Prototyping of Hypermedia Services. *Communications of the ACM*, 38(8):109-111, 1995.

[4] Kasper Østerbye. Literate Smalltalk Programming Using Hypertext. *IEEE Transactions on software engineering*, 21(2):138–145, 1995.

Refereed Conference Articles

[5] Lars Bendix. An Integrative Model for Configuration Management and Version Control. In *Proceedings of the 5th International Workshop on Software Configuration Management*, Seattle, Washington, 1995.

[6] Torben M. Hagensen, Bent B. Kristensen. Consistency in Software System Development: Framework, Model, Techniques & Tools. In *Proceedings of the ACM SIGSOFT'92: Fifth Symposium on Software Development Environments*, 1992.

[7] Bent B. Kristensen. Transverse Activities: Abstractions in Object-Oriented Programming. In *Proceedings of International Symposium on Object Technologies for Advanced Software* (ISOTAS'93), 1993.

[8] Bent B. Kristensen. Complex Associations: Abstractions in Object-Oriented Modeling. In *Proceedings of Conference on Object-Oriented Programming Systems, Languages, and Applications* (OOPSLA'94), 1994.

[9] Bent B. Kristensen, Daniel C. M. May. Modeling Activities in C++ and Eiffel. In *Proceedings of International Conference on Technology of Object-Oriented Languages and Systems* (TOOLS PACIFIC 94), 1994.

[10] Bent B. Kristensen. Object-Oriented Modeling with Roles. In *Proceedings of the 2nd International Conference on Object-Oriented Information Systems*, Dublin, Ireland, 1995.

[11] Kurt Nørmark. A Hyperstructure Programming Environment for CLOS. *Technology of Object-Oriented Languages and Systems – TOOLS4*, Prentice-Hall, 1991.

[12] Kurt Nørmark, Kasper Østerbye. Representing Programs as Hypertext. In *Proceedings of Nordic Workshop on Programming Environment Research*, Lund, 1994.

[13] Uffe K. Wiil. Issues in the Design of EHTS: A Multiuser Hypertext System for Collaboration. In *Proceedings of the 25th IEEE Hawaii International Conference on System Sciences*, 1992.

[14] Uffe K. Wiil, John J. Leggett. Hyperform: Using Extensibility to Develop Dynamic, Open and Distributed Hypertext Systems. In *Proceedings of the Fourth ACM Conference on Hypertext*, 1992.

[15] Uffe K. Wiil, John J. Leggett. Concurrency Control in Collaborative Hypertext Systems. In *Proceedings of the Fifth ACM Conference on Hypertext*, 1993.

[16] Uffe K. Wiil. HyperDisco: An Object-Oriented Hypermedia Framework for Flexible Software System Integration. In *Proceedings of the 19th IEEE International Computer Software and Applications Conference*, 1995.

[17] Kasper Østerbye. Structural and cognitive problems in providing version control for hypertext. In D. Lucarella, J. Nanard, M. Nanard, and P. Paolini, editors, *ECHT'92 Proceedings of the ACM conference on hypertext*, pp. 33–42, 1992.

[18] Kasper Østerbye, Kurt Nørmark. An Interaction Engine for Rich Hypertexts. In *Proceedings of the European Conference on Hypermedia Technology*, ACM Press, 1994. An extended version is available as [40].

[19] Kasper Østerbye. Structural Constraints in Rich Hypertext. In *Proceedings of the International Workshop on Hypermedia Design'95*. Published as *Workshops in Computing: Hypermedia Design*, Springer-Verlag 1995.

Ph.D. Theses

[20] Jeppe Sommer. *Active, Distributed and persistent Object-Oriented Systems.* Institute of Electronic Systems, Aalborg University, 1993.

[21] Uffe K. Wiil. *Extensibility in Open, Distributed Hypertext Systems.* Aalborg University, 1993.

Book Chapters

[22] Uffe K. Wiil. Using Events as Support for Data Sharing in Collaborative Work. In Gorling and C. Sattler, editors, *International Workshop on CSCW*, Institut für Informatik und Rechentechnik, Berlin, Germany, 1991.

Unrefereed Journal Articles and Invited papers

[23] Lars Bendix. General Object-Based Environments: A Basis for Cooperative Software Development. In *Proceedings of the 19th International Conference on Information Technologies and Programming*, Sofia, Bulgaria, 1994.

[24] Bent B. Kristensen. Transverse Classes & Objects in Object-Oriented Analysis, Design and Implementation. *Journal of Object-Oriented Programming*, 1993.

[25] Bent B. Kristensen, Kasper Østerbye. Conceptual Modeling and Programming Languages. *Sigplan Notices*, 29(9), 1994.

[26] Uffe K. Wiil. Hyperbase Research at The University of Aalborg. *SIGLINK Newsletter*, 1(2):12–14, 1992.

[27] Uffe K. Wiil. Experiences with HyperBase: A Multiuser Hypertext Database. *SIGMOD RECORD*, 22(4):19–25, 1993.

[28] Kasper Østerbye. Hypertext Servers for Team Environments. *SIGOIS Bulletin*, 13(1):26–27, 1992.

Books and Proceedings Edited

[29] Jan Stage, Kurt Nørmark, Kim G. Larsen, editors. *Quality Software: Concepts and Tools.* Institute of Electronic Systems, Aalborg University, 1994.

[30] Uffe K. Wiil, Kasper Østerbye, editors. *Proceedings of the ECHT '94 Workshop on Open Hypermedia Systems.* Published as Technical Report R 94-2038, Department of Mathematics and Computer Science, Aalborg University.

[31] Kasper Østerbye, editor. *Hypertext in Engineering.* Published as Technical Report R 93-2030, Department of Mathematics and Computer Science, Aalborg University, 1993.

Technical Reports

[32] Bent B. Kristensen. *Abstraction Mechanisms for Object-Oriented Modeling of the Organization and Cooperation of Classes and Objects.* R 94-2001, Institute of Electronic Systems, Aalborg University, 1994.

[33] Bent B. Kristensen. *Object-Oriented Modeling with Roles: A Conference Organizing Example.* R 95-2014, Institute of Electronic Systems, Aalborg University, 1995.

[34] Danny Lange, Kasper Østerbye, Helge Schütt. *Hypermedia storage.* R 92-2009, Department of Mathematics and Computer Science, Aalborg University, 1992.

[35] Kurt Nørmark. *An Evaluation of Eiffel as the first Object-oriented Programming language in the CS Curriculum.* Presented at the workshop "Eiffel in Schule und Hochschule" in Darmstadt, May 1995. R 95-2017, Institute of Electronic Systems, Aalborg University, 1995.

[36] Uffe K. Wiil, John J. Leggett. *Hyperform: An Extensible Hyperbase Management System.* Department of Computer Science, Technical Report TAMU-HRL 92-003, Texas A&M University, College Station, Texas, 1992.

[37] Uffe K. Wiil, John J. Leggett. *Hyperform: A Hypermedia System Development Environment.* R 95-2007, Department of Mathematics and Computer Science, Aalborg University, 1995.

[38] Kasper Østerbye. *Literate Smalltalk Programming using Hypertext.* R 93-2025, Department of Mathematics and Computer Science, Aalborg University, 1993.

[39] Kasper Østerbye. *Hyperstructure Program Development Environments: An Experiment with Literate Programming.* In [31].

[40] Kasper Østerbye, Kurt Nørmark. *Tailorable interaction using the HyperPro interaction engine.* R 94-2017, Institute of Electronic Systems, Aalborg University, 1994.

[41] Kasper Østerbye, Bent B. Kristensen. *Roles.* R 95-2006, Institute of Electronic Systems, Aalborg University, 1995.

Other Publications

[42] Bo Bai, Kim H. Christensen, Bent B. Kristensen, Peter Rasmussen, Karsten Thygesen. *The Experimental Programming Environment G-RAPH: A first running prototype*. IR 92-2004, Institute of Electronic Systems, Aalborg University, 1992.

[43] Lars Bendix. Integrating Version Control and Configuration Management. In *Proceedings of NWPER'94 the Nordic Workshop on Programming Environment Research*, Lund, Sweden, 1994.

[44] Jörg M. Haake, Catherine Marshall, Uffe K. Wiil. Open Issues in Collaborative Hypermedia Systems. In J. M. Haake, editor, *Proceedings of the CSCW '94 Workshop on Collaborative Hypermedia Systems*, GMD-Studien Nr. 239, GMD-IPSI, Darmstadt, Germany, 1994.

[45] Bent B. Kristensen. *The Experimental Programming Environment G-RAPH: A Preliminary Proposal*. IR 92-2002, Institute of Electronic Systems, Aalborg University, 1992.

[46] Kurt Nørmark, Kasper Østerbye. *The Design of a Hyperstructure Programming Environment*. IR 93-2002, Institute of Electronic Systems, Aalborg University, 1993.

[47] Uffe K. Wiil. Towards Industrial Strength Hyperbase Systems. In J. Leggett, editor, *Proceedings of the Hypertext '93 Workshop on Hyperbase Systems*, Department of Computer Science, Technical Report TAMU-HRL 93-009, Texas A&M University, College Station, Texas, 1993.

[48] Uffe K. Wiil, Kasper Østerbye. OHS - A Taxonomy. In Uffe K. Wiil and Kasper Østerbye, editors, *Proceedings of the ECHT '94 Workshop on Open Hypermedia Systems*, Technical Report R 94-2038, Department of Mathematics and Computer Science, Aalborg University, 1994.

[49] Kasper Østerbye, Kurt Nørmark, Hans M. Jeppesen. HyperStructure Programming Environments. In *Proceedings of the Nordic Workshop on Programming Environment Research*, Tampere, 1992.

[50] Kasper Østerbye, Kurt Nørmark. *The Vision and the Work in the HyperPro Project*. IR 93-2012, Institute of Electronic Systems, Aalborg University, 1993.

Software

[51] Kasper Østerbye, Kurt Nørmark. *HyperPro*. HyperPro is the experimental platform for the research in program understanding and hypertext-based

programming environments. It consists of 15k lines of Smalltalk, and 4k lines of Emacs-lisp code. The system is documented in references [12, 18, 2, 46, 50].

[52] Kasper Østerbye. *LiST*. LiST supports literate programming in Smalltalk. It consits of 11k lines of Smalltalk code. It is documented in references [17, 4].

[53] Kasper Østerbye and Bent B. Kristensen. *STROLE*. STROLE implements the role concept in the Smalltalk language. The system consists of 500 lines of Smalltalk code. This system is documented in reference [41].

[54] Kasper Østerbye. *STRelations*. STRelations implements relations for Smalltalk. The system consists of 400 lines of Smalltalk code. The system was developed to test *LiST* [4].

[55] Uffe K. Wiil. *Emacs HyperText System (EHTS)*. EHTS is a multiuser hypertext system supporting asynchronous and synchronous sharing of hypertexts. It consists of 600K bytes of C, C++, and Lisp code and is available from `ftp://ftp.iesd.auc.dk/pub/packages/hypertext/`. The system is documented in references [13, 22, 27].

[56] Uffe K. Wiil. *Hyperform*. Hyperform is a hypermedia system development environment. It consists of 270K bytes of C, Scheme, and Lisp code. The system is documented in references [3, 14, 21, 36, 37].

[57] Uffe K. Wiil. *HyperDisco*. HyperDisco provides an object-oriented framework for flexible software-system integration. HyperDisco is built using Hyperform [56]. The system is documented in reference [16].

Chapter 7

Information Systems

7.1 Profile

The research area of the Information Systems Group is the application of information technology. The focus is on development of computer-based systems and on management and use of information technology in organizations. The research approach is based on theoretical and conceptual studies in close interaction with experimental and empirical research activities. The results include:

- Theoretical contributions in the form of frameworks and principles that increase our understanding of the field.
- Methods in the form of prescriptions that guide practical action.
- Practical results in the form of specific industrial collaborations that improve practice in selected areas of the field.

The Information Systems Group has its roots in a Scandinavian systems development tradition based on a socially responsible approach to computing. From this background the group has, since its start in 1987, developed a profile which combines insights from the information systems field with problems, challenges and approaches from the software engineering field. At the last research evaluation in 1990, a policy was formulated to "remain open towards information systems and organizations, while at the same time emphasizing a narrower orientation towards computer systems and software development." At that time the group consisted of one full professor, three assistant professors, and a couple of research scholars.

7.2 Activities and Results

The stronger orientation towards software engineering has had considerable impact on the activities and results of the group from 1990–1995. The majority

of the contributions of the group during this period fall clearly within software engineering even though most of them are influenced by information systems traditions. The group has published its main results in the form of scientific articles and technical reports. In addition, the group has produced a number of books reflecting its practical and industrial orientation. These books are a combination of academic contributions to the field and practical guidance for computing professionals.

The research activities of the group consist of initiatives in which a few members develop results, often as part of an international collaboration, and initiatives that are organized in larger programs and projects. In the period 1990–1995, the whole group has played an active role in the Software Engineering and the Software Factory Programmes in the department. In addition, two projects have been successfully completed within the group. First, Ivan Aaen and Carsten Sørensen have completed the TEQ (Tools for Efficiency and Quality) project. This project aimed to investigate the practical aspects of Computer Aided Software Engineering (CASE). It included close collaboration with researchers from University of Jyväskylä, Finland, and it resulted in a Ph.D. thesis and a number of scientific articles. Second, Lars Mathiassen, Peter Axel Nielsen, and Jan Stage have completed the first part of the OOA&D (Object Oriented Analysis & Design) project. This project has been carried out in close collaboration with Andreas Munk-Madsen, Metodica, Denmark. The project has in its present stage resulted in a method for object-oriented analysis and design, documented in two books and a few scientific articles. The method is used in many public and private curricula for computer professionals in Denmark, it is being used in some Swedish and Norwegian curricula as well, and the first commercial software organizations have started to use the method as a framework for practical software development.

The contributions of the Information Systems Group from 1990–1995 can be summarized as follows. Most of the contributions focus on software and information systems development [3, 4, 9, 10, 15, 16, 17, 18, 19, 20, 21, 22, 23, 26, 27, 28, 29, 30, 34, 35, 36, 37, 39, 40, 41, 42, 43]. A second major group of contributions addresses various management issues related to the use of computer-based information technology in organizations [1, 7, 8, 11, 12, 13, 14, 25, 31, 33, 38, 46, 48, 49, 50]. Finally, there are a number of contributions addressing more general issues related to information systems and software engineering [2, 5, 6, 24, 32].

7.2.1 The Software Engineering Programme (SE)

The goal of this Programme was to contribute to reducing the software crisis through the creation and testing of techniques and tools for the effective development of high-quality software. The program has been active for five years, the first period from 1990–1992, and the second period from 1993–1994. In both periods, it has been a joint effort between the Information Systems Group and

two other departmental groups. The program has been supported by the Danish Natural Science Research Council with a total of DKK 1.7 million.

During its first three years, the Software Engineering Programme focused on techniques and tools for the effective development of high-quality software covering all aspects of software development, i.e., analysis, design, and implementation. During the last two years, the program has focused more directly on the design activity. The results produced by the Information Systems Group within this program includes research contributions addressing the following issues:

- *Software Engineering Environments:* Qualities, potential applications, and practical use of CASE tools. Development of a preliminary version of a complete CASE tool for our object-oriented analysis and design method.

- *Software Engineering Techniques and Methods:* Conceptual modeling supporting abstraction. Notations and methods for object-oriented analysis and design of computer systems.

- *Software Engineering Management:* Risk-based software development. Quality assurance and management. Software process improvements related to the effective and efficient use of CASE technology.

- *Software Engineering Concepts and Theories:* Software quality. Complexity and uncertainty in software development. Philosophical foundations for software development.

In 1993, the first period of the program was evaluated with a very positive result. In 1995, a new program based on the results of this program was given financial support for at least three years. The new program is described in Section 7.2.4.

7.2.2 Tools for Efficiency and Quality (TEQ)

TEQ was a four-year project to investigate aspects of the practical use of CASE (Computer Aided Software Engineering). The project ran from 1990 to 1994 with Ivan Aaen and Carsten Sørensen as members of the project group.

The objective of TEQ was to contribute to an understanding of the factors affecting the effective and efficient use of CASE-tools in practical settings. The focus has been on factors and relationships affecting the diffusion of CASE in industry, and on the interplay between methods, techniques, and tools in practical settings.

A major part of the project was based on a cooperation with researchers from the Finnish SYTI Group at Jyväskylä University. This collaboration—the CASE Monitor Project—was initiated and lead by the TEQ Group and investigated the use of CASE technology in Denmark and Finland via a questionnaire study. The

findings of the CASE Monitor project have been published in a number of articles, conference proceedings and Ph.D. theses, e.g. [9, 20, 22, 35, 36, 43, 50].

Other results from the TEQ project include discussions on CASE usability [21, 40, 50], literature studies [10, 41], and theoretical contributions [34]. Carsten Sørensen's Ph.D. study [43] was part of the TEQ project.

7.2.3 Object-Oriented Analysis and Design (OOA&D)

The OOA&D project has, so far, lasted for five years. Its purpose is to develop a methodology for the analysis and design of computer systems. The participants are Lars Mathiassen, Andreas Munk-Madsen (Metodica), Peter A. Nielsen, and Jan Stage.

The methodology that has been developed is based on a combination of classical functional thinking and recent object-oriented approaches. The methodology is being developed experimentally through an interplay between theoretical considerations about the form of the methodology and its application in both educational/pedagogical and practical settings.

The results of the project are published in two academic books [3, 4]. The original principles were published in a preliminary version [29]. In addition, one of the main ideas of the analysis methodology has been published in articles [15, 16, 28]. Finally, the basis of the early activities of analysis are described in a separate article [17].

7.2.4 The Software Factory Programme (SF)

The Software Factory Programme, subtitled "Objects, Frameworks and Environments", is conducted by an interdisciplinary group with participants from the department's Programming Systems Group, Database Systems Group, and Information Systems Group. The aim of the program is to develop methodologies, languages, and tools for modern software factories.

The software factory is a software development idea focusing on the high complexity of software components, involving several developers, and supported by productive technical and organisational settings. A disciplined development process is created where both products and processes are designed to facilitate human comprehension, effective collaboration, and technical flexibility.

The program explores the theoretical and conceptual foundation for object-oriented software development and develops specific frameworks and environments intended for industrial use or experiments in industrial settings. The effort is designed to foster close collaboration between software engineers, language and database designers, and tool constructors.

The goal of the program is to develop frameworks and environments that software organisations can use to respond effectively to a number of practical

problems. The concept of a software factory was developed in which the object-oriented paradigm is used to achieve: (i) sufficient technical flexibility to support integration, modification and maintenance of software; (ii) reuse of high-quality concepts and software components; (iii) construction and thorough testing of concepts and components; (iv) coherence in frameworks used in different life-cycle activities; and (v) a technical and organisational environment that supports a disciplined approach to software development.

The program is based on a five-year research collaboration: The Software Engineering Programme, 1990–1994. The results and experiences from this effort were used to identify essential state-of-the-art problems, and to lay the theoretical and technical foundation for the Software Factory.

7.2.5 The LUKAS Network

The Information Systems Group has played a leading role in establishing a local network, the LUKAS collaboration, between IT organizations in Northern Jutland and researchers in the Software Engineering Programme at the department.

The network activities to date have involved some 150 people from about 30 enterprises and institutions, and these activities have stimulated considerable collaboration with industry concerning research, education, and technology transfer.

The main focus of the network is on software process improvement, quality improvement, project management, database technology, and on object-oriented methods and technologies. The activities were carried out according to the EU Community Initiative concerning "Science and Technology for Regional Innovation and Development in Europe," STRIDE. The activities in the program were partially financed by the European Regional Development Fund and the European Social Fund.

7.3 Organization and Staff

The table below summarizes the staff during the evaluation period.

Teaching Positions					
	1991	1992	1993	1994	1995
Full Professors	1	1	1	1	1
Associate Professors	2	3	3	3	3
Assistant Professors	1				
Teaching Assistants		1	1	1	1
Research Positions					
Ph.D. Students	2	2 1	1 2	2 4	3 4

Lars Mathiassen

Academic Degrees

1981	Dr.scient. (Ph.D.) in Computer Science, University of Oslo
1975	Cand.scient. (M.S.) in Computer Science, Århus University

Positions

1993–	Head of the Doctoral School in Technology and Science, Faculty of Engineering and Science, Aalborg University
1990–	Visiting Professor, Institute Theseus, Sophia Antipolis, France
1987–	Full Professor, Department of Mathematics and Computer Science, Aalborg University
1975–82	Researcher and Associate Professor, Department of Computer Science, University of Århus

In 1991, Lars Mathiassen was awarded Teacher of the Year by the Faculty of Engineering and Science at Aalborg University (Prize: DKK 25,000). In 1994, Aalborg University promoted a small number of its approximately 60 full professors. Together with six colleagues, Lars Mathiassen was awarded an annual bonus (DKK 45,000) for his contributions to research, research education, and education in general.

Peter Axel Nielsen

Academic Degrees

1990	Ph.D. in Systems and Information Management, Lancaster University
1986	Cand.scient. (M.S.) in Computer Science, Århus University

Positions

1992–	Associate professor in Computer Science, Aalborg University, Institute of Electronic Systems
1992–93	Reseach fellow (part-time), Århus University, Department of Computer Science
1989–92	Assistant professor in Computer Science, Aalborg University, Institute of Electronic Systems
1986–89	Ph.D. Scholarship, Århus University, Faculty of Science

Jan Stage

Academic Degrees

1989	Dr.scient. (Ph.D.) in Computer Science, University of Oslo
1984	Cand.scient. (M.S.) in Computer Science, Aalborg University

Positions

1991–	Associate Professor in Computer Science, Aalborg University, Institute of Electronic Systems
1991	Visiting Assistant Professor in Information Systems, State University of New York at Binghamton, School of Management
1988–91	Assistant Professor in Computer Science, Aalborg University, Institute of Electronic Systems
1985–88	Ph.D. Scholarship, Århus University, Computer Science Department
1985	Research Assistant in Computer Science, Aalborg University, Institute of Electronic Systems

Ivan Aaen

Academic Degrees

1990	Lic.scient. (Ph.D.) in Computer Science, Aalborg University
1980	Cand.polyt. (M.E.) in Systems Construction, Aalborg University

Positions

1991–	Associate Professor in Computer Science, Aalborg University, Institute of Electronic Systems
1986–91	Assistant Professor in Computer Science, Aalborg University, Institute of Electronic Systems
1983–86	Ph.D. Scholarship, Århus University, Computer Science Department
1980–83	Research Assistant, Aalborg University, Institute of Production

7.4 Collaboration

The Information Systems Group has continued to emphasize research collaboration throughout the period 1990–1995. This is reflected on various levels and in relation to various groups:

- collaboration within the group,
- collaboration with colleagues from other groups within the department,
- collaboration with industry,
- collaboration with Danish research institutions, and
- international collaboration with institutions and individuals.

Below, we describe the group's collaboration with partners outside the department.

The group has collaborated with many Danish IT organizations. Research projects and consulting have been carried out together with Bankdata, Den Danske Bank, Kommunedata, Metodica, Nykredit, Systematic, and Aalborg Kongres- & Kulturcenter. The Information Systems Group continues to maintain and further develop its widespread and smoothly functioning network of practitioners with whom it establishes research and development collaborations.

Over the last couple of years, the group has developed close collaboration with the Systems Analysis Department, Risø National Laboratory, Denmark. The collaboration includes joint Ph.D. studies and it has resulted in a number of joint articles [23]. The focus of this collaboration is object-orientation and collaborative computing. The group has also collaborated with the Software Engineering Group, Aarhus University, Denmark. Peter Axel Nielsen has worked as visiting faculty there for one semester, resulting in a joint publication [27]. In addition, an international conference on "Computers in Context" was arranged in collaboration between the two groups and other researchers in August 1995.

The group has developed and further utilized its international network of individuals and groups with whom it collaborates. In continuation of professor Heinz Klein's stay at Aalborg University as visiting professor, Jan Stage was on a one-semester visit to the Information Systems Group, State University of New York, Binghamton, USA, during which he started a research collaboration with professor Richard Baskerville [46]. In addition, Ph.D. student John Venable has visited Aalborg University for a two-year period [42].

Lars Mathiassen has collaborated closely with professor Bo Dahlbom, Department of Informatics, Göteborg University, Sweden, resulting in several publications [2, 5, 6, 24]. During 1995, activities have been initiated to broaden this collaboration to include more members of the two research groups.

Ivan Aaen, Lars Mathiassen, and Ph.D. students Carsten Sørensen and Jan Damsgaard collaborated with professor Kalle Lyytinen and several of his colleagues from Department of Information and Computer Science, University of Jyväskylä, Finland. Students were exchanged and the collaboration has resulted in several publications [12, 13, 14, 22, 33, 50].

In 1994, collaboration was begun with the Software Engineering Institute (SEI) at Carnegie Mellon University, Pittsburgh, USA. Ph.D. student Peter Bøttcher visited SEI for half a year, Ivan Aaen was on a shorter visit there, and collaboration was begun with the Danish software company Kommunedata. This collaboration has, so far, resulted in one published article [11].

Ph.D. student Jan Damsgaard visited the Department of Information and Management Systems, Hong Kong University of Science and Technology, and he visited and initiated collaboration with professor John L. King, Department of Computer Science, University of California, Irvine, USA. Ph.D. student Birgitte Krogh visited and initiated a collaboration with Senior Researcher Eswaran Subrahmanian, Engineering Design Research Center, Carnegie Mellon University, Pittsburgh, USA. In addition, the Information Systems Group has developed a

close collaboration with the Software Engineering Group, Hamburg University, Germany, and with Institute Theseus, Sophia Antipolis, France.

7.5 Ph.D. Projects

During the period 1991–1995, the Information Systems Group supervised six Ph.D. students. Two of them finished successfully in 1993, one plans to finish early in 1996, and the remaining three started their study within the last year.

In addition to the projects described below, Lars Mathiassen served as the formal Ph.D. advisor for Christian S. Jensen who defended his degree in January 1991 and Lars Bækgaard who defended his degree in January 1993.

Process Quality and Product Quality as a Framework for Innovation Processes in Software Organizations

Name:	Peter Bøttcher
Education:	Cand.scient. (M.S.) in Computer Science and Mathematics, 1993
Duration:	April, 1994–March, 1997
Status:	In progress
Funding:	Danish Academy of Technical Sciences and Kommunedata I/S
Advisor:	Ivan Aaen

The focus areas of this Ph.D. project are software process improvement (SPI), Software Engineering Institute's Capability Maturity Model, and product quality.

The Capability Maturity Model provides a framework for assessing an organization's software processes and for planning software process improvements. As part of this Ph.D. project, supplementary techniques for tracking ongoing improvement efforts are offered. Such delta measurements offer a focused and operational supplement to the broad and strategic orientation of the Capability Maturity Model. In contrast to previous techniques that are based on questionnaire responses or voiced opinions on the practices performed, the delta measurements are based on assessments of products developed throughout the software development life cycle.

In conjunction with the development of the delta measurement method, a conceptual product quality model for products is developed. Finally, the relation between the delta measures and the product qualities will be formulated.

EDI Diffusion: A Three-layered Approach

Name:	Jan Damsgaard
Education:	Cand.scient. (M.S.) in Computer Science and Psychology, 1993
Duration:	August, 1993–July, 1996 (expected)
Status:	In progress
Funding:	Faculty of Engineering and Science, Aalborg University
Advisor:	Ivan Aaen

This Ph.D. project describes and evaluates patterns of EDI (Electronic Data Interchange) diffusion. A framework is constructed which takes into account institutional, industry-specific, and organizational factors in the study of EDI diffusion processes, and thus extends the analysis beyond organizational borders.

The research approach is both theoretical and empirical. The theoretical part consists of a literature study to build up a three-layer framework for examining existing literature. The empirical part consists of a number of field studies covering Hong Kong (longitudinal), Finland, and Denmark.

The dissertation proposes that a there is a need to orchestrate multi-level, multi-theory approaches to study the diffusion of complex and networked technologies such as EDI.

Computer Supported Cooperative Work—As Time Goes By

Name:	Steffen Herskind
Education:	Cand.scient. (M.S.) in Computer Science and Mathematics, 1994
Duration:	June, 1995–May, 1998 (expected)
Status:	In progress
Funding:	Danish Research Academy and Risø National Laboratory
Advisor:	Peter Axel Nielsen

Inside the computer-supported cooperative work community, much time and effort is spent on developing theories for supporting coordination of distributed, cooperative work by computers. These approaches deal with how to coordinate actors, tasks, roles, resources, activities, etc., but no attention is paid to coordination according to time.

Through a field study in distributed planning of truck transport empirical findings are collected on how a truck driver and a forwarding agent coordinate their work. The findings will be analysed, and their effort to coordinate according to "time categories" (such as cyclic time, sequential time, deadlines, etc.) is the main result.

The results from the field study will be used for conceptually designing a coordination mechanism which can be used in the transport sector and as basis

for theoretical considerations of computer-supported cooperative work according to time.

Object-Oriented Design of Computer Systems to Support Complex Cooperative Work Settings

Name:	Birgitte Krogh
Education:	Cand.scient. (M.S.) in Computer Science and Mathematics, 1993
Duration:	July, 1994–June, 1997 (expected)
Status:	In progress
Funding:	Institute of Electronic Systems, Aalborg University
Advisor:	Lars Mathiassen

This Ph.D. project seeks to examine the intersection of two fields of research: object-orientation and computer supported cooperative work (CSCW). The project takes the perspective that special attention should be paid to the question of applying object-oriented design methodologies to computer support for cooperative work settings. A survey and comparison is made of the perspectives on and requirements for systems design methodologies that are prevalent in current object-orientation and CSCW research, respectively. The project combines the two paradigms in a single design rationale that outlines strengths and weaknesses of existing techniques and principles with focus on a congested set of design requirements. Furthermore, additional and modified object-oriented techniques and principles will be suggested and tested, and the contributions will be placed in the broader context of theoretical and field studies on the topic.

Introducing Computer Aided Software Engineering (CASE) Tools into Software Organizations

Name:	Carsten Sørensen
Education:	Cand.scient. (M.S.) in Computer Science and Mathematics, 1989
Duration:	February, 1990–August, 1992
Status:	Degree awarded February, 1993
Funding:	Department of Computer Science, Århus University
Advisor:	Lars Mathiassen

This dissertation work investigates factors affecting the diffusion of CASE technology in software organizations, and is formed as an aggregation of four contributions. Contribution 1 surveys the other contributions and discusses the lessons learned in relation to state-of-the-art CASE research. Contribution 2 reports the results from a questionnaire survey of CASE experiences in 100 Danish and Finnish software organizations. Contribution 3 presents a theoretical line of argument for the possibilities and limitations for software organizations to adopt CASE technology. Contribution 4 takes a closer look at the factors affecting

CASE use in organizations by presenting a discrete-graph model of dependencies between a set of organizational diffusion factors.

One of the main results of this research was the investigation of the relationships between successful CASE implementation and organizational maturity with respect to software development. In particular, it was argued that, on the one hand, the technology is not sufficiently mature in terms of supporting needs, and on the other hand, organizations should be willing to put considerable effort into changing work practices as a result of introducing CASE technology.

CoCoA: A Conceptual Data Modeling Approach for Complex Problem Domains

Name:	John T. Venable
Education:	M.S. in Management Science, Binghamton University, 1983
	M.S. in Advanced Technology, Binghamton University, 1985
Duration:	1987–1994
Status:	Degree obtained
Advisor:	Lars Mathiassen and Jan Stage

This dissertation research describes and evaluates the suitability of a conceptual data model named CoCoA (for Complex Covering Aggregation) for the description and modeling of complex problem domains. The CoCoA Model is compared to a number of other conceptual data models and is also applied as an example to the meta-modeling and integration of various conceptual data models and data flow models. The dissertation additionally proposes a software architecture for information systems that support work in complex problem domains. The example covered is that of ICASE (Integrated Computer-Aided Software/System Engineering) environments. Standard mappings from CoCoA to the architecture are described.

7.6 Service and Research-related Activities

Scandinavian Journal of Information Systems

Since the Scandinavian Journal of Information Systems' establishment in 1989, the group has played a significant role in producing, editing, and improving it. The journal is published by the group and has currently about 300 individual and institutional subscribers, primarily in the Nordic countries with approximately 15% international subscribers. From 1991 to 1993 Lars Mathiassen was editor-in-chief, with Peter Axel Nielsen in charge of production and administration. Since 1993, Peter Axel Nielsen has been editor-in-chief and in charge of production, with Ivan Aaen in charge of administration.

Program Committees

- *The 7th Conference on Advanced Information Systems Engineering*, Jyväskylä, Finland, 1995 (L. Mathiassen)
- *3rd European Conference on Information Systems*, Athens, Greece, 1995 (L. Mathiassen)
- *International Conference on Information Systems*, Copenhagen, Denmark, 1990 (L. Mathiassen)
- *IFIP WG 8.2 Working Conference on The Impact of Computer Supported Technologies on Information Systems Development*, Minneapolis, Minnesota, June 1992 (I. Aaen)
- *IFIP WG 8.6 Working Conference on Diffusion and Adoption of Information Technology*, Oslo, Norway, 1995 (L. Mathiassen).
- *International Conference on Information Systems Development: Human, Social and Organizational Aspects*, Noordwijkerhout, Netherland, 1993 (L. Mathiassen)
- *International Conference on The Impact of Computer Supported Technologies on Information Systems Development*, Minneapolis, USA, 1992 (L. Mathiassen)
- *IRIS 18, Nordic Conference*, Gjern, August, 1995 (J. Stage)
- *International Conference on Computers in Context*, Aarhus, Denmark, 1995 (L. Mathiassen)

Organising Committees and Panels

Only organizing committees for international conferences and the most significant panels are listed.

- Organizing committee, *Computers in Context*, Aarhus, Denmark, August, 1995 (P. A. Nielsen)
- Panel participation, "Can North America Learn Anything from European Research?" *International Conference on Information Systems*, New York, USA, 1991 (L. Mathiassen)

Refereeing and Editorial Services

Apart from the last item, the services concern journals.

- *Accounting, Management, and Information Technologies*, 1991–95 (L. Mathiassen, referee)

- *Communications of the ACM*, 1991–95 (L. Mathiassen, referee)
- *European Journal of Information Systems* (L. Mathiassen, editorial board, 1991–95; I. Aaen, referee, 1992)
- *Information and Software Technology*, 1993 (I. Aaen, referee)
- *Information, Technology and People*, 1991–95 (L. Mathiassen, editorial board)
- *Journal of Information Systems*, 1991–95 (L. Mathiassen, referee).
- *MIS Quarterly*, 1991–95 (L. Mathiassen, referee)
- *Scandinavian Journal of Information Systems*, (L. Mathiassen, editor-in-chief, 1991–93; P. A. Nielsen, editor-in-chief, 1993–1995 and referee, 1991–95; J. Stage, referee, 1991–1995; I. Aaen, referee, 1991–95)
- *Computers in Context* (international conference), April-May 1995 (J. Stage, P. A. Nielsen, I. Aaen, referees)

Evaluation Committees

- Braa, Kristin, Ph.D., Department of Informatics, Oslo University, December 1995 (P. A. Nielsen)
- Grant, Delvin, Ph.D., State University of New York at Binghamton, School of Management, January 1991 (J. Stage)
- Kautz, Karlheinz, Dr. Philos., University of Oslo, Institute of Informatics, September 1993 (J. Stage)
- Malmborg, Lone, Ph.D., Copenhagen Business School, June 1995 (P. A. Nielsen)
- Adjunkt, Computer Science, Aalborg University, May 1995 (J. Stage)
- Adjunkt, Computer Science, Copenhagen Business School, March 1995 (P. A. Nielsen)
- Associate professor (1. amanuensis), System Development/HCI, University of Trondheim, September 1995 (J. Stage)
- Associate Professor, Informatics, Helsinki School of Economics, 1993 (L. Mathiassen)
- Professor, Computer Science, University of Trondheim, 1995 (L. Mathiassen)

The lists of services and research-related activities given above are not all-encompassing. However, they do include all important activities and illustrate the character of the group's engagement.

7.7 The Group's Own Evaluation

According to the plans made in January 1991, the group intended to:

1. Turn its focus and frame of reference to the international research community within both Information Systems and Software Engineering.

2. Publish research results internationally and in top-ranking journals.

3. Maintain an orientation towards both Information Systems and Software Engineering.

4. Take an experimental and empirical approach as the foundation of its research.

As mentioned in Section 7.1, the plans were to be carried out in two projects, TEQ and RSM, with the SE Programme as the sole source of financial support. The TEQ Project was to address the problem of introducing CASE tools into organisations of computing professionals. The purpose of the RSM Project was to develop a modern methodology for analysis and design of information systems. The TEQ Project with Ivan Aaen and Carsten Sørensen was basically conducted as planned. The project has been successful in its collaboration with a group of Finnish researchers from University of Jyväskylä. A joint article was written comparing the introduction of CASE in software development organisations. A number of articles were published in journals and at international conferences. These are now being acknowledged in much of the recent literature of CASE research, especially within the Nordic community. Carsten Sørensen's Ph.D. degree was based on some of these articles.

The RSM Project with Lars Mathiassen, Peter Axel Nielsen, and Jan Stage in collaboration with Andreas Munk-Madsen of the private company Metodica has developed a complete methodology for analysis and design of information systems based on object-oriented concepts and principles. The methodology is documented in two academic books currently used for teaching at several universities and computing colleges; and the adoption of the methodology in the computing industry is gradually taking off. In this respect, the project has been particularly satisfying and successful. However, the project did not proceed as originally planned, in all respects. First, object-orientation came to be the main theoretical underpinning of the methodology, whereas prototyping, risk management, and soft systems came to play a minor role. Consequently, the project changed its name to the OOA&D Project. Second, the empirical basis was originally planned to be experimentation with the methodology in the computing industry. This was changed to a research approach, where experience gained through teaching more and more elaborate versions of the methodology to practitioners and students formed the practical background for the design of the

methodology. While this is a perfectly useful approach, it does not carry the validity inherent in empirical research.

Other activities outside the projects were to build research networks, and maintain collaboration with researchers from other universities. From this stems significant publications, and important relationships have been established, developed, and utilised. Lars Mathiassen has written an acclaimed book on computing philosophy with Bo Dahlbom, Göteborg University. Jan Stage has worked with Richard Baskerville, State University of New York, on prototyping. Peter Axel Nielsen has worked with Morten Kyng, Århus University, on object-oriented modeling.

All in all, the past five years have brought substantial changes in the group's research orientation.

1. The group has successfully turned to the international research community when considering frame of reference, publication, and collaboration. It has, however, not produced results that are used by other international researchers to a satisfactory degree. The group's orientation towards information systems in general, and software engineering in particular, implies that the group is located between two major research fields. The growing concern for "soft" problems within software engineering research may provide new opportunities for obtaining higher international visibility for the group's research.

2. The results of the research are now to a considerably larger degree being published internationally. Although articles have been placed at important international conferences and journals, neither of the two projects have fulfilled the ambition to publish in top-ranking journals.

3. The group has utilised and further developed its orientation towards a synthesis of IS and SE to create a strong position for itself. The group has strengthened the focus on software engineering while the focus on informations systems has been somewhat weakened.

4. The research approach has to a sufficient degree taken an empirical orientation, although improvements still can be made in documenting the empirical basis of the results of the research.

The SE and SF Programmes have provided the necessary financial support for the TEQ and the OOA&D projects. The organisation in projects has proven useful for conducting the planned research, by facilitating internal cooperation without restraining the possibilities for collaboration with external researchers.

Within the Lukas framework, as well as within TEQ and OOA&D, the group has been active in spreading its research results to Danish industry. This has been rewarding, but also quite energy-consuming.

7.8 Research Plan 1996–2000

7.8.1 Topics

The group has developed a paradigm, called ITEM 2001, for its research over the next five years: *Information Technology Engineering & Management*. The thrust of the paradigm is a commitment to professionalize the IT field, covering systems development understood in its broadest sense, systems maintenance and use, information technology management, innovation, and utilisation in change processes. Intellectual support is sought in other disciplines such as management, systems, sociology, knowledge production, and technology in general.

The ITEM 2001 paradigm can be characterized along two dimensions. First, the research will address a variety of systems ranging from a narrow technical interest in software systems to a broader human-centered view of information systems. Second, the research will emphasize engineering as well as management activities related to the application of information technology in organizations. This approach provides the rationale for the ITEM name, and it expresses at the same time the underlying assumptions behind the paradigm: a professionalization strategy must (1) be based on an understanding of the fundamental technical systems involved, and it must emphasize the social and organizational context in which these are embedded; simultaneously, such a strategy must (2) strive to improve our constructive capacity to develop such systems as engineers together with a visionary and critical ability to plan and manage effective organizational interventions.

The topics in which the group will engage within ITEM 2001 fall into the following four categories.

- *IT usage and management:* business process reengineering, management of IT, IT infra-structure, electronic data interchange.
- *Professionalisation:* capability maturity models, management of IT organisations, quality management, computer-aided software engineering, process improvement.
- *Strategies for systems development:* object-oriented methodologies, soft systems methodologies, generic systems development, introduction of methodologies.
- *Models and design:* object-oriented models, quality of models, patterns and frameworks, design of artifacts.

ITEM 2001 is rooted in a tradition where conceptual development and documented empirical findings are important. Specific research initiatives may take the form of action research or interpretive studies, and they may result in theoretical contributions or in the development of methodologies and practical principles.

All empirical findings are to be related to a theory and any theory is to be related to empirical findings. Dissemination of research results plays an important role, not only after the research has been done but also as a way of testing results.

ITEM 2001 will serve as a basic paradigm for a variety of research activities in the group over the next five years. It will provide a shared orientation and will strengthen collaboration within the group. The paradigm is designed as a managerial research framework in which the group will seek to

- Maintain and develop its competence within software engineering while broadening its perspective to increase its focus on information technology.
- Supplement its traditional concern for engineering with a stronger interest in managerial activities related to the application of information technology.
- Establish a network of close collaborations with a number of IT professionals and organizations focusing on professionalization of the IT field. These collaborative activities will be the backbone in providing a stronger empirical orientation in the group's research activities.
- Maintain and further develop its close and productive interaction with a small number of international researchers and research groups.
- Maintain its tradition for developing useful material for IT professionals and at the same time giving priority to publication of results in first-class international journals and conferences.

The group will continue to finance its research activities through basic funding in combination with smaller dedicated applications. In parallel with this, the group will look for opportunities to participate in large-scale projects with industrial partners and academic partners from both within and outside the institute.

7.8.2 Staff

During the next five years, the Information Systems Group will continue to consist of a kernel four senior permanent staff members. In addition, the group plans to include at any point in time 1–2 assistant professors or visiting faculty. A group of two to five Ph.D. students will continue to play an important role as part of the group's research activities.

Bibliography

Academic Books

[1] Bang, Stig, Stig Efsen, Peter Hundborg, Henrik Janum, Lars Mathiassen & Christian Schultz (1991). *Quality Management in Systems Development* (In Danish: Kvalitetsstyring i Systemudvikling). Teknisk Forlag. 190 pages.

[2] Dahlbom, Bo & Lars Mathiassen (1993). *Computers in Context – The Philosophy and Practice of System Design.* Blackwell. 306 pages.

[3] Mathiassen, Lars, Andreas Munk-Madsen, Peter A. Nielsen & Jan Stage (1995). *Object-Oriented Design* (In Danish: Objektorienteret design). Aalborg: Marko. 312 pages.

[4] Mathiassen, Lars, Andreas Munk-Madsen, Peter A. Nielsen & Jan Stage (1993). *Object-Oriented Analysis* (In Danish: Objektorienteret analyse). Aalborg: Marko. 259 pages.

Articles in Scientific Journals

[5] Dahlbom, Bo & Lars Mathiassen (1994). A Scandinavian View on ACM's Code of Ethics. In *Computers and Society, ACM*, 24(2):14–22.

[6] Dahlbom, Bo & Lars Mathiassen (1992). Systems Development Philosophy. In *Computers and Society, ACM*, 22(1–4):12–23.

[7] Mathiassen, Lars, Thomas Seewaldt & Jan Stage (1995). Prototyping and Specifying: Principles and Practices of a Mixed Approach. In *Scandinavian Journal of Information Systems*, 7(1):55–72.

[8] Mathiassen, Lars & Jan Stage (1992). The Principle of Limited Reduction in Software Design. In *Information, Technology, and People*, 6(2–3):171–185.

[9] Aaen, Ivan (1994). Problems in CASE Introduction: Experiences from User Organizations. In *Information and Software Technology*, 36(11):643–654.

[10] Aaen, Ivan & Carsten Sørensen (1991). A CASE of Great Expectations. In *Scandinavian Journal of Information Systems*, 3(1):3–23.

Articles in Proceedings of International Conferences

[11] Bøttcher, Peter, Robert Stoddard & David Zubrow (1995). A Baysian Approach for Modeling Quality in Software Products and Processes. In *Proceedings of the Fifth International Conference on Software Quality*, Austin, Texas.

[12] Damsgaard, Jan & Kalle Lyytinen (1994). Can the Dragon jump on the EDI bandwagon? In *Proceedings of the second SISNet conference*, Barcelona, Spain.

[13] Damsgaard, Jan, André Rogaczewski & Kalle Lyytinen (1993). How Information Technologies Penetrate Organizations: An Analysis of Four Alternative Models. In *Proceedings of the IFIP TC8 Working Conference on Diffusion, Transfer and Implementation of Information Technology (IFIP)*, Pittsburgh, Pennsylvania. Amsterdam: North-Holland.

[14] Lyytinen, Kalle, Lars Mathiassen & Janne Ropponen (1995). A Behavioral Model of Software Risk Management. In G. Doukidis *et al.* (Eds.), *Proceedings of the Third European Conference on Information Systems*, pp. 261–276. Athens University of Economics and Business: Department of Informatics.

[15] Mathiassen, Lars, Andreas Munk-Madsen, Peter A. Nielsen & Jan Stage (1994). Modelling Events in Object-Oriented Analysis. In D. Patel *et al.* (Eds.), *Proceedings of the International Conference on Object-Oriented Information Systems,* pp. 88–104. Berlin: Springer-Verlag.

[16] Mathiassen, Lars, Andreas Munk-Madsen, Peter A. Nielsen & Jan Stage (1994). Combining two Approaches to Object-Oriented Analysis. In E. Bertino *et al.* (Eds.), *Proceedings of the International Symposium on Object-Oriented Methodologies and Systems,* pp. 158–170. Berlin: Springer-Verlag.

[17] Mathiassen, Lars, Andreas Munk-Madsen, Peter A. Nielsen & Jan Stage (1991). Soft Systems in Software Design. In M. C. Jackson *et al.* (Eds.), *Systems Thinking in Europe,* pp. 311–317. New York: Plenum Press.

[18] Mathiassen, Lars & Carsten Sørensen (1994). Managing CASE Introduction – Beyond Software Process Maturity. In J. W. Ross (Ed.), *Proceedings of the ACM SIGCPR Conference*, pp. 242–251. ACM Press.

[19] Stage, Jan (1991). The Use of Descriptions in Analysis and Design of Information Systems. In R. K. Stamper *et al.* (Eds.), *Collaborative Work, Social Communications, and Information Systems,* pp. 237–260. Amsterdam: North-Holland.

[20] Aaen, Ivan (1993). CASE User Satisfaction – Impact Evaluations in User Organizations. In H.-Y. Lee *et al.* (Eds.), *Proceedings of CASE '93 – Sixth*

International Workshop on Computer-Aided Software Engineering, pp. 66–74. IEEE Computer Society.

[21] Aaen, Ivan (1991). CASE Tools Supporting Software Engineering Education. In V.-P. Tahvalainen *et al.* (Eds.), *Proceedings of the Second Workshop on The Next Generation of CASE Tools*, pp. 333–344. University of Jyväskylä.

[22] Aaen, Ivan, Aila Siltanen, Carsten Sørensen, Veli-Pekka Tahvanainen (1992). A Tale of Two Countries – CASE Experiences and Expectations. In K. E. Kendall *et al.* (Eds.), *Proceedings of IFIP WG 8.2 Conference*, Minnesota, USA.

Articles in Proceedings of Nordic Conferences

[23] Carstensen, Peter H., Birgitte Krogh & Carsten Sørensen (1995). Object-Oriented Modeling of Coordination Mechanisms. In B. Dahlbom *et al.* (Eds.), *Proceedings of the Eighteenth Information Systems Research Seminar in Scandinavia*, pp. 113–129. University of Gothenburg.

[24] Dahlbom, Bo & Lars Mathiassen (1994). The Future of Computing. In P. Kerola *et al.* (Eds.), *Proceedings of the Seventeenth Information Systems Research Seminar in Scandinavia*, pp. 2–16. University of Oulu.

[25] Damsgaard, Jan (1994). How EDI Reorganizes Organizations: A TCA view. In P. Kerola *et al.* (Eds.), *Proceedings of the Seventeenth Information Systems Research Seminar in Scandinavia*, pp. 593–605. University of Oulu.

[26] Krogh, Birgitte (1994). Object Orientation and Articulation Work. In P. Kerola *et al.* (Eds.), *Proceedings of the Seventeenth Information Systems Research Seminar in Scandinavia*, pp. 593–605. University of Oulu.

[27] Kyng, Morten & Peter A. Nielsen (1993). Domain Modelling and Application Frameworks. In J. Bansler *et al.* (Eds.), *Proceedings of the Sixteenth Information Systems Research Seminar In Scandinavia*, pp. 221–235. University of Copenhagen: Department of Computer Science.

[28] Mathiassen, Lars, Andreas Munk-Madsen, Peter A. Nielsen & Jan Stage (1992). Modelling Events in Object-Oriented Analysis. In G. Bjerknes *et al.* (Eds.), *Proceedings of the Fifteenth Information Systems Research Seminar in Scandinavia,* Part II, pp. 742–757. University of Oslo: Department of Informatics.

[29] Mathiassen, Lars, Andreas Munk-Madsen, Peter A. Nielsen & Jan Stage (1991). Rapid Systems Modelling: The Soul of a New Methodology. In O. Forsgren *et al.* (Eds.), *Report of the Fourteenth Information Systems Research Seminar in Scandinavia,* Part II, pp. 742–757. University of Umeå.

[30] Mathiassen, Lars & Peter A. Nielsen (1994). Interaction and Transformation in Soft Systems Methodology. In P. Kerola *et al.* (Eds.), *Proceedings of the Seventeenth Information Systems Research Seminar in Scandinavia*, pp. 290–302. University of Oulu.

[31] Mathiassen, Lars & Carsten Sørensen (1995). The Why, What, Who, Where, and How of CASE Management. In B. Dahlbom *et al.* (Eds.), *Proceedings of the Eighteenth Information Systems Research Seminar in Scandinavia*, pp. 479–492. Göteborg University.

[32] Peter A. Nielsen & Jan Stage (1994). The Software Designer: Skills and Change. In P. Kerola *et al.* (Eds.), *Report of the Seventeenth Information Systems Research Seminar in Scandinavia,* Part II, pp. 799–813. University of Oulu: Department of Information Processing Science.

[33] Ropponen, Janne, Lars Mathiassen & Kalle Lyytinen (1992). Risk Management in Information Systems Development. In G. Bjerknes *et al.* (Eds.), *Report of the Fifteenth Information Systems Research Seminar in Scandinavia,* Part I, pp. 325–344. University of Oslo: Department of Informatics.

[34] Aaen, Ivan (1994). CASE as a Maturity Lever – Institutionalization via CASE. In P. Kerola *et al.* (Eds.), *Report of the Seventeenth Information Systems Research Seminar in Scandinavia,* Part II, pp. 705–714. University of Oulu: Department of Information Processing Science.

[35] Aaen, Ivan (1993). Problems in CASE Introduction – Experiences from User Organizations. In J. Bansler *et al.* (Eds.), *Proceedings of the Sixteenth Information Systems Research Seminar In Scandinavia*, pp. 550–569. University of Copenhagen: Department of Computer Science.

[36] Aaen, Ivan (1992). CASE User Satisfaction – Impact Evaluations in User Organizations. In G. Bjerknes *et al.* (Eds.), *Report of the Fifteenth Information Systems Research Seminar in Scandinavia,* Part I, pp. 412–424. University of Oslo: Department of Informatics.

[37] Aaen, Ivan, Peter Bøtcher & Lars Mathiassen (1995). Delta Measurement in Software Process Improvement based on Product Attributes. In B. Dahlbom *et al.* (Eds.), *Proceedings of the Eighteenth Information Systems Research Seminar in Scandinavia*, pp. 15–28. Göteborg University.

Book Contributions

[38] Mathiassen, Lars (1995). Control and Creativity in Computing Practices. In S. Hildebrandt *et al.* (Eds.), *Management II. 10 Danish Professors' Views on Contemporary Management* (In Danish: Ledelse II. 10 danske professorers bud på aktuel ledelse), pp. 127–156. København: Børsen.

[39] Mathiassen, Lars, Thomas Seewaldt & Jan Stage (1994). Prototyping and Specifying: Principles and Practices of a Mixed Approach. In J. Stage *et al.* (Eds.), *Quality Software: Concepts and Tools,* pp. 19–40. Aalborg University, Institute of Electronic Systems.

[40] Aaen, Ivan (1992). CASE Tool Bootstrapping – How little Strokes fell great oaks. In K. Lyytinen *et al.* (Eds.), *Next Generation CASE Tools*, pp. 8–17. IOS Press.

[41] Aaen, Ivan and Carsten Sørensen (1994). A CASE of Great Expectations. Quality Software-Concepts and Tools. In J. Stage *et al.* (Eds.), *Quality Software: Concepts and Tools,*. Aalborg University, Institute of Electronic Systems.

Ph.D. Theses

[42] Venable, John R. (1993). *CoCoA: A Conceptual Modelling Approach for Complex Problem Domains.* State University of New York at Binghamton: Thomas J. Watson School of Engineering and Applied Science.

[43] Sørensen, Carsten (1993). *Introducing CASE Tools into Software Organizations.* Aalborg University, Institute of Electronic Systems.

Books Edited

[44] Stage, Jan, Kurt Nørmark & Kim G. Larsen, Editors (1994). *Quality Software: Concepts and Tools.* Aalborg University, Institute of Electronic Systems.

[45] Dahlbom, Bo, Frederik Kämmerer, Fredrik Ljungberg, Jan Stage & Carsten Sørensen, Editors (1995). *Proceedings of the Eighteenth Information Systems Research Seminar in Scandinavia.* Göteborg University, Department of Informatics.

Other Publications

[46] Baskerville, Richard L. & Jan Stage (1991). *Developing the Prototype Approach in Rapid Systems Modelling.* Aalborg University, Institute of Electronic Systems, R 91–35.

[47] Damsgaard, Jan & Kalle Lyytinen (1994). *Government intervention in the Diffusion of EDI.* Working Paper WP-31, University of Jyväskylä.

[48] Damsgaard, Jan (1995). *Brølet fra Øst.* Aalborg University, Institute of Electronic Systems, R 95–2001. (In Danish.)

[49] Lyytinen, Kalle, Lars Mathiassen & Janne Ropponen (1993). *Software Risk Management as Satisficing Behavior: an Environmental Model.* University of Jyväskylä: Department of Computer Science and Information Systems, WP-27.

[50] Aaen, Ivan and Carsten Sørensen (1991). *CASE in Use: Results from the CASE Monitor Project.* Aalborg University, Institute of Electronic Systems.

Chapter 8

Database Systems

8.1 Profile

The importance of information in most organizations, and thus the value of databases, has resulted in the development of a large body of concepts, theories, and systems for the effective and efficient management of databases. The research area of the Database Systems Group is database management technology—technology aimed at meeting the needs for effective and efficient retrieval and storage of information in large databases.

This area is currently witnessing a rapid growth in several respects. Increasingly larger amounts of data are being managed using database management technology. The types of data being managed are moving beyond traditional business data to include, e.g., temporal, spatial, time-series, and multimedia data. New technologies, most prominently object-oriented and object-relational, are entering the market place, which until recently has been dominated by relational technology. Last but not least, the number of types of applications that attempt to utilize database technology is growing.

The research approach in the group has a technological focus and is primarily constructive in its outset, but also integrates experimental and analytical elements. Constructive activities include the design of concepts and frameworks, as well as the design and implementation of algorithms, languages, and systems. Experimental activities include the testing of the results of the constructive activities, including both prototype-based experiments and simulations. Analytical activities include complexity analysis and language evaluation. the emphasis is on the development of theoretically sound results that solve actual real-world problems.

Database management technology is related to a variety of other areas, such as proof systems, knowledge bases, expert systems, information systems, and programming languages that can exploit the potential of database management technology. The group emphasizes activities that reach out to these areas, par-

ticularly when they are represented within the department.

Current research interests in the group span all aspects of temporal databases, including conceptual modeling and database design, data models, query processing, storage management, indexing, and applications. Also included are data warehousing, data mining, database semantics, general query optimization, integrity constraints, deductive databases, and temporal reasoning.

At the start of the evaluation period, the group included one assistant professor and one Ph.D. student, and these goals were chosen for the period: more faculty members and a number of M.S. and Ph.D. students in the group, a database systems laboratory, and the definition of an "umbrella" project that was to integrate activities in the group and thus ensure that the activities would benefit from each other. No database courses were at that time included in the computer science curriculum.

8.2 Activities and Results

The IM/T Project, 1991–1993

Participants included C. E. Dyreson (James Cook University), S. Hsu (University of Arizona), C. S. Jensen (principal investigator), R. T. Snodgrass (University of Arizona), M. D. Soo (University of South Florida), L. Mark (Georgia Institute of Technology), and N. Roussopoulos (University of Maryland). The project was funded by the Danish Natural Science Research Council and Christian and Otilia Brorsons Mindelegat.

The focus of this project was on efficient support for transaction time and valid time in database systems. This challenging problem was approached from several directions. First, the efficient query processing framework developed during the earlier ERA project [9, 10] was investigated further, with a focus on the integration of valid and transaction time as well as on incremental join processing [8, 33]. Second, application-specific interrelations between valid and transaction time were explored, along with the possibilities for exploiting them to improve performance [7, 19]. Third, temporal generalizations of conventional joins were defined, and the efficient implementation of the central valid-time natural join studied [15, 63]. Fourth, timestamp formats were developed that are superior to those of SQL–92. Finally, related issues were studied, such as the semantics of time-varying attributes in temporal relations.

The TSQL2 Design Effort, 1992–1995

The participants in this effort included I. Ahn (Samsung Industries), G. Ariav (Tel Aviv University), D. Batory (University of Texas), J. Clifford (New York University), C. E. Dyreson (James Cook University), R. Elmasri (University of

Texas), Fabio Grandi (University of Bologna), C. S. Jensen, N. Kline (Microsoft), K. Kulkarni (Tandem Computers), W. Käfer (Daimler Benz), T. Y. C. Leung (IBM Santa Teresa), N. A. Lorentzos (Agricultural University of Athens), J. F. Roddick (University of South Australia), A. Segev (University of California and Lawrence Berkeley Lab), R. T. Snodgrass (University of Arizona), M. D. Soo (University of South Florida), and S. M. Sripada (ECRC).

One goal of this effort was to consolidate previous research contributions and achieve a consensus temporal query language on which future research in temporal databases could be based. Another goal was to provide industry with a single, fully specified design for a temporal query language that the research community recommends for integration into commercial products. Yet another goal was to provide recommendations and guidelines for the standards bodies.

The results of the project are documented in a 68-page language specification extension to the SQL–92 standard [32, 41] and in a series of technical commentaries that cover the main aspects of the language design. This documentation has recently been published in a book.

With various collaborators, most notably R. T. Snodgrass, C. S. Jensen has designed core aspects of the TSQL2 language, including the data model [6, 17, 22, 23]; the select (valid-time projection), the from, and the where (valid-time selection) clauses [24, 25]; update [26]; database variables [28, 62]; surrogates [21]; and such aspects as programming language interaction [27], an underlying algebra [31], and an architectural framework [30]; etc. [16, 20, 38, 60].

The relevant ISO and ANSI committees are currently adding constructs and concepts developed during this effort to the part SQL/Temporal of the SQL3 standard [35].

Consolidation of TSQL2, 1994–1995

The participants included M. Böhlen, C. S. Jensen (principal investigator), L. Mark (Georgia Institute of Technology), R. T. Snodgrass (University of Arizona), and M. D. Soo (University of South Florida). The project was funded by the Danish Natural Science Research Council.

Wide acceptance of the consensus temporal query language TSQL2 is likely, in part because its design committee has wide, indeed global, participation of leading temporal database researchers. But while TSQL2 is the most comprehensive temporal query language proposal to date, several aspects of the language design still remain to be consolidated to unlock its full potential for practical applicability.

Although the design of relational database schemas is covered in numerous textbooks, the design of temporal relational database schemas has been treated only sporadically. The project generalized conventional normal forms and proposed new ones based on temporal attribute semantics [51, 11, 51]. In temporal databases, logical deletions transform into insertions at the physical level, thus

creating a need for novel physical-deletion facilities, not found in non-temporal data models. The project designed TSQL2-language facilities for physical deletion and developed correctness criteria for deletions [29]. A comprehensive foundation for the use of database variables in query languages was also provided [52]. It is a challenging task to provide an efficient implementation of TSQL2 that attempts to exploit the temporal dimensions of data. The project consolidated the design of an appropriate algebra for TSQL2. Efficient implementation was considered for the timeslice and join operators [50]. Finally, the project provided an analysis and evaluation of the TSQL2 language design [46, 13, 14], which in turn led to proposals for improving aspects of the design .

The MultiCal Project, 1991–

Participants in the project include R. T. Snodgrass (principal investigator, University of Arizona), C. E. Dyreson (James Cook University), C. S. Jensen (Aalborg University), N. Kline (Microsoft) M. D. Soo (University of South Florida), L. So, and J. Whelan (University of Arizona). The objective of the MULTICAL Project was the development of the MULTICAL DBMS.

MultiCal is a relational DBMS that supports the internationalization of time constants. The system augments SQL2 with calendar-independent constructs that allow for the convenient use of multiple calendars in a database context. The system provides limited extensibility, in that users may add new calendars (in multiple natural languages and using multiple input and output formats) to the system [64].

Release 1.0 consists of about 48K source lines of C code and uses a query processor consisting of 63K lines of C code. Orthogonal functionality, such as multi-user support, is not included. Release 1.0 was made available in October 1993 as public domain software running on Sun-4 machines under SunOS. Release 1.1 that followed in September 1995 added new functionality, including, e.g., a calendric systems generator and support for DEC Alpha machines. A 330-page manual accompanies the system [72].

The Consensus Temporal Database Glossary Initiative, 1992–

A common technical language is an important infra-structural component of any scientific community. To be effective, such a language should be well-defined, intuitive, and agreed-upon. The purpose of this on-going initiative is to establish and maintain important aspects of such a language. The initiative meets a need for creating a higher degree of consensus on the definition and naming of concepts central to temporal databases. The use of inconsistent terminology adversely affects the accessibility of the literature—to members of the community as well as others—and has an adverse effect on progress.

The initiative is coordinated by C. S. Jensen, who has also headed two editorial boards during the initiative. Co-editors include J. Clifford (New York University), R. Elmasri (University of Texas), S. K. Gadia (Iowa State University), P. Hayes (Beckman Institute), S. Jajodia (George Mason University), A. Segev (University of California), R. T. Snodgrass (University of Arizona), and A. Tansel (City University of New York). In excess of 25 researchers are contributing.

An initial glossary of temporal database concepts arose from e-mail discussions when appropriate terminology was considered for the first book on temporal databases [34]. That glossary also appeared in the September 1992 issue of the ACM SIGMOD Record [44]. The effort continued, independently of the book, and status documents appeared in December 1992, in March 1993, and in May 1993 [42]. Following discussions at the "ARPA/NSF International Workshop on an Infrastructure for Temporal Databases," a revision of the glossary was released [40]. Work on a new release is scheduled to start in 1996.

The Consensus Temporal Database Benchmark Initiative, 1993–

This initiative, coordinated by C. S. Jensen, currently includes 22 contributing researchers. The central goal of this initiative is to provide the temporal database community with an extensive consensus test suite for temporal relational query languages that is independent of any existing language proposal.

Two test suites have been released, one at the at the ARPA/NSF International Workshop on an Infrastructure for Temporal Databases, held in Arlington [43], and one following that workshop [56].

The TempIS Project, 1982–

The participants of this project include, among others, R. T. Snodgrass (principal investigator, University of Arizona), C. E. Dyreson (James Cook University), C. S. Jensen, M. D. Soo (University of South Florida), N. Kline (Microsoft), and S. Hsu (University of Arizona).

Over the years, many researchers have pioneered numerous aspects of temporal database system technologies within the framework of this project. This is undoubtedly the widest-ranging and most productive temporal database research project. The project has been supported by National Science Foundation (three grants), IBM (two grants), NCR Copenhagen, and the Office of Naval Research. Results are documented in more than 50 TempIS technical reports and several major database system prototypes, in addition to many conference and journal papers.

TimeCenter, 1995–2000

TIMECENTER is an international center for the support of temporal database applications on traditional and emerging DBMS technologies. Current participants include M. Böhlen, H. Gregersen, C. S. Jensen (co-director), and K. Torp (Aalborg University); R. T. Snodgrass (co-director, University of Arizona); C. E. Dyreson (James Cook University, Australia); M. D. Soo (University of South Florida); and A. Steiner (ETH Zurich). The center was created only a few months ago, and more participants are expected to join. Also, funding is currently being sought from Danish and international sources. Below is a brief mission statement for the center; additional information is provided in Section 8.8.3.

Recent advances in temporal query languages and data models clearly demonstrate that database applications that manage time-varying data—and most do—may benefit substantially from built-in temporal support in the database technology used. In spite of this, and although temporal databases have been an active area of research for fifteen years, temporal database technology has so far had little impact on practice. It is the goal of TIMECENTER to improve this state of affairs.

While others are pursuing valuable organizational solutions to the general problem of effective technology transfer, TIMECENTER focuses on technical solutions and aims at developing temporal database technology that is expressly *transferable*. Such technology allows for the continued operation of legacy code, for the harmonious coexistence of legacy and new application code, and for the reuse of programmer expertise and knowledge. It also allows for maximal, effective, and efficient reuse of the functionality already provided by current database management systems. This leads to a technology that is attractive to use and manageable to implement.

TIMECENTER aims at exploring the theoretical foundation for developing such a technology and at actually developing it. The TIMECENTER architecture, in addition to being transferable, is *migratable*, i.e., it is open for the gradual integration of high-performance, special-purpose temporal query processing modules.

At the time of writing, prototypes of two software components have been constructed, both of which are publicly available. TIMEIT is an integrated testbed for testing temporal modules before they are integrated into the TIMECENTER architecture. TIMEDB is a preprocessor for Oracle that accepts TSQL2 queries and generates SQL code to evaluate these on Oracle.

The Software Engineering Programme, 1990–1994

This project, funded by the Danish Natural Science Research Council as a framework program (in Danish, "rammeprogram"), was a collaborative effort between three research groups in the unit. The Database Systems Group was part of the project during its second period, 1993–1994. A full description is offered in

the section of this report devoted to the Information Systems Group (see Section 7.2.1).

The Software Factory, 1995–1997

This project, also funded by Danish Natural Science Research Council, represents a continued collaboration between the Database Systems, Information Systems, and Programming Systems Groups, and it may be seen as the successor to the Software Engineering Programme, just described. Again, the description may be found elsewhere (Section 7.2.4).

Event Modeling

Lars Bækgaard collaborates with Leo Mark (Georgia Tech), and Jens Christian Godskesen (Tele Danmark Research).

The purpose of this activity is to design languages for the specification of dynamic database constraints and to develop methods for the efficient monitoring of dynamic database constraints. A database can be seen as a model of some universe of discourse (UoD). This implies that a database should be a model of both static and dynamic UoD aspects. As a result of our activity, various language constructs for the specification of transaction-based, dynamic constraints have been designed. A language based on regular expressions combined with transaction preconditions for relational databases has been described in [66]. A similar language for object-based databases has been described in [57]. A CCS-like language have been described in [48].

Intelligent Query/Answer Systems

Participants: Henrik L. Larsen (Roskilde University, principal investigator), Lars Bækgaard, and Troels Andreasen (Roskilde University). The project has received support from the Danish National Research Councils, DKK 150,000 in 1994 and DKK 250,000 in 1995.

The purpose of the project is to develop and experiment with techniques that can assist users in their quest for information. Contemporary systems create query answers in very rigid ways. In many situations users know what they want, but do not know how to formulate the queries that will extract the desired information. The following techniques illustrate the focus of the project: Query generalization, query relaxation, fuzzy logic, similarity measures/concepts, data mining. The project has been a creative environment for mutual discussions and for the creation of new ideas and results.

The development of the document retrieval prototype CoopBase [53, 37, 12] has been supported by the intelligent query/answer systems project and the Software Engineering Programme. CoopBase utilizes knowledge from historical query

sessions to create answers to new queries by maintaining a dynamic network of similarity measures between keywords and documents. The similarity measures are maintained by analysing keywords and answer documents in query sessions. The established measures are then used to create fuzzy answers to new queries. Fuzzy answers from CoopBase are presented in a ranked way based on a fuzzy aggregation of the similarity measures. CoopBase has made it possible to qualify the idea of behavior mining in query/answer systems.

Incremental Computation of Complex Queries

Lars Bækgaard is the principal investigator. Leo Mark (Georgia Tech) and Nick Roussopoulos (University of Maryland) are collaborators.

Incremental computation is a cost-efficient alternative to recomputation when the same query is computed repeatedly and if the intermediate database changes are relatively few. In situations where views must be maintained in physically distributed environment there is no alternative to incremental computation. Incremental techniques for time-varying queries [5], nested queries [4], and set difference queries [54] have been created. These results significantly extends the class of queries that can be computed incrementally. During the period October 1995 through January 1996 Lars Bækgaard visited Nick Roussopoulos, University of Maryland, and Leo Mark, Georgia Tech in order to analyze the use of incremental techniques in data warehousing environments.

The Danish Database Workshops

Lars Bækgaard and Christian S. Jensen have been active participants at the four workshops that have been held during the period. The workshops have served as an intimate environment for discussions and presentations among about 10 danish researchers. Lars Bækgaard edited the abstracts and papers from three of the workshops [61, 58, 49].

8.3 Organization and Staff

The table below summarizes the staff during the evaluation period.

Teaching Positions					
	1991	1992	1993	1994	1995
Associate Professors				1	1
Assistant Professors	1	1 2	2	1	1 2
Teaching Assistants				1	1
Research Positions					
Visiting Professors					2
Ph.D. Students	1	1			1

In Fall 1995, L. Mark and R. T. Snodgrass had each a one-month position as a visiting professor. It follows from the figures in the table that the group had at its disposition only approximately six person years for research in the evaluation period.

Lars Bækgaard

Academic Degrees

1993	Ph.D. in Computer Science, Aalborg University
1988	Cand.scient. (M.S.) in Computer Science, Aalborg University

Positions

1995–96	Visitor, University of Maryland, Department of Computer Science
1992–	Assistant Professor, Aalborg University, Department of Mathematics and Computer Science
1990–92	Ph.D. Scholarship, Aalborg University, Department of Mathematics and Computer Science, stationed at University of Maryland, Department of Computer Science
1984–86	Software Developer (part time), Kommunedata, Aalborg
1980–82	Software Developer, Kommunedata, Aalborg

Michael Böhlen

Academic Degrees

1994	Dr. sc. tech. (Ph.D.), ETH Zürich, Department of Computer Science
1990	Dipl. Informatik-Ing. ETH (M.S.), ETH Zürich, Department of Computer Science

Positions

1995–	Assistant professor, Aalborg University, Department of Mathematics and Computer Science
1994–95	Post Doctoral Scholarship, University of Arizona, Department of Computer Science
1990–94	Ph.D. Scholarship (Assistent), ETH Zürich, Department of Computer Science
1990–94	Lecturer, IGS, Grenchen, Switzerland (IGS stands for Ingenieurschule Grenchen-Solothurn)

Heidi Gregersen

Academic Degrees

1995	Cand.scient. (M.S.) in Computer Science, Aalborg University

Positions

1995–	Teaching Assistant, Aalborg University, Department of Mathematics and Computer Science

Christian S. Jensen

Academic Degrees

1991	Ph.D. in Computer Science, Aalborg University
1988	Cand.scient. (M.S.) in Computer Science, Aalborg University

Positions

1994–	Associate Professor, Aalborg University, Department of Mathematics and Computer Science
1994–95	Visiting Associate Professor, University of Arizona, Department of Computer Science
1990–94	Assistant Professor, Aalborg University, Department of Mathematics and Computer Science
1991, 1992	Visiting Scholar, University of Arizona, Department of Computer Science
1988–90	Ph.D. Scholarship, Aalborg University, Department of Mathematics and Computer Science, stationed at University of Maryland, Department of Computer Science

Kristian Torp

Academic Degrees

1994	Cand.scient. (M.E.) in Computer Science, Aalborg University

Positions

1995–	Research Scholar, Aalborg University, Department of Mathematics and Computer Science
1994–95	Software Developer, Stibo Datagraphics, inc., Aarhus, Denmark

8.4 Collaboration

Richard T. Snodgrass, University of Arizona R. T. Snodgrass was a main collaborator for M. Böhlen during a one-year post doctoral stay at the University of Arizona. They have collaborated on the consolidation of TSQL2 and in TIMECENTER, and their joint work includes temporal database topics such as integrity constraints, TSQL2, and completeness. During the period, R. T. Snodgrass and C. S. Jensen have collaborated very closely on numerous projects and topics. Many sections of this chapter illustrate aspects of this collaboration.

Leo Mark, Georgia Tech, USA Leo Mark was Ph.D. advisor for both C. S. Jensen and L. Bækgaard. As such he has played a central role in the establishment of the Database Systems Group. L. Bækgaard and C. S. Jensen have visited him several times at Georgia Tech during the period, and he visited our Department as a visiting professor. The collaboration has resulted in several publications [4, 5, 54, 8, 9, 10].

Curtis E. Dyreson, James Cook University; Nick Kline, Microsoft; Michael D. Soo, University of South Florida At the beginning of the period, C. E. Dyreson N. Kline, and M. D. Soo were Ph.D. students at University of Arizona. C. E. Dyreson collaborated with C. S. Jensen in the IM/T project, the TSQL2 Design Effort, the MultiCal project, the TempIS project, the glossary and benchmark initiatives, and TIMECENTER. In addition to these, M. D. Soo also participated in Consolidation of TSQL2. N. Kline participated with C. S. Jensen on the TSQL2 Design Effort, the MultiCal project, the TempIS project, and the benchmark and glossary initiatives.

James Clifford and Tommy Isakowitz, University of New York J. Clifford and T. Isakowitz collaborated with C. S. Jensen in the TSQL2 Design Effort and Consolidation of TSQL2. In addition, J. Clifford collaborated with C. S. Jensen as a co-author and co-editor on the glossary initiative and as the PC chair

and a co-steering committee member for the 1995 International Workshop on Temporal Databases.

Arie Segev, University of California at Berkeley and Lawrence Berkeley Laboratory A. Segev collaborated with C. S. Jensen on the TSQL2 Design Effort, the glossary initiative, and as the steering committee chair for the 1995 International Workshop on Temporal Databases, where they co-authored the workshop report.

Andreas Steiner, ETH Zürich A. Steiner collaborates with M. Böhlen and C. S. Jensen on TIMECENTER. He has implemented major parts of TIMEDB while a teaching assistant at ETH Zürich.

Robert Marti, ETH Zürich R. Marti was the Ph.D. advisor for M. Böhlen. As such, he contributed to definition and development of ChronoLog.

Jan Chomicki, Monmoth College J. Chomicki collaborates with M. Böhlen on studies of the expressive power of query languages. This includes studies of the mapping between temporal logic and TSQL2.

Nick Roussopoulos, University of Maryland, USA Nick Roussopoulos collaborated with C. S. Jensen in the IM/T Project. At the end of 1995. L. Bækgaard visited Nick Roussopoulos in order to initiate a collaboration on incremental computation methods in data warehousing environments. Preliminary results are described in [47].

Jens C. Godskesen, Tele Danmark Research Jens Christian Godskesen collaborates with L. Bækgaard on formal specification and verification methods for dynamic database constraints [48]. The collaboration took place while Jens Christian Godskesen was on leave from Tele Danmark Research.

Troels Andreasen, Henrik L. Larsen, and Dan Rasmussen, Roskilde University These researchers collaborated with L. Bækgaard on the project "ntelligent Query/Answer Systems." See the project description in Section 8.2.

John F. Roddick, University of South Australia; Wolfgang Kæfer, Daimler Benz; Fabio Grandi, University of Bologna; T. Y. C. Leung, IBM Santa Theresa; Suchen Hsu, University of Arizona These researchers have collaborated with C. S. Jensen in the TSQL2 Design Effort.

In addition to the above researchers, C. S. Jensen has collaborated less intensively with a number of other researchers during the TSQL2 Design Effort, the glossary initiative and the benchmark initiative.

8.5 Ph.D. Projects

Efficient Monitoring of Database Transaction Dependencies

Name:	Lars Bækgaard
Education:	Cand.scient. (M.S.) in Computer Science, 1993
Duration:	September, 1990–January, 1993
Status:	Degree awarded February, 1993
Funding:	Department of Computer Science, Aarhus University
Advisor:	Leo Mark, then at University of Maryland

A declarative language, TDS, in which database evolution can be specified in terms of sequential dependencies between database transactions is described. More specifically, transaction dependencies are expressed as predicates on relational queries. This introduces a number of unsolved efficiency problems. Nested queries with set membership predicates or set inclusion predicates occur naturally in TDS specifications. An algorithm for transforming such queries into equivalent flat relational algebra queries is described. This unnesting algorithm makes it possible to utilize existing algorithms to optimize and efficiently evaluate nested queries. The correctness of the algorithm is formally proven. The unnesting algorithm transforms nested queries into queries that make extensive use of the relational set difference operator making the efficiency of this operator an important issue. Algorithms for the incremental evaluation of set differences are described. Time varying queries, i.e., queries in which selection or join predicates refer to the state of a clock, occur naturally in TDS specifications as a notationally convenient way to express preconditions. Algorithms for the incremental evaluation of such queries are described. The novel and useful notions of superviews, super caches, and predicate caches are introduced and described. In many cases the incremental algorithms are very cost efficient when compared to existing re-execution algorithms.

Aspects of Implementation of Temporal Databases

Name:	Kristian Torp
Education:	Cand.polyt. (M.E.) in Computer Science, 1994
Duration:	August, 1995–July, 1998 (expected)
Status:	In progress
Funding:	Institute of Electronic Systems, Aalborg University
Advisor:	Christian S. Jensen

At the time of writing, this Ph.D. project is still in the defining stages. A current goal is to investigate an architecture for a temporal database management system (TDBMS) that relies heavily on existing, commercially available database systems for its functionality, but also allows for the gradual use of temporal query

processing modules, as these become available. This architecture is new and has until recently not been considered appropriate for building a TDBMS. Therefore, the first step is an empirical validation of the architecture. Preliminary results are promising.

A next step is to detect performance bottlenecks in the architecture, and in temporal data management in general, and to then attempt to eliminate these. Special focus will be placed on storage management techniques when working with the bottlenecks. These techniques are promising, and only little work has been done is this area for temporal databases. As a basis for these activities, a survey of different approaches to storage management will be conducted. While general storage management techniques will be studied, emphasis will be on special temporal database features, e.g., partitioned storage, indexing, and timestamping.

8.6 Service and Research-related Activities

This section primarily documents services provided to the international scientific community by the group members. The members all benefit from the services provided by the community and thus find it essential to also provide services in return.

Conferences and Workshops

- Program committee member, *Fifth International Conference on Information and Knowledge Management*, USA, 1996 (M. Böhlen).
- Program committee member, *Logic in Databases*, San Miniato, Tuscany, Italy, June 1–2, 1996 (M. Böhlen).
- Program committee member, *Fifth International Conference on Extending Database Technology*, Avignon, France, March 1996 (C. S. Jensen).
- Program committee member, *Twelfth IEEE International Conference on Data Engineering*, February/March 1996 (C. S. Jensen).
- Presenter, with R. Snodgrass, of a half-day tutorial on Temporal Databases, at the *Twelfth International Conference on Data Engineering*, New Orleans, Louisiana, February/March 1996 (C. S. Jensen).
- Session chair, Non-Traditional Modelling Approaches, *Fourteenth International Conference on Object-Oriented and Entity Relationship Modeling*, Queensland, Australia, December 13–15, 1995 (C. S. Jensen).
- General chair and steering committee member (co-members include A. Segev (chair), J. Clifford (PC chair), T. Özsu, B. Pernici, N. Pissinou, and R. T. Snodgrass), *International Workshop on Temporal Databases*, Zürich, Switzerland, September 1995. This is the first workshop ever to be held

in cooperation with the VLDB Endowment, Inc. As general chair, I handled potential conflict-of-interest papers (C. S. Jensen).

- Panel participants, panel on "The State-of-the-Art in Temporal Data Management: Perspectives from the Research and Financial Applications Communities," *International Workshop on Temporal Databases*, Zürich, Switzerland, September 1995 (M. Böhlen and C. S. Jensen).
- Demo, with R. T. Snodgrass, and A. Steiner, of the TimeDB System, *International Workshop on Temporal Databases*, Zürich, Switzerland, September 1995 (M. Böhlen and C. S. Jensen).
- Presenter, with R. T. Snodgrass, of a half-day tutorial on Temporal Databases, at *Twentyfirst International Conference on Very Large Data Bases (VLDB)*, Zürich, Switzerland, September 1995 (C. S. Jensen).
- Presenter of a half-day tutorial on Temporal Databases, at *Database Workshop*, Århus, Denmark, June, 1995 (C. S. Jensen).
- Program committee member, *Sixth IFIP TC–2 Working Conference on Data Semantics (DS–6)*, Atlanta, GA, May/June 1995 (C. S. Jensen).
- Program committee member, *Eleventh IEEE International Conference on Data Engineering*, Taipei, Taiwan, February 1995 (C. S. Jensen).
- Program committee member, *Twentieth International Conference on Very Large Data Bases*, Santiago de Chile, Chile, August 1994 (C. S. Jensen).
- Recipient, Certificate of Appreciation, *1994 ACM International Conference on the Management of Data (SIGMOD)*, Minneapolis, MN, May 1994 (C. S. Jensen).
- Session chair, Spatial Joins, *1994 ACM International Conference on the Management of Data*, Minneapolis, MN, May 1994 (C. S. Jensen).
- Program committee member, *1994 ACM International Conference on the Management of Data*, Minneapolis, MN, May 1994 (C. S. Jensen).
- Program committee member, *Tenth IEEE International Conference on Data Engineering*, Houston, TX, February 1994 (C. S. Jensen).
- Program committee member, *ARPA/NSF International Workshop on an Infrastructure for Temporal Databases*, Dallas, TX, June 1993 (C. S. Jensen).

Editorial Activities

- Member of the TSQL2 Language Design Committee, June 1993–September 1995. Chair: R. T. Snodgrass. (C. S. Jensen)
- Coordinator and head of two editorial boards for the ongoing Consensus Temporal Database Glossary Initiative, March 1992–present. Co-editors

include J. Clifford, R. Elmasri, S. K. Gadia, P. Hayes, S. Jajodia, A. Segev, R. Snodgrass, and A. Tansel. In excess of 25 researchers are contributing. (C. S. Jensen)

- Editor for the Temporal Database Benchmark Initiative, March 1993-present. More than twenty researchers are contributing. (C. S. Jensen)

Reviewing

- *ACM Transactions on Database Systems* (M. Böhlen and C. S. Jensen)
- *The Computer Journal* (L. Bækgaard and C. S. Jensen)
- *IEEE Transactions on Knowledge and Data Engineering* (L. Bækgaard, M. Böhlen, C. S. Jensen, and K. Torp)
- *Information Processing Letters* (C. S. Jensen)
- *Information Sciences* (C. S. Jensen)
- *Information Systems* (C. S. Jensen)
- *The International Journal on Very Large Data Bases* (M. Böhlen and C. S. Jensen)
- The National Science Foundation (USA) (C. S. Jensen)
- The Swedish Research Council for Engineering Sciences (C. S. Jensen)
- External reviewing for several conferences, including *VLDB, ICDE, SIGMOD, 1995 International Workshop on Temporal Databases,* and *Hypertext*

Professional Organizations

- Member of the Association for Computing Machinery (L. Bækgaard, M. Böhlen, and C. S. Jensen)
- Member of the Institute of Electrical and Electronics Engineers (L. Bækgaard, M. Böhlen, and C. S. Jensen), including the IEEE Computer Society and the Technical Committee on Data Engineering (C. S. Jensen)
- Member of the ZobIS (Zeitorientierte betriebliche Informationssysteme) working group. Part of the German Gesellschaft für Informatik (C. S. Jensen)

8.7 The Group's Own Evaluation

At the start of the evaluation period, the group consisted of a newly hired assistant professor and a Ph.D. student. At that time, the most important objectives for 1991–1995 were to create a well-functioning, visible, internationally oriented, and independent group that was an equal to the already established groups.

While these objectives obviously were quite ambitious, they were nevertheless felt to be important *goals*; creation of a fertile research environment and consolidation of the group were vital to the group's meaningful existence as a research *group* in the longer term. Specific means included the recruitment of more faculty to the group, the recruitment of a number Ph.D. and M.S. students into the group, and the establishment of a database systems laboratory.

Recruiting new faculty within databases in Denmark is difficult because database research is rather neglected at Danish universities. However, the group has recently been able to attract a faculty member with a Ph.D. from one of Europe's most prestigious computer science institutions, ETH Zürich. The group has also recruited one Ph.D. student from the first group of students to graduate with an M.S. in databases; and during the last three years, the group has attracted 16 M.S. students.

The group shares laboratory space with another group. Hardware facilities in the laboratory have included an adequate number of workstations, and the processing capabilities has been satisfactory for all software, with the exception of Ingres, which needs more powerful hardware. In addition to Ingres (now being changed to Oracle), which has been used for both teaching and research, various free database software, e.g., Postgres, Ode, ChronoLog, ChronoSQL, TimeIT, SDT, and ConceptBase, have been installed for use in projects.

During the period, the group has been responsible for teaching both an introductory and an advanced course on database managment and systems. These courses have been well-received by the students, with good evaluations and attendance. It is felt that the window they have provided into databases is the major reason why the group has been able to attract a good portion of M.S. students. The group is worried that reductions of its current presence in the curriculum may jeopardize the group's ability to attract M.S. students.

Three means of obtaining visibility and maintaining an international orientation were identified, namely publication of research results in well-recognized international outlets, participation in professional activities, and collaboration with established research institutions.

The group aimed at publishing its work in the outlets that present the internationally (rather than merely nationally or regionally) best research results. It is felt that the group achieved this goal and has effectively disseminated its work to the international community. Specifically, the majority of the research results of the group have been published in very reputable international journals and conferences.

During the period, the group has been represented on the program committees of all the most prestigious database conferences. In addition, the group members have served as reviewers for all the best database journals. Within temporal databases, the group has played a central role at the two workshops held during the period. At the first workshop, concerning an infrastructure for temporal databases, the Consensus Temporal Database Glossary Initiative and

the Temporal Database Benchmark Initiative were prominent ingredients, and the group was also represented on the program committee for that workshop. At the second workshop, the group was represented in the organizing team, in the General Chair position, and in the Steering Committee. Also at that workshop, members of the group participated in panels and in the demonstration of their TimeDB temporal database management system prototype. The group members have also been centrally involved in designing and partially implementing TSQL2, a consensual temporal database query language and data model that extends SQL–92 with temporal support. TSQL2 was designed by an international panel of temporal database researchers and is fast becoming a central part of the temporal database community infrastructure. The group is also working with the standards bodies to migrate aspects of TSQL2 into the SQL/Temporal part of the SQL3 standard. Finally, the group has promoted temporal databases (and its research and TSQL2) by giving invited talks and temporal database tutorials, e.g., at two general database conferences.

As witnessed by an earlier section, the group has also collaborated closely with leading international researchers. Indeed, the group members have visited and worked with leading international researchers within their respective areas. The group members appreciate very much the opportunities they have been given to carry out the visits. Indeed, these stays have had a tremendously positive effect on the group members' research.

On this basis, it is felt that the group has reached its goal of obtaining high visibility.

However, the many activities during the period come at a cost. The service activities are by themselves quite time consuming. For example, providing quality reviews for conferences and journals has required substantial effort. Remembering the reductions in research time, as described in the general chapter for the unit, it should not be surprising that service by itself at times has occupied almost all the regular time available for research. The number of research activities has only been possible through a consistent effort that substantially exceeds the normed work load. In the evaluation period, it has also been possible to boost research through relatively frequent half-year visits at international research centers. As this often does not go well with having a family, this opportunity may not be available to the same degree in the period to come. Finally, it is noted that the rewards for a substantial investment in research are currently external to the Faculty of Engineering and Science and compete with other rewarding activities.

A final objective was to achieve research collaboration within the group by means of defining a so-called "umbrella" project. Unfortunately, there has not been joint research between the group members during the five-year period.

8.8 Research Plan 1996–2000

8.8.1 Overall Goals

The group expects to maintain its international focus during the next period. The means include the continued publication of research results in recognized international outlets, continued participation in activities in the international research community, and continued cooperation with established research institutions. The group expects to obtain good visibility in the general database community and very high visibility in temporal databases.

In the previous period, the group successfully obtained a strong position in the international temporal database community. In the next period, the group will build on this foundation by attempting to make temporal databases its main, but not sole, research focus. The TimeCenter project, described below, will serve as a framework for most of the temporal database activities. The Data Warehousing and Chorochronos projects (also described below) are also important contexts for research.

A quick survey of international database journals and conference proceedings clearly demonstrates that database research at Danish universities is in its infancy. This is in contrast to the surrounding countries, e.g., Germany, Norway, and Sweden, and it is also in contrast to the widespread use of database technology in Denmark. The group would like to take an active part in activities aimed at changing this state of affairs.

The presence of a solid student population within databases may become an important means of maintaining a fertile database research environment. With the reduction of database topics in the curriculum, the group must expect to have to devote extra resources to attracting and maintaining a student population.

Departmental policy for the next five years states that the Database Systems Group can have at most two tenured faculty members; and special permission is needed to participate in announcements of un-tenured positions when the sum of tenured and un-tenured faculty exceeds one. It is currently not realistic to assume that there will be Danish applicants for assistant-level database positions. Rather, applicants must be found internationally. Based on this outlook, it is a goal to ensure that the group does not split into separate subgroups—this would jeopardize the presence of a database research *group* in the unit.

In relation to the above, it is a final, but important, goal for the next period is to establish research collaboration among all the members of the Database Systems Group. This is important in order for the participants to achieve the benefits that stem from being part of a research group, e.g., easy day-to-day feedback on research work, greater external visibility, being more attractive as a partner in larger national and international projects, and being able to define larger projects internally.

Descriptions of three specific projects follow. The actual work put into each

project will depend substantially on the future staff situation. Christian S. Jensen has received grants from the Danish Technical Research Council and the European Commission for research within TIMECENTER and Chorochronos, respectively. His main activities will be within these projects. Michael Böhlen participates in Chorochronos and has committed to work on TIMECENTER. Lars Bækgaard is the primary force behind the Data Warehousing project, but Michael Böhlen and Christian S. Jensen are also interested in the application of temporal database techniques in data warehousing. The Ph.D. studies of Heidi Gregersen and Kristian Torp are within TIMECENTER.

8.8.2 Data Warehousing

Data warehousing is an upcoming idea in the database community that can utilize many of the group's research results. Data warehouses can be defined in several ways. For example, a data warehouse can be a view defined on a collection of source databases. A warehouse can be used for many different purposes. It can be used as a basis for data mining, statistical analysis, decision support systems, ad hoc querying etc. From a user's point of view a warehouse is read-only. It is only updated in order to reflect changes in the source databases. This indicates that a warehouse should be seen as a medium for data analysis and querying.

The sources may be loosely connected to each other and to the warehouse. Therefore, the warehouse must be a materialized view that is maintained incrementally. This raises a number of research questions that are concerned with efficiency. How is the view to be organized and maintained? How are changes in the source databases propagated to changes in the warehouse? Are the sources responsible for the propagation? Is the warehouse responsible for the propagation? Which data structures could support efficient change propagation? To what extend can network communication between sources and the warehouse be minimized? The group's previous results in incremental view maintenance should be utilized to solve these problems.

The definition and maintenance of the warehouse view present a number of research questions that are related to database design. The warehouse is a time-varying view that must be updated from time to time. Within which time constraints (momentarily, daily, weekly etc.) should the warehouse be brought up-to-date? The source databases may be defined using any data model, including flat file models. What methods could or should be used for the definition of an integrated view of a number of heterogeneous sources? To what extent could the definition and maintenance of the view get DBMS support? The group's previous results in temporal databases and data modeling should be utilized to solve these problems.

The use of warehouse data presents a wide variety of research challenges. The idea of data mining is a very fast-growing research topic that must be combined with the idea of data warehousing. A warehouse can contain potentially

invaluable information, but its users may not be able to retrieve the relevant information in a relevant form. Traditional query languages like SQL are not well suited to solve such problems. Such query languages presume that the users know what they want and how to get it. Data mining techniques can be used to (semi)automatically identify information patterns that may be relevant. Techniques like co-occurrence analysis, fuzzy logic, statistical analysis could be very useful. The group's previous results in temporal databases, behavior mining, and cooperative query-answering should be utilized to solve these problems.

8.8.3 TimeCenter

TimeCenter is envisioned as an international center for the support of temporal database applications using traditional and emerging DBMS technologies. It is expected to provide the context for much of the group's research in temporal databases in the coming period.

Below, we describe the background for the center.

A wide range of database applications manage time-varying information. These include financial applications such as portfolio management, accounting, and banking; record-keeping applications, including personnel, medical-record, and inventory; and travel applications such as airline, train, and hotel reservations, and schedule management.

Currently, such applications typically run on top of conventional relational systems. However, recent advances in temporal data model and query language technologies clearly demonstrate that applications that manage temporal data may benefit substantially from built-in temporal support in the database management system. The potential benefits from such support are several. Application code is substantially simplified. Due to faster development of code, which is also easier to comprehend and thus maintain, higher programmer productivity results. With built-in support, more data processing may be left to the DBMS, potentially leading to better performance.

Yet, the temporal support in current commercial database management products is at a very primitive stage. The TimeCenter will attempt to identify and eliminate the technical obstacles to bringing state-of-the-art temporal data management technologies into practice.

One obstacle is the large investment in existing, or legacy, database application code based on current, non-temporal database system technologies. It is unrealistic to expect potential users to abandon existing application code to adopt a temporally enhanced database management system. Rather, a smooth migration of the bulk of application code from the current systems to temporal database management systems must be possible. There is a need to explore the requirements, beyond upwards compatibility, that temporal data models and query languages must satisfy to guarantee a smooth transition to a temporal database management system and a subsequent harmonious coexistence of legacy code

and new temporal application code. Finally, there is a need to explore how such requirements may be satisfied by temporal data models and query languages.

The predominant assumption—that a temporal data model and query language must be supported by a database management system that is new and inherently temporal—presents another obstacle. Building a complete temporal database management system is at best a lengthy and labor-intensive task. Complicating factors include the existing lack of insight into the relevant implementation aspects, such as temporal data representation, indexing, query processing, and transaction processing.

To eliminate this obstacle, we propose the notion of a migratable temporal database systems architecture is proposed. With this architecture, a temporal data model and query language may, in the short term, be implemented on top of an existing database management system. This provides early availability of temporal support. As temporal implementation techniques become well-known and available, the architecture permits a smooth integration of these into the system. Specifically, adding a new component does not necessitate changes to existing components. As more and more system components add temporal support, improved performance of temporal-data query processing will result.

We also stress that temporal implementation techniques are important, in the medium and long term, in order to provide enhanced performance. To facilitate the development of such techniques, the development of a test environment is part of TIMECENTER.

A third obstacle is the lack of a user-friendly methodology for the design of temporal databases. Such a methodology exploits the temporal semantics of data that may be captured by a temporal data model. It is maximally independent of the available implementation platform for the database. This provides platform independence and reuse of designs. Also, a temporal design methodology fully exploits the functionality available on the given implementation platform.

This leads to two foci of the TIMECENTER. First, the center will explore the theoretical foundations that enable the migration of temporal database technology from the research community to its practical adoption in temporal data management applications. Second, the center will develop an integrated architecture that supports all aspects of temporal data management applications, covering both design and production environments. In design, the Center will focus on all aspects of database design: conceptual, logical, and physical database design. In production, the Center will explore the foundations for the processing of temporal queries on platforms ranging from currently available relational platforms to novel, higher-performance temporal ones.

In summary, is is an central goal of TIMECENTER to seek to provide *technical* solutions to the oft-mentioned technology transfer problem.

8.8.4 Chorochronos

Chorochronos is a research network devoted to the design, implementation, and application of database technology for the handling of spatio-temporal information, i.e., the temporal and spatial information associated with objects in the world. The network is funded by the European Commission under the TMR Programme, and the participants include National Technical University of Athens, Aalborg University, Fern Universität Hagen, University of L'Aquila, UMIST, Manchester, Politecnico di Milano, Vertigo group, CNAM, Paris, University of of Thessaloniki, Technical University of Vienna, and ETH Zürich.

Spatial and temporal databases have been important sub-areas of database research for a long time. Researchers in both areas have always felt that there are important connections in the problems addressed by each area, and the techniques and tools utilized for their solution. There are many publications in temporal databases which conclude with the phrase "the ideas in this paper can be extended to spatial data management." Similarly, many papers in spatial databases suggest that techniques developed for spatial databases are applicable for the management of temporal data by restricting attention to one dimension only. But up to now, little has been done for the systematic interaction and synergy of these two areas so that the respective claims can be formally *verified, refuted*, or *appropriately qualified*. This network is devoted to establishing exactly this kind of interaction and synergy between European researchers.

We believe that applications like GIS, remote sensing, environmental information systems and CAD/CAM are important, and spatio-temporal database management systems (STDBMS) can become the enabling technology for these applications. This is not just our belief, but also that of a large class of users from the above application areas.

Chorochronos aims at enabling European researchers in spatial and temporal databases to coordinate their work and integrate their findings in their respective areas of expertise. The participants are established researchers in temporal and spatial database systems. Most of them have so far been working exclusively on temporal or spatial databases, and can now work together to unite their expertise for the creation of a spatio-temporal database infrastructure.

Bibliography

Ph.D. Theses

[1] Böhlen, M., "Managing Temporal Knowledge in Deductive Databases," Department of Computer Science, ETH Zürich, 1994.

[2] Bækgaard, L., "Efficient Monitoring of Database Transaction Dependencies." Departmental Technical Report number R-93–2026, 1993.

[3] Jensen, C. S., "Towards the Realization of Transaction Time Database Systems." Departmental Technical Report number R-91–11, March 1991, 151 pages.

Refereed Journal Articles

[4] Bækgaard, L. and L. Mark, "Incremental Computation of Nested Relational Query Expressions," *ACM Transactions on Database Systems*, Vol. 20, No. 2, June 1995, pp. 111–148.

[5] Bækgaard, L. and L. Mark, "Incremental Computation of Time-Varying Query Expressions," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 7, No. 4, 1995, pp. 583–590.

[6] Jensen, C. S., M. D. Soo, and R. T. Snodgrass, "Unifying Temporal Data Models via a Conceptual Model," *Information Systems*, Vol. 19, No. 7, December 1994, pp. 513–547.

[7] Jensen, C. S. and R. T. Snodgrass, "Temporal Specialization and Generalization," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 6, No. 6, December 1994, pp. 954–974.

[8] Jensen, C. S., L. Mark, N. Roussopoulos and T. Sellis, "Using Differential Techniques to Efficiently Support Transaction Time," *The VLDB Journal*, Vol. 2, No. 1, January 1993, pp. 75–111.

[9] Jensen, C. S. and L. Mark, "Queries on Change in an Extended Relational Model," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 4, No. 2, April 1992, pp. 192–200.

[10] Jensen, C. S., L. Mark and N. Roussopoulos, "Incremental Implementation Model for Relational Databases with Transaction Time," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 3, No. 4, December 1991, pp. 461–473.

Refereed Conference Articles

[11] Jensen, C. S. and R. T. Snodgrass, "Semantics of Time-Varying Attributes and Their Use for Temporal Database Design," *Fourteenth International Conference on Object-Oriented and Entity Relationship Modeling*, Queensland, Australia, December 13–15, 1995, pp. 366–377.

[12] Bækgaard, L., J. H. Jørgensen, T. L. Nielsen, and D. B. Rasmussen, "Fuzzy Ranking of Documents Based on Behavior Mining," *15th Database Conference. DATASEM'95*, Brno, 1995, pp. 265–274.

[13] Böhlen, M., C. S. Jensen, and R. T. Snodgrass, "Evaluating the Completeness of TSQL2," *Proceedings of the International Workshop on Temporal Databases*, Zürich, Switzerland, September 17–18, 1995, pp. 153–172. The proceedings are entitled *Recent Advances in Temporal Databases* and are published by Springer-Verlag in their Workshops in Computing Series.

[14] Böhlen, M. and R. Marti, "On the Completeness of Temporal Database Query Languages," *Proceedings of the First International Conference on Temporal Logic*, Bonn, Germany, July 11-14, 1994, pp. 283–300. Published as Lecture Notes in Artificial Intelligence, No. 827, edited by D. M. Gabbay and H. J. Ohlbach, Springer-Verlag.

[15] Soo, M. D., R. T. Snodgrass, and C. S. Jensen, "Efficient Evaluation of the Valid-Time Natural Join," *Proceedings of the Tenth IEEE International Conference on Data Engineering*, Houston, TX, February 14–18, 1994, pp. 282–292.

[16] Jensen, C. S. and R. T. Snodgrass, "Three Proposals for a Third-Generation Temporal Data Model," *Proceedings of the International Workshop on an Infrastructure for Temporal Databases*, Arlington, TX, June 14–16, 1993, pp. T-1–T10.

[17] Jensen, C. S., M. D. Soo, and R. T. Snodgrass, "Unification of Temporal Data Models," *Proceedings of the Ninth IEEE International Conference on Data Engineering*, Vienna, Austria, April 19–23, 1993, pp. 262–271.

[18] Böhlen, M. and R. Marti, "Handling Temporal Knowledge in a Deductive Database System," *Datenbanksysteme in Büro, Technik und Wissenschaft*, edited by W. Stucky and A. Oberweis, Springer-Verlag, 1993.

[19] Jensen, C. S. and R. T. Snodgrass, "Temporal Specialization," *Proceedings of the Eighth IEEE International Conference on Data Engineering*, Phoenix, AZ, February 3–6, 1992, pp. 594–603.

Book Chapters

[20] Snodgrass, R. T., C. S. Jensen, C. E. Dyreson, W. Kæfer, N. Kline, and John F. Roddick, "A Second Example," Chapter 4, pp. 49–73, in *The TSQL2 Temporal Query Language*, edited by R. T. Snodgrass, Kluwer Academic Publishers, 1995, 674+xxiv pages.

[21] Jensen, C. S. and R. T. Snodgrass, "The Surrogate Data Type," Chapter 9, pp. 153–156, in *The TSQL2 Temporal Query Language*, edited by R. T. Snodgrass, Kluwer Academic Publishers, 1995, 674+xxiv pages.

[22] Jensen, C. S., R. T. Snodgrass, and M. D. Soo, "The TSQL2 Data Model," Chapter 10, pp. 157–240, in *The TSQL2 Temporal Query Language*, edited by R. T. Snodgrass, Kluwer Academic Publishers, 1995, 674+xxiv pages.

[23] Snodgrass, R. T., C. S. Jensen, and M. D. Soo, "Schema Specification," Chapter 11, pp. 241–243, in *The TSQL2 Temporal Query Language*, edited by R. T. Snodgrass, Kluwer Academic Publishers, 1995, 674+xxiv pages.

[24] Snodgrass, R. T., C. S. Jensen, and F. Grandi, "The From Clause," Chapter 12, pp. 245–249, in *The TSQL2 Temporal Query Language*, edited by R. T. Snodgrass, Kluwer Academic Publishers, 1995, 674+xxiv pages.

[25] Hsu, S., C. S. Jensen, and R. T. Snodgrass, "Valid-Time Selection and Projection," Chapter 13, pp. 251–298, in *The TSQL2 Temporal Query Language*, edited by R. T. Snodgrass, Kluwer Academic Publishers, 1995, 674+xxiv pages.

[26] Leung, T. Y. C., C. S. Jensen, and R. T. Snodgrass, "Update," Chapter 14, pp. 299–304, in *The TSQL2 Temporal Query Language*, edited by R. T. Snodgrass, Kluwer Academic Publishers, 1995, 674+xxiv pages.

[27] Jensen, C. S., R. T. Snodgrass, and T. Y. C. Leung, "Cursors," Chapter 15, pp. 305–310, in *The TSQL2 Temporal Query Language*, edited by R. T. Snodgrass, Kluwer Academic Publishers, 1995, 674+xxiv pages.

[28] Clifford, J., C. E. Dyreson, R. T. Snodgrass, T. Isakowitz, and C. S. Jensen, "Now," Chapter 20, pp. 385–394, in *The TSQL2 Temporal Query Language*, edited by R. T. Snodgrass, Kluwer Academic Publishers, 1995, 674+xxiv pages.

[29] Jensen, C. S., "Vacuuming," Chapter 23, pp. 451–462, in *The TSQL2 Temporal Query Language*, edited by R. T. Snodgrass, Kluwer Academic Publishers, 1995, 674+xxiv pages.

[30] Soo, M. D., C. S. Jensen, and R. T. Snodgrass, "An Architectural Framework," Chapter 24, pp. 465–473, in *The TSQL2 Temporal Query Language*, edited by R. T. Snodgrass, Kluwer Academic Publishers, 1995, 674+xxiv pages.

[31] Soo, M. D., C. S. Jensen, and R. T. Snodgrass, "An Algebra for TSQL2," Chapter 27, pp. 505–546, in *The TSQL2 Temporal Query Language*, edited by R. T. Snodgrass, Kluwer Academic Publishers, 1995, 674+xxiv pages.

[32] Snodgrass, R. T., I. Ahn, G. Ariav, D. S. Batory, J. Clifford, C. E. Dyreson, R. Elmasri, F. Grandi, C. S. Jensen, W. Käfer, N. Kline, K. Kulkarni, T. Y. C. Leung, N. Lorentzos, J. F. Roddick, A. Segev, M. D. Soo, and S. M. Sripada, "Language Syntax," Chapters 28–38, pp. 549–631, in *The TSQL2 Temporal Query Language*, edited by R. T. Snodgrass, Kluwer Academic Publishers, 1995, 674+xxiv pages.

[33] Jensen, C. S. and L. Mark, "Differential Query Processing in Transaction-Time Databases," Chapter 19, pp. 457–491, in *Temporal Databases: Theory, Design, and Implementation*, edited by A. Tansel et al., Benjamin/-Cummings Publishers, Database Systems and Applications Series, 1993. Invited and refereed.

[34] Jensen, C. S., J. Clifford, S. K. Gadia, A. Segev, and R. T. Snodgrass "A Glossary of Temporal Database Concepts," Appendix A, pp. 621–633, in *Temporal Databases: Theory, Design, and Implementation*, edited by A. Tansel et al., Benjamin/Cummings Publishers, Database Systems and Applications Series, 1993. Invited.

Invited Papers and Unrefereed Journal Articles

[35] R. T. Snodgrass, M. Böhlen, C. S. Jensen, and A. Steiner, "Change Proposal to SQL/Temporal: Adding Valid Time — Part A." The proposal was submitted to the *International Organization for Standardization* as an ANSI Expert's Contribution, December 5, 1995, 40 pages.

[36] Segev, A., C. S. Jensen, and R. T. Snodgrass, "Report in The 1995 International Workshop on Temporal Databases," in *ACM SIGMOD Record*, Vol. 24, No. 4, December 1995, pp. 46–52.

[37] Bækgaard, L., J. H. Jørgensen, T. L. Nielsen, and D. B. Rasmussen, "Fuzzy Behavior Mining in Information Retrieval Systems," *FQAS'94 Workshop on Flexible Query-Answering Systems*, Roskilde University, Denmark, 1994, pp. 113–126.

[38] Snodgrass, R. T., I. Ahn, G. Ariav, D. S. Batory, J. Clifford, C. E. Dyreson, R. Elmasri, F. Grandi, C. S. Jensen, W. Käfer, N. Kline, K. Kulkarni, T. Y. C. Leung, N. Lorentzos, J. F. Roddick, A. Segev, M. D. Soo, and S. M. Sripada, "A TSQL2 Tutorial," in *ACM SIGMOD Record*, Vol. 23, No. 3, September 1994, pp. 27-33.

[39] Dahl, K., H. Gregersen, C. A. Have, C. S. Jensen, J. Sigurðsson, J. S. Winter, "Database-benchmarks," in *PROSA-bladet*, No. 5, May 1994, pp. 15–17 (in Danish).

[40] Jensen, C. S. et al. (editors, with multiple other contributors), "A Consensus Glossary of Temporal Database Concepts," in *ACM SIGMOD Record*, Vol. 23, No. 1, March 1994, pp. 52–65 (Special Section: Temporal Database Infrastructure).

[41] Snodgrass, R. T., I. Ahn, G. Ariav, D. S. Batory, J. Clifford, C. E. Dyreson, R. Elmasri, F. Grandi, C. S. Jensen, W. Käfer, N. Kline, K. Kulkanri, T. Y. C. Leung, N. Lorentzos, J. F. Roddick, A. Segev, M. D. Soo, and S. M. Sripada, "TSQL2 Language Specification," in *SIGMOD Record*, Vol. 23, No. 1, March, 1994, pp. 65–86 (a preliminary version).

[42] Jensen, C. S. (editor, with multiple other contributors), "Proposed Temporal Database Concepts—May 1993" (including an addendum), in *Proceedings of the International Workshop on an Infrastructure for Temporal Databases,* Arlington, TX, June 14–16, 1993, pp. A-1–A-29.

[43] Jensen, C. S. (editor, with multiple other contributors), "The TSQL2 Benchmark," in *Proceedings of the International Workshop on an Infrastructure for Temporal Databases,* Arlington, TX, June 14–16, 1993, pp. QQ-1–QQ-28.

[44] Jensen, C. S., J. Clifford, S. K. Gadia, A. Segev, and R. T. Snodgrass, "A Glossary of Temporal Database Concepts," in *ACM SIGMOD Record*, Vol. 21, No. 3, September 1992, pp. 35–43.

Technical Reports and Other Unrefereed Publications

The list includes only reports *not* published in journals and conferences in revised form. A number of the reports are in submission.

[45] Böhlen, M., R. T. Snodgrass, and M. D. Soo, "Coalescing in Temporal Databases." Manuscript submitted for publication. 27 pages.

[46] Böhlen, M., J. Chomicki, R. T. Snodgrass, and D. Toman, "Querying TSQL2 Databases with Temporal Logic." This manuscript will appear in the proceedings of the *Fifth International Conference on Extending Database Technology*, Avignon, France, March 1996.

[47] Bækgaard, L., "The Predicate Indexed Incremental Join." Departmental Technical Report number R-95–2019, 1995.

[48] Bækgaard, L. and J. C. Godskesen, "Transaction-Based Specification and Verification of Real-Time Dynamic Database Constraints." Departmental Technical Report number R-95–2018, 1995.

[49] Bækgaard, L. (editor), "Database Workshop III." Departmental Technical Report number IR-95–2003, 1995.

[50] Torp, K., L. Mark, and C. S. Jensen, "Efficient Differential Timeslice Computation." Departmental Technical Report number R-94–2055, December 1994, 26 pages.

[51] Jensen, C. S., R. T. Snodgrass, and M. D. Soo, "Extending Existing Dependency Theory to Temporal Databases." Departmental Technical Report number R-94–2050, December 1994, 58 pages. This paper has been accepted for publication in *IEEE Transactions on Knowledge and Data Engineering.*

[52] Clifford, J., C. Dyreson, T. Isakowitz, C. S. Jensen, and R. T. Snodgrass, "On the Semantics of 'Now' in Temporal Databases." Departmental Technical Report number R-94–2047, November 1994, 50 pages. In submission.

[53] Bækgaard, L., J. H. Jørgensen, T. L. Nielsen, and D. B. Rasmussen, "CoopBase: A Cooperative Information Retrieval Prototype." Departmental Technical Report number R-94–2037, 1994.

[54] Bækgaard, L. and L. Mark, "Incremental Computation of Set Difference Views," Departmental Technical Report number R-94–2036, 1994. This paper has been accepted for publication in *IEEE Transactions on Knowledge and Data Engineering.*

[55] Bækgaard, L. and L. Mark, "Incremental Computation of Time-Varying Queries," Departmental Technical Report number R-94–2035, 1994. Parts of this report has been published elsewhere.

[56] Jensen, C. S. (editor) et al., "A Consensus Test Suite of Temporal Database Queries." Departmental Technical Report number R-93–2034, November 1993, 45 pages.

[57] Bækgaard, L. and L. Mark, "TDS – A Language for the Specification of Dynamic Integrity Constraints." Departmental Technical Report number R-94–2033, 1994.

[58] Bækgaard, L. (editor), "Database Workshop II." Departmental Technical Report number IR-94–2032, 1994.

[59] Böhlen, M., "Valid Time Integrity Constraints." Technical Report TR 94-30, University of Arizona, Department of Computer Science, Tucson, AZ 85721, November 1994, 23 pages.

[60] Snodgrass, R. T. (editor), I. Ahn, G. Ariav, P. Bayer, J. Clifford, C. E. Dyreson, F. Grandi, L. Hermosilla, C. S. Jensen, W. Käfer, N. Kline, K. Kulkarni, T. Y. C. Leung, N. Lorentzos, Y. Mitsopoulos, J. F. Roddick, M. D. Soo, and S. M. Sripada, "An Evaluation of TSQL2," 53 pages, in *The TSQL2 Language Specification*, University of Arizona, Department of Computer Science, Tucson, Arizona 85721, USA, September 1994, 546 pages.

[61] Bækgaard, L. (editor), "Database Workshop I." Departmental Technical Report number IR-94–2002, 1994.

[62] Dyreson, C., R. T. Snodgrass, and C. S. Jensen, "On the Semantics of 'Now' in Temporal Databases," TempIS Technical Report 42, University of Arizona, Department of Computer Science, Tucson, Arizona 85721, USA, April 1993, 22 pages.

[63] Jensen, C. S. and M. D. Soo, "Temporal Joins in a Two-Level Data Model," Unpublished manuscript. June 1992. 24 pages.

[64] Soo, M. D., R. T. Snodgrass, C. E. Dyreson, C. S. Jensen, and N. Kline, "Architectural Extensions to Support Multiple Calendars," TempIS Technical Report 32, University of Arizona, May 1992, 74 pages.

[65] Böhlen, M. and R. Marti, "A Temporal Extension of the Deductive Database System ProQuel," Technical Report, Department of Computer Science, ETH Zürich, 1992.

[66] Bækgaard, L. and L. Mark, "Specification and Monitoring of Transaction Dependencies," CS-TR–2678, Department of Mathematics and Computer Science, University of Maryland, USA, 1991.

Software

[67] Böhlen, M., "ChronoLog—A Temporal Deductive Database System," Release 4.0, December 1995.

[68] Steiner, A. and Böhlen, M., "The TIMEDB Temporal Database Prototype," Release 1.0, September 1995.

[69] Bækgaard, L., "CoopBase—A Cooperative Information retrieval Prototype," Release 1.0, September 1995.

[70] Snodgrass, R. T., C. E. Dyreson, C. S. Jensen, N. Kline, M. D. Soo, L. So, and J. Whelan, "The MULTICAL Temporal Database Management System," Release 1.0, October 1993 and Release 1.1, September 1995. For additional detail, see Section 8.2.

Manuals

[71] Böhlen, M., "ChronoLog 4.0—Reference Manual," December 1995.

[72] Snodgrass, R. T., C. E. Dyreson, C. S. Jensen, N. Kline, L. Soo, and M. D. Soo, "The MULTICAL System," Manual and Systems Documentation, University of Arizona, Department of Computer Science, Tucson, Arizona 85721, USA, June 1993, ca. 330 pages.

Chapter 9

Decision Support Systems

9.1 Profile

Our activities are a part of *Artificial Intelligence* (AI). There are several approaches in AI. Our approach aims at constructing systems that solve human tasks—hopefully better than humans—and the tasks that we are interested in are decision making, particularly decisions under uncertainty.

So, we are not aiming at building systems that "think" as humans, nor are we aiming at systems that either act as or replace humans. Rather, we are aiming at systems that can advise humans on what to do if they act "rationally." By *acting rationally* we mean acting so as to achieve one's goals given one's beliefs.

Our approach to reasoning under uncertainty is the so-called *Bayesian* approach (which in fact is classical probability theory), and our approach to decisions is that of maximizing the expected utility (the so-called *normative* approach). When building decision support systems we construct models of the domain rather than models of the decision maker's lines of reasoning.

A language for building such models is *Bayesian networks*, which are graphical models of the causal relations in the domain, with conditional probabilities modeling the strengths of the causal links. When decisions are incorporated in the models they are called *influence diagrams*.

Bayesian networks were introduced in the mid 80's, and by the end of that decade efficient algorithms were available for entering evidence and updating probabilities given evidence. A group at Aalborg University—ODIN, described below—played an active role in this development.

ODIN developed the so-called Lauritzen-Spiegelhalter method and improved it to the HUGIN-method, which is now recognized as the most efficient updating method for Bayesian networks. It is implemented in the DSS-shell HUGIN, which is marketed worldwide by Hugin Expert Ltd. situated at NOVI (the Science Park of Northern Jutland).

In the spring of 1991, the ODIN group received a 4-year funding (DKK 2.5

million pr. year) from the Danish Research Councils through the so-called PIFT Programme, and in the fall of 1991 the leader of ODIN, Finn V. Jensen, became a member of the Computer Science unit, forming the Decision Support Systems Group.

The aims and scope of the DSS Group (as part of the ODIN Programme) have been to extend the scope of normative systems. Particularly to

- develop and implement methods for fast belief updating
- implement tractable methods for various decision support system tasks
- construct decision support systems for specific domains (particularly that of agriculture)

9.2 Activities and Results

The core of the group's activities has been a part of the ODIN Programme. The participants in ODIN have come from the DSS Group, from the Statistics Group, and from the Medical Informatics Groups at the Department of Medical Informatics and Image Analysis (MIBA). The DSS Group has contributed to the ODIN Programme with the following results:

- a method and an implemented system for analyzing evidence for internal conflicts—e.g., to identify "red herrings" [5]
- experiments with the simulated annealing method for triangulation. Triangulation is used for building the basic data structure for the HUGIN method. Its optimality is NP-hard [9]
- an implemented system for automatic adaptation of networks to incoming experience [17]
- methods and an implemented system for handling models for domains varying over time [14, 16]
- a method and an implemented system for constructing simple structures for approximate belief updating [24, 15]
- a system for analyzing which information source to consult next [21, 6]
- a method and a system exploiting the HUGIN method for solving influence diagrams [11]
- methods solving the optimality issues for HUGIN propagation - except triangulation, which is NP-hard [13]
- a new design of the stochastic method, Gibbs sampling [8]
- methods for sensitivity analysis: which parts of the evidence have had which impact on the system's conclusions? [19]

- a propagation method supporting sensitivity analysis and conflict analysis [12]

The ODIN Group has also taken part in the ESPRIT Basic Research action DRUMS (Defeasable Reasoning and Uncertainty Management Systems). The DRUMS consortium covers all approaches to formal uncertainty management, and the ODIN Group was invited to contribute on Bayesian methods. The contributions from the DSS Group to the DRUMS Programme are concentrated on developing a design for combining exact and stochastic methods for belief updating [4, 10], and to contribute to a *Handbook of Defeasable Reasoning* (in preparation).

In cooperation with the Image Analysis Group at MIBA, we have worked on using Bayesian networks for automatic object recognition [20, 22].

9.3 Organization and Staff

The tables below give the numerical information on the staffing (both on teaching and pure research) in the period. Numerical information is given for each year with more detailed half-year information provided by left- or right-shifting of numbers.

Teaching Positions					
	1991	1992	1993	1994	1995
Full Professors					
Associate Professors	1	1	1	1	1
Assistant Professors				1	1

Research Positions					
Visiting Professors					
Visiting Researchers		1	1	1	
Research Assistants	1	1	2	2	
Ph.D. Students	2 2	3 2	4 3	3 3	3 3

The following persons have worked on the group's research tasks during the period. Finn V. Jensen and Uffe Kjærulff were employed throughout the period.

Finn Verner Jensen

Academic Degrees

1977 Pædagogikum (Secondary School Teacher's Certificate)
1974 Dr.math. in Mathematical Logic, Warsaw University
1970 Cand.scient. (M.S.) in Mathematics, Århus University

Positions

1989–	Reading Professor (Docent), Aalborg University
1986–89	Project Manager, JUDEX Ltd., Aalborg
1974–86	Associate Professor, Aalborg University
1972–74	Research Fellow (Kandidatstipendiat), Århus University (studies in Warsaw)
1970–72	Scientific Assistant (Kandidatinstruktør), Århus University
1969–70	Scientific Assistant (Videnskabelig Assistent), Oslo University

Uffe Kjærulff

Academic Degrees

1993	Ph.D. in Computer Science, Aalborg University
1985	Cand.polyt. (M.E.) in Computer Engineering, Aalborg University

Positions

1994–	Assistant Professor, Aalborg University
1993–94	Research Assistant, Aalborg University
1990–93	Ph.D. Scholarship, Aalborg University
1989–90	Ph.D. Scholarship, JUDEX A/S, Aalborg
1985–89	Research Assistant, JUDEX A/S, Aalborg

Other Staff

Frank Jensen (1991–1994 as a research assistant)
Kristian G. Olesen (1991–1992 as a Ph.D. student)
Søren L. Dittmer (1992–1995 as a Ph.D. student)
Claus S. Jensen (1993–1995 as a Ph.D. student)
Liang Jianming (1992–1994 as a visiting fellow)

9.4 Collaboration

Within the Group

Regarding domestic research collaboration, the DSS Group has been an integral part of the ODIN Group. Within ODIN, the participants did have labels indicating unit membership, but these have had no impact on the research work. However, Frank Jensen and Uffe Kjærulff were the most active with respect to system implementations. The work in ODIN has been proceeded through weekly meetings/seminars and monthly meetings where the status of the various research tasks have been reported. Every half-year, a brain storming has taken

place, where research tasks were taken up and closed, and where the participants committed themselves to the various tasks.

Within the Institute of Electronic Systems

The group has been engaged in a collaboration with the group for image analysis at MIBA. This started through the ESPRIT project Vision as Process (VAP), where Finn V. Jensen was a consultant, and it evolved to a more specific collaboration using Bayesian networks for scene analysis. Also, a small project on Bayesian networks and karyotyping was carried out.

With Other Danish Institutions

The ODIN Programme has been in close collaboration with the agricultural program, DINA (Danish Informatics Network in Agriculture). Various projects on specific decision support systems are formed, but no tangible results have yet been produced. Also, Finn V. Jensen has supervised the DINA Ph.D. student Allan Leck Jensen (thesis submitted in September 1995).

ODIN and Hugin Expert Ltd. have been in close contact during the entire period. In this way, Hugin staff have contributed to the development of methods, ODIN has had a good channel to "practice," and Hugin Expert has had the benefit of knowing first of the results obtained in ODIN. Obviously, Hugin Expert had no influence on the research tasks taken up in ODIN, and all results from ODIN are publicly available.

The DSS Group has been involved in a couple of project proposals for Danish industrial companies, but so far none of them have taken off.

Internationally

The Decision Support Systems Group considers itself a part of the international society of researchers in computerized uncertainty management. This is, for example, reflected in Section 9.6 "Services and Research-Related Activities," and in the involvement of the ESPRIT Basic research Action DRUMS.

The DSS Group has participated in the EC-supported research project ESPRIT BRA 6156-DRUMS II on Defeasible Reasoning and Uncertainty Management Systems II. The contribution of the group was concentrated on a subproject concerning the development of methods for integrating exact and approximate algorithms for reasoning in Bayesian networks. The subproject was run jointly with IIIA CEAB (Blanes, Spain) and DECSAI (University of Granada, Spain).

Members of the Decision Support Systems Group have participated in several DRUMS workshops, presenting ideas and results from on-going research. In 1993, Uffe Kjærulff participated in a workshop in Blanes, Spain, where he gave a

demonstration of HUGIN. In 1994, he participated in a workshop in Torremolinos, Spain, presenting preliminary work which was later published in [16].

Finn V. Jensen is contributing to and taking part in the editing of the Handbook of Defeasible Reasoning.

Furthermore, Uffe Kjærulff, Søren L. Dittmer, and Claus S. Jensen have had longer stays at various institutions in the USA.

During his Ph.D. study, Uffe Kjærulff was a visiting student at the Section of Medical Informatics at Stanford University, California, from April to October 1990. Collaboration on reasoning in dynamic, temporal Bayesian networks was initiated with dr. Paul Dagum. Some of the results of this work have been published in [10].

Claus S. Jensen visited the University of Chicago from September 1994 to February 1995, and developed and implemented methods for performing linkage analysis in genetic pedigrees. Linkage analysis is an algorithm used for the mapping of genes on the chromosome.

In 1992, Microsoft Inc. head-hunted three of the leading American researchers on normative systems, and they have now built up a group at the Microsoft Research Division. We are in close contact with this group. Søren L. Dittmer visited the Microsoft Research Group from July to November 1994 and developed tools for making optimal decisions in influence diagrams in close cooperation with his hosts. He was also involved in the application of Bayesian networks in end-user software products. Also, Eric Horvitz from Microsoft and Finn V. Jensen are program co-chairs for the next Conference on Uncertainty in Artificial Intelligence.

In the period, the ODIN Group has had a number of foreign visitors for stays of two weeks or more (we can not distinguish ODIN visitors from DSS visitors), and members of the ODIN Group have visited several foreign universities.

9.5 Ph.D. Projects

In the period 1991–1995 the DSS Group supervised five Ph.D. students, and one student has finished with supervison from outside the group. Three of the six Ph.D. studies have finished successfully within the period, one is expected to finish in 1996, one in 1997, and one in 1998.

HUGIN and MUNIN, Past, Present and Future

Name:	Kristian Grønborg Olesen
Education:	Cand.scient. (M.S.) in Computer Science and Mathematics, 1985
Duration:	August, 1986–March, 1992
Status:	Degree awarded March, 1992
Funding:	The Faculty of Engineering and Science, Aalborg University and the ESPRIT project P599
Advisor:	Finn V. Jensen

Kristian G. Olesen took part in the development of MUNIN, which was the first DSS-system based on a Bayesian network, and he played a central role in the development of the DSS-shell HUGIN. Also, he participated in the implementation and experimentation of the system for automatic adaptation of the model through incoming experience. His contributions are reported in the Ph.D. thesis.

Aspects of Efficiency Improvement in Bayesian Networks

Name:	Uffe Bro Kjærulff
Education:	Cand.polyt. (M.E.), 1986
Duration:	February, 1989–April, 1993
Status:	Degree awarded April, 1993
Funding:	Faculty of Engineering and Science, AAU, and ODIN
Advisor:	Steffen Lauritzen, Statistics Group at Aalborg University

Exact inference in Bayesian networks is generally NP-hard and has complexity determined by an underlying undirected graph. This graph must define a decomposable (hypergraph) cover of the independence graph of the Bayesian network. The choice of cover is essential to the efficiency of inference. A subproject dealt with developing algorithms for selecting optimal or near-optimal decomposable covers.

When exact inference is prohibitive, the exactitude of representation and/or inference must to be compromised. A second subproject focused on approximating the representation through altering the conditional independence properties of a network such that the complexity of inference decreases.

Domains involving repeated observations of a collection of random quantities arise in many fields of science. To cope with such dynamic systems using Bayesian

networks multiple instances of static networks must be interconnected. A third subproject developed methods for inference in dynamic Bayesian networks.

All three sub-projects contained both a theoretical and a practical component. The latter involved implementation and thorough evaluation of the methods developed.

A Probabilistic Model Based Decision Support System for Mildew Management in Winter Wheat

Name:	Allan Leck Jensen
Education:	Cand.scient. (M.S.) in Mathematics and Biology, 1991
Duration:	February, 1992–August, 1995
Status:	Degree awarded September, 1995
Funding:	DINA
Advisor:	Finn V. Jensen

The opinion that the environmental impact of agricultural production must be reduced has become widespread during the last decade, politically as well as publicly. As a consequence, action plans to reduce the nitrogene discharge as well as the pesticide consumption were passed in 1987 by the Danish government, and in 1991 the government decided on an action plan for a sustainable agriculture.

The intention of this project has been to construct a prototype of a DSS for farmers and the agricultural advisory service, which can give reliable recommendations of timing and dosing of mildew treatments in winter wheat cropping. To obtain reliable treatment recommendations, a Bayesian network for forecasting the consequences of treatment decisions have been constructed. The network model contains variables describing the important features of the crop, the disease, the fungicide, the field, the weather and the farmer, in order to predict the interactive, temporal development of the crop disease.

Recommendations from the DSS are computed from the Bayesian network by an expansion of the HUGIN method aimed at solving influence diagrams.

Implementation of Normative Systems for Multiprocessor Computers

Name:	Søren Lindequist Dittmer
Education:	Cand.polyt. (M.E.), 1993
Duration:	February, 1993–August, 1996
Status:	In progress
Funding:	The Faculty of Engineering and Science, Aalborg University
Advisor:	Finn V. Jensen

The development of methods for inference in Bayesian networks has led to an increase in the application areas for model-based decision support systems. The

latest development are facilities to solve decision problems, e.g., treatment planning, represented by influence diagrams comprising a utility function.

This project began by the involvement in the design and implementation of methods for solving decision problems, and has proceeded by a parallel version of the inference engine. The final phase of the project is the development of explanation facilities, which are necessary for the user to develop trust in a machine-based decision support system. Methods for explaining decisions made by influence diagrams, and designs for explanation interfaces are currently being explored.

Blocking Gibbs Sampling in Very Large Probabilistic Expert Systems with Emphasis on Genetic Applications

Name:	Claus Skaanning Jensen
Education:	Cand.polyt. 1993
Duration:	February, 1994–January, 1997 (expected)
Status:	In progress
Funding:	ODIN and the Department of Mathematics and Computer Science, Aalborg University
Advisor:	Uffe Kjærulff (and Finn V. Jensen)

Due to the NP-hardness of inference in Bayesian nets, it has been necessary to use approximate methods when dealing with large and complex nets. One such method is Gibbs sampling, that obtains dependent realizations from a complicated Bayesian net by sampling one component at a time, conditional on the remaining. One such component can consist of one or more variables from the net.

Single-site Gibbs sampling is a variant of Gibbs sampling, where only one variable is sampled at a time conditional on the remaining variables. This method is simple to implement and has been used extensively in research. It is, however, prone to very slow (or zero) convergence as the nets become more and more complex. However, if strongly correlated variables are sampled simultaneously, much faster convergence can be attained. This principle is used in blocking Gibbs sampling, which combines Gibbs sampling with exact local computations in junction trees. Blocking Gibbs at the same time attempts to minimize the time spent at simulating the components with exact local computations, and maximize the rate of convergence—two opposed criteria. The study is aimed at analyzing, optimizing and implementing this algorithm.

Gibbs sampling has been used very extensively in the area of genetics for the very computationally demanding methods of pedigree and linkage analysis. Here, the problems of slow (or zero) convergence have been present as in many other fields, and in addition, due to the near-logical conditional probability tables used in genetics, it is hard to determine when the induced Markov chain will be

irreducible. The study is also aimed at solving this problem, and additional problems experienced in genetics.

9.6 Service and Research-related Activities

Editorial Service

- Program co-chair for the *Twelfth Conference on Uncertainty in Artificial Intelligence*, Portland, Oregon 1996 (FVJ)
- Member of Review Board for the *International Journal of Applied Intelligence* (UK)

Conference and Workshop Organization

Program chair and workshop chair of the *Fourth Workshop on Normative Systems*, Aalborg 1994 (FVJ)

Program Committees

- *Tenth Conference on Uncertainty in Artificial Intelligence*, Seattle 1994 (FVJ)
- *Eleventh Conference on Uncertainty in Artificial Intelligence*, Montreal 1995 (UK)
- *IJCAI 1995* workshop *Building Probabilistic Networks*, Montreal 1995 (FVJ)
- *Fifth Workshop on Normative Systems*, Washington D.C. 1995 (FVJ)

Refereeing for Journals and External Refereeing for Conferences

- *Artificial Intelligence Journal* (FVJ)
- *Statistics and Computing* (FVJ, UK)
- *Journal of Artificial Intelligence Research* (FVJ, UK)
- *European Journal of Operational Research* (UK)
- *IEEE Transactions on Pattern Analysis and Machine Intelligence* (UK)
- *International Journal of Applied Intelligence* (UK)
- *Biometrika* (UK)
- *IEEE Transactions on Systems, Man, and Cybernetics* (UK)
- *Journal of the American Statistical Association* (UK)

- *Journal of Computational and Graphical Statistics* (UK)
- *Journal of the Royal Statistical Society (Series B)* (UK)
- *Nordic Journal of Computing* (FJ)
- *The Fourth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems* (IPMU), 1992 (FVJ)
- *The Ninth Conference on Uncertainty in Artificial Intelligence*, Washington D.C. 1993 (FVJ, UK, FJ)
- *The Tenth Conference on Uncertainty in Artificial Intelligence*, Seattle 1994 (UK)
- *International Joint Conference on Artificial Intelligence*, Montreal 1995 (FVJ)

Refereeing for Funding Organizations

- European Commission, ESPRIT III (FVJ)
- European Commission ESPRIT - long term IT research (FVJ)
- National Science Foundation, USA (FVJ)
- The Danish Ministry for Agriculture and Fishing (FVJ)

Other Academic Services for Foreign Institutions

- One week Ph.D. course on Bayesian networks, San Sebastian 1995 (FVJ)
- Ph.D. Board for Hung Xu, INRIA, Brussels, 1995 (FVJ)

9.7 The Group's Own Evaluation

Research in decision support have three components:

- development of tractable methods for DSS tasks
- efficient implementation of general methods
- construction of specific systems

All three components are interdisciplinary in nature. The development and implementation of tractable methods requires insight an creative work in computer science and mathematics as well as in philosophy. The construction of specific systems requires insight and imaginative work in computer science and psychology as well as in the domain in question.

Therefore the success of the DSS Group is not only dependent on the creativity and industry of the members of the group, but it is also dependent on the opportunities given for interdisciplinary work. For the DSS Group at Aalborg University, the situation has been excellent for collaboration with mathematicians and experts on specific domains (particularly agriculture). We find that the results of this collaboration are satisfactory, and it is our impression that the group has a good reputation in the field of computerized management of uncertainty.

However, there are some points where we find that improvements are needed.

Although we are in close collaboration with the agricultural research institutions in Denmark, we would like also to work on decision support systems for other areas. Particularly, we would welcome projects in the financial sector. So far, our attempts have not been successful.

Due to the interdisciplinary character of the research, we have realized that the collaboration in the group has suffered, and is now reduced to Ph.D. supervision.

Also, we have realized that while we have very good research collaboration with groups outside the computer science unit, we have no collaboration with other groups inside the unit. In the implementation of systems we have several times faced problems which to us looked appropriate for other groups in the unit, for example,

- knowledge acquisition and analysis of decision options seems appropriate for the Information Systems Group,
- the implementation on multiprocessor computers seems appropriate for the Computer and Formal Systems Groups,
- data mining (also called batch learning) seems appropriate for the Database Systems Group, and
- the construction of interfaces for specific DSS's as well as interface generators seems appropriate for the Programming Systems Group.

So far, the attempts to involve other groups from the unit in our work have not been successful. We do not blame the other groups: what from an outside point of view seems appropriate may from an inside point of view seem out of place. Also, in research you do not change tracks when you are succeeding with the direction in which you are working at the moment.

9.8 Research Plan 1996–2000

9.8.1 Research Tasks

In the coming five years we intend to focus our research efforts into the following four areas:

Applications

From the work with agricultural applications we have learnt that it is much more demanding than anticipated to construct useful decision support systems. It requires at least three enthusiastic parties: domain experts, DSS specialists, and domains users. We intend to follow a specific decision support system through all of its phases from idea to maintenance. The system may be an agricultural system, a vision system or a financial system. We will put much effort into selecting a project with all three parties present.

Use of Modern Information Technologies

The innovations in both multimedia and multiprocessor computers provide opportunities for constructing flexible and efficient decision support systems. We intend to pursue these opportunities.

More Flexible Inference Methods

The core of any decision support system is inference from evidence to beliefs and the expected outcomes of actions. The monstrous combinatorial explosion always lies and waits for detonation. Current inference methods have pushed the combinatorial explosion some steps away, but there is still a need for methods to push it further away. We will continue our work on designs where several exact and approximate methods can be combined in one inference.

Social Choice

Our work on decisions has so far been in connection with influence diagrams, where we have exploited the HUGIN method for solving them. We will now expand the scope to decision scenarios with several decision makers. They may be competing (game theory) or cooperating.

As mentioned earlier, the success of our efforts is heavily dependent on collaboration with other research groups. We intend to keep the collaboration with the Statistics Group here as well as with the Danish agricultural research institutions, and we will try to get in contact with other groups.

9.8.2 Organization and Funding

It is not possible at this stage to refine the research plans further than above. A determining factor is the amount of external funding, and we are at present in a situation where we can not foresee this amount of external funding. When the ODIN Programme stopped in March 1995, it continued in a revised form as a sub-program of DINA under the name DINA-Aalborg. DINA was given funding for one year, with the expectation that when the underlying PIFT Programme has been through some necessary transformations, DINA would be given funding for a longer span of years. However, the transformation of PIFT turned out to be an annihilation. This means that at present everything concerning external funding is very uncertain.

With respect to internal funding, we aim for a steady state situation with two permanent positions, one assistant professor, and three Ph.D. students.

Bibliography

Ph.D. Theses

[1] Jensen, A. L. *A Probabilistic Model Based Decision Support System for Mildew Management in Winter Wheat*, Department of Mathematics and Computer Science, Aalborg University, Denmark. DINA Research Report No. 39, 1995.

[2] Kjærulff, U. *Aspects of Efficiency Improvement in Bayesian Networks*, Department of Mathematics and Computer Science, Aalborg University, Denmark, 1993.

[3] Olesen, K. G. *HUGIN and MUNIN, Past, Present and Future*, Department of Mathematics and Computer Science, Aalborg University, Denmark, 1992.

Refereed Contributions to Books

[4] Dawid, A. P., Kjærulff, U. and Lauritzen, S. L. Hybrid propagation in junction trees, *in* B. Bouchon-Meunier, R. R. Yager and L. A. Zadeh (eds.), *Advances in Intelligent Computing*, Vol. 945 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 87–97, 1995.

[5] Jensen, F. V., Chamberlain, B., Nordahl, T. and Jensen, F. Analysis in HUGIN of data conflict, *in* P. P. Bonissone, M. Henrion, L. M. Kanal and J. F. Lemmer (eds.), *Uncertainty in Artificial Intelligence 6*, Elsevier Science Publishers B. V. (North-Holland), Amsterdam, pp. 519–528, 1991.

[6] Jensen, F. V. and Liang, J.-M. drHUGIN – a system for hypothesis driven data request, *in* A. Gammerman (ed.), *Probabilistic Reasoning and Bayesian Belief Networks*, Alfred Waller Ltd., pp. 109–124, 1995.

Refereed Journal Articles

[7] Andreassen, S., Jensen, F. V. and Olesen, K. G. Medical expert systems based on causal probabilistic networks, *International Journal of Biomedical Computation* **28**: 1–30, 1991.

[8] Jensen, C. S., Kong, A. and Kjærulff, U. Blocking-Gibbs sampling in very large probabilistic expert systems, *International Journal of Human-Computer Studies* **42**: 647–666, 1995.

[9] Kjærulff, U. Optimal decomposition of probabilistic networks by simulated annealing, *Statistics and Computing* **2**: 7–17, 1992.

[10] Kjærulff, U. dHugin: a computational system for dynamic time-sliced Bayesian networks, *International Journal of Forecasting* **11**: 89–111, 1995.

Conference on Uncertainty in Artificial Intelligence

[11] Jensen, F., Jensen, F. V. and Dittmer, S. From influence diagrams to junction trees, *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers, San Francisco, California, pp. 367–373, 1994.

[12] Jensen, F. V. Cautious propagation in Bayesian networks, *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers, San Francisco, California, pp. 323–328, 1995.

[13] Jensen, F. V. and Jensen, F. Optimal junction trees, *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers, San Francisco, California, pp. 360–366, 1994.

[14] Kjærulff, U. A computational scheme for reasoning in dynamic probabilistic networks, *Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers, San Francisco, California, pp. 121–129, 1992.

[15] Kjærulff, U. Reduction of computational complexity in Bayesian networks through removal of weak dependences, *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers, San Francisco, California, pp. 374–382, 1994.

[16] Kjærulff, U. HUGS: Combining exact inference and Gibbs sampling in junction trees, *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers, San Francisco, California, pp. 368–375, 1995.

[17] Olesen, K. G., Lauritzen, S. L. and Jensen, F. V. aHUGIN: A system creating adaptive causal probabilistic networks, *Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers, San Francisco, California, pp. 223–229, 1992.

Other Refereed Conference Articles

[18] Jensen, F. V. Calculation in HUGIN of probabilities for specific configurations — a trick with many applications, *Scandinavian Conference on Artificial Intelligence*, IOS Press, pp. 176–186, 1991.

[19] Jensen, F. V., Aldenryd, S. and Jensen, K. B. Sensitivity analysis in Bayesian networks, *Proceedings of the Third European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty* (ECSQARU'95), *Lecture Notes in Computer Science*, Springer-Verlag, 1995.

[20] Jensen, F. V., Christensen, H. I. and Nielsen, J. Bayesian methods for interpretation and control in multi-agent vision systems, *in* K. W. Bowyer (ed.), *Proceedings of Applications of Artificial Intelligence X: Machine Vision and Robotics*, SPIE — The International Society for Optical Engineering, pp. 536–548, 1992.

[21] Jensen, F. V. and Liang, J.-M. drHUGIN - a system for value of information in Bayesian networks, *Proceedings of the Fifth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*, Cité Internationale Universitaire, Paris, pp. 178–183, 1994.

[22] Liang, J.-M., Christensen, H. I. and Jensen, F. V. Qualitative recognition using Bayesian reasoning, *in* Gelsema et al. (eds.), *Proceedings of Pattern Recognition in Practice IV*, Vlieland, Holland, 1994.

Research Reports

[23] Jensen, F. Implementation aspects of various propagation algorithms in Hugin, *Research Report R-94-2014*, Department of Mathematics and Computer Science, Aalborg University, Denmark, 1994.

[24] Kjærulff, U. Approximation of Bayesian networks through edge removals, *Research Report IR-93-2007*, Department of Mathematics and Computer Science, Aalborg University, Denmark, 1993.

Software

[25] cHUGIN; a system for editing of and belief propagation in Bayesian networks with both discrete and conditional Gaussian distributed variables (K. G. Olesen, F. Jensen).

[26] chHUGIN; a system for editing of and belief propagation in chain graphs (F. Jensen).

[27] dHUGIN; a system dealing with time series of Bayesian networks (U. Kjærulff).

[28] drHUGIN; a system for data request (J.-M. Liang, F. V. Jensen).

[29] extHUGIN; facilities for propagation of most probable configuration, random configuration and fast retraction (F. Jensen).

[30] Blocking Gibbs; a system for doing inference in very large and complex Bayesian networks (C. S. Jensen). It uses a refined version of Gibbs sampling.

[31] HTG; Hugin Table Generator (K. G. Olesen).

[32] hc+; extended Hugin compiler capable of compiling arbitrarily complex networks using simulation (U. Kjærulff).

[33] HUGS; Hugin extended with capabilities for combining exact and approximate computation (U. Kjærulff).

[34] gt; a system for finding (sub)optimal triangulations of Bayesian networks (U. Kjærulff).

Manuals

[35] Kjærulff, U. dHUGIN API reference manual, *Technical Report IR-93-2004*, Department of Mathematics and Computer Science, Aalborg University, Denmark, 1993.

[36] Kjærulff, U. User's guide to `dhugin`, *Technical Report IR-93-2005*, Department of Mathematics and Computer Science, Aalborg University, Denmark, 1993.

[37] Kjærulff, U. User's Manual to `gt`, Department of Mathematics and Computer Science, Aalborg University, Denmark, 1993.

Part III

Summary in Danish

Chapter 10

Resumé

Ifølge retningslinier fra det Teknisk-naturvidenskabelige Fakultet ved Aalborg Universitet skal dets forskningsenhder evalueres hver femte år. Denne rapport dokumenterer i overensstemmelse hermed den anden forskningsevaluering af forskningsenheden Datalogi og dækker perioden fra 1991 til 1995.

Ifølge Fakultetets retningslinier har forskningsevalueringen tre formål, idet den skal

a) vurdere, om der er tilfredsstillende overensstemmelse mellem de tildelte interne og eksterne forskningsressourcer og det udførte forskningsarbejde,

b) vurdere om der er rimelig sammenhæng mellem forskningsenhedens mål for den forskningsmæssige indsats, herunder som den fremgår af fakultetets langsigtede faglige planlægning, og den gennemførte forskning og

c) rådgive forskningsenheden om dens fremtidige indsats og forskningsorganisering.

Forskningsenheden tilføjede et ekstra punkt, som tildels understreger visse aspekter af Fakultetets formål:

> Inden for forskningsenheden skal evalueringen, processen såvel som den endelige rapport, konstruktivt hjælpe medarbejderne med at evaluere og forbedre deres effektfuldhed som forskere, gruppeledere og som administratorer af forskning på afdelingsniveau. Specielt er evalueringen en anledning til at vurdere afdelingens forskningspolitik.

Processen, der ledte frem til denne rapport, strakte sig over halvandet år og beskrives kort i det følgende.

Evalueringsprocessen tog sin start i begyndelsen af 1995. En af de vigtige, tidlige aktiviteter var at etablere et evalueringsudvalg. Professor Kim G. Larsen accepterede at være lokal repræsentant i udvalget (lokale medlemmer har ingen stemmeret), og det var med stor tilfredshed i forskningsenheden, at professor Sture Hägglund fra Linköping Universitet og professor Stig Skelboe fra

Københavns Universitet også accepterede at deltage i udvalget. Det var vores fornemmelse, at dette udvalg af anerkendte og erfarne seniorforskere ville være istand til at dække de forskelligartede forskningsområder, som Datalogi rummer, og ville kunne udarbejde en insigtsfuld evaluering.

I løbet af foråret 1995 blev der også gennemført en række møder, hvor forskellige af medarbejdere præsenterede forskellige perspektiver på forskningsevaluering. Mere specifikt blev medarbejderene bedt om at beskrive de kriterier, de ville anvende, hvis de selv skulle evaluere deres egen forskning. Disse møder gav indblik i en mangfoldighed af opfattelser af forskning, og de var med til at gøre forskningsevaluering til et diskussionsemne i Datalogi.

Endelig vedtog vi i foråret 1995 den overordnede struktur for evalueringsprocessen og for denne rapport. Det var en stor hjælp, at vi allerede havde gennemført én evaluering.

Rapporten skulle have to hoveddele, nemlig Datalogis beskrivelse af sin forskning og så selve udvalgets evaluering af Datalogi. Forskningsbeskrivelsen skulle struktureres, så der var et kapitel for Datalogi generelt og et for hver af de seks forskningsgrupper i Datalogi.

Der blev også vedtaget overordnede skabeloner for disse kapitler. herefter var det op til ledelsen og hvor forskningsgruppe at udfylde skabelonerne. Formålet med dette design var at tillade de fagligt forskellige grupper en vis grad af frihed til at rapportere deres forskning samtidig med at resultatet stadig blev en tilpas homogen rapport.

I begyndelsen af december 1995 blev en midlertidig udgave, på godt og vel 170 sider, af forskningsbeskrivelsen (den anden del af denne rapport) sendt til evalueringsudvalget sammen med et udvalg af repræsentativer videnskabelige publikationer fra grupperne.

Evalueringsprocessen kulminerede i januar 1996 med at der afholdtes et todags internat, hvor alle forskningsmedarbejder, herunder Ph.D. studenter, og administrativt personale var inviteret. Den første dag (evalueringsudvalget ankom dagen før for at forberede sig samlet) præsenterede hver forskningsgruppe sin forskning, og der var lejlighed til at panelet og andre kunne stille spørgsmål til præsentationerne og det skrevne materiale. Den første halvdel af den anden dag arbejdede udvalget separat. Alle andre deltog i forskellige aktiviteter omkring temaet forskningsevaluering. Om eftermiddagen fremlagde udvalget sin evaluering og besvarede spørgsmål og modtog kommentarer under den efterfølgende diskussion.

Efter internatet færdiggjorde udvalget sin evaluering. Samtidig producerede ledelsen og forskningsgrupperne reviderede udgaver af deres bidrag til rapporten, idet de baseret på kommentarer fra internatet typisk foretog relativt mindre forbedringer af de foreløbige beskrivelser. Det er disse reviderede udgaver, der efter nogen redigering fremstår som kapitler i rapportens anden del. Derfor er evalueringen i den første del og beskrivelserne i den anden del ikke fuldstændigt synkroniserede.

Den foreliggende rapport har tre dele. Den første del rummer selve evalueringen af Datalogi. Denne del beskriver først den formelle ramme for for evalueringen og præsenterer derefter udvalgets evaluering (konklusionerne herfra følger på dansk i næste kapitel). Som omtalt ovenfor har den anden del et kapitel for forskningsenhden i sin helhed og et for hver af de seks forskningsgrupper i enheden. Det første kapitel beskriver generelle forhold ved Datalogi. Det dækker Datalogis baggrund og organisatoriske kontekst samt den overordnede organisering af enhedens forskning. Desuden præsenteres forskningsplanen for perioden 1991 til 1995, og de tilgængelige ressourcer for gennemførelsen af planen beskrives. Endelig evalueres den overordnede administration og organisering af forskinngen i enheden, og en plan for 1996 til 2000 præsenteres. De enkelte forskningsgruppers kapitler sammenfatter gruppernes forskningsaktiviter og -resultater og beskriver gruppernes organisering og bemanding. Så følger beskrivelser af forskningssamarbejde, Ph.D. projekter, og service- og forskningsrelaterede aktiviteter. Endelig præsenteres selvevalueringer og planer for den næste periode. Rapportens sidste del er denne opsummering på dansk.

Chapter 11

Evalueringsudvalgets konklusion

Forskningsenheden Datalogi ved Aalborg universitet har dokumenteret fuldt tilfredsstillende forskningsaktiviteter i evalueringsperioden fra 1991 til 1995. Alle forskningsenhedens seks forskningsgrupper udviser fra god til høj performance med hensyn til både kvantiteten og kvaliteten af de publicerede resultater. Ph.D. uddannelsen synes at fungere meget effektivt, og en stor del af de studerende opnår Ph.D. graden inden for den afsatte tid. Samspillet mellem forskning og uddannelse er mindre godt dokumenteret, men det synes at være velfungerende. Til trods for at mængden af formaliseret internationalt samarbejde ikke er specielt høj, findes der gode internationale kontakter på flere områder. Antallet af kontakter til industrien og af andre kontakter er tilfredsstillende, men dette antal kan godt forøges.

Planerne for den næste femårsperiode kunne i nogle tilfælde have været mere visionære hvad angår de langsigtede forskningsmål, men de er generelt velafbalancerede, realistiske og relevante. Forskningsenheden Datalogi er stadig i vækstfasen og består af forholdsvis unge og entusiastiske forskere, der forfølger emner tæt på forskningsfronten. Den relativt langsomme og omhyggelige opbygning af personalet har resulteret i en god aldersfordeling og i velfungerende aktiviteter. Dog er nogle grupper underbemandet i forhold til deres forskning, og den konsekvente undervisningsoverbelastning forværrer situationen.

Selve evalueringsprocessen har efter vores mening været en success, og evalueringsudvalget er meget tilfreds med det udleverede materiale og præsentationerne ved evalueringsinternatet.

Sture Hägglund	Kim G. Larsen	Stig Skelboe
professor	professor	professor
Linköping Universitet	Aalborg Universitet	Københavns Universitet