GRADUATE COURSES 93/94

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE

LINKÖPING UNIVERSITY

Institutionen för datavetenskap • Universitetet och Tekniska högskolan i Linköping • 581 83 Linköping
Forskn.sekr. Lillemor Wallgren • Telefon: 013/28 14 80 • Telefax: 013/14 22 31 • Elektronisk post: lew@ida.liu.se

Contents

General Information for the Graduate Program in Computer Science 1994 - 1995

This program describes the graduate courses planned for the academic year 1994 - 1995 within the areas covered by the Department of Computer and Information Science (Institutionen för datavetenskap, IDA): Computer Science, Computer Systems, Library and Information Science, Economic Information Systems, Computational Linguistics, Engineering Information Systems and 'Informationssystemutveckling'. Other courses and seminar series may also be presented during the year, particularly intensive courses given by visiting researchers.

The program describes the following types of courses:

Graduate Courses at the Department of Computer and Information Science
Graduate Courses in the School of Engineering
Recommended Masters courses
Laboratory-Oriented Courses and Activities
Research Organization and Laboratories
Faculty

A comprehensive list of all graduate courses in the School of Engineering will be issued by the Planning Office in the fall.

Any changes in the course information will be announced by e-mail, in the IDA-Kuriren, posted on bulletin boards at IDA or through contact with those graduate students enrolled in a particular course.

Although not mandatory, it is strongly recommended that all graduate students participate in the following activities:

Introduction meeting of graduate students: Tuesday, August 30, at 13.15 in the IDA Seminar room, 1st floor, E-building.

The main seminar series on Tuesdays at 13.15.
Seminars are announced by e-mail, in the IDA-Kuriren, posted in the seminar file and occasionally by special announcement. The seminars are usually held in the IDA seminar room, 1st floor, E-block (see the inside back cover).

IDA-fika (Departmental coffee-breaks).
Current information, short presentations of new arrivals and visitors in the department , descriptions of trips and conferences etc. are given **every Tuesday at 12.15 in the coffee area, E-block, 1st floor.**

Further information concerning the contents of this program can be obtained from Per-Olof Fjällström, Director of Graduate Studies, tel. 013 28 24 12 and Lillemor Wallgren, Administrator of Graduate Studies, tel. 013 28 14 80, or for a particular course from the person responsible for the course within the Department of Computer and Information Science, Linköping University, S-581 83 Linköping.

Linköping, June 27, 1994
Lillemor Wallgren

VLSI System Design

Recommended: ***Lectures:*** ***24 h***

For graduate students in Computer Systems
and Computer Science

Goals:

The goal of the course is to present for students basic synthesis methods used for digital hardware on the register transfer level and higher.

Prerequisites:

Knowledge about digital hardware and programming languages.

Organization:

The course will consist of lectures discussing synthesis methods used at logic, register and system level.

Contents:

1. Introduction to synthesis
2. Hardware description languages and their relevance to synthesis
3. Logic-level synthesis
4. Technology mapping
5. Register-transfer synthesis
6. High-level synthesis
7. System-level synthesis
8. Formal methods for synthesis
9. Synthesis related aspects of simulation and testing

Literature:

P. Michel, U. Lauther, P. Duzy, *The Synthesis Approach to Digital Systems Design*, Kluwer Academic Publisher, 1992.

Teachers:

Krzysztof Kuchcinski, Zebo Peng

Schedule:

October- December

Credit:

3 points

Distributed Databases

Recommended for: Graduate students.

***Lectures:* 16 h**

The course last ran:
New course.

Goals:
To provide in-depth knowledge of distributed data management technology. Traditional distributed relational databases will be covered as well as more recent developments for object-oriented distributed databases.

Prerequisites:
Undergraduate courses in computer science, incl. TDDA 38.

Organization:
Seminars covering the core material of the course.

Contents:
- Goals and promises of traditional distributed data management, and their fullfillments.
- Important remaining research issues.
- Replication.
- Fragmentation.
- Reliability.
- Distributed transaction management.
- Distributed query processing.
- Impact of new technologies on DDBMSs.
- Object distribution.
- Heterogeneous databases.
- Multi-databases.

Literature:
M.T. Ozsu, P. Valduriez: *Principles of distributed database systems* + articles.

Teacher:
Tore Risch.

Examiner:
Tore Risch.

Schedule:
September - October, 1993.

Examination:
Written exam.

Credit:
2 points

Parallel Programming and Compilation Techniques

Recommended for: ***Lectures + seminares :*** **40 h**

Graduate students.

The course last ran:

About 2/3 of the course material has been covered by the previous courses *Compiling for Parallelism* (1990/91) and *Parallel Execution Models* (1991/92).

Goals:

To give an understanding of different parallel programming models and compilation techniques for serveral kinds of programming languages.

Prerequisites:

The undergraduate compiler courses Compiler Construction (TDDA37) or Compilers and Interpreters (TDDA28), or equivalent. The process programming course.

Contents:

Parallel execution models, languages, compilation techniques etc. For example: definition of parallel computing, measures of performace, parallel processors, shared-memory parallel programming, distributed-memory parallel programming, object-oriented parallel programming, data parallel programming, functional dataflow programming, scheduling parallel programs, loop scheduling, parallelizing serial programs, parallel programming support environments. Practical programming exercises on a parallel machine gives 1-2 extra points.

Literature:

Ted Lewis, Hesham El-Rewini: Introduction to Parallel Computing, Prentice Hall, 1992.
Hans Zima, Barbara Chapman: Supercompilers for Parallel and Vector Computers, ACM Press, 1991.

Teachers:

Peter Fritzson and invited lecturers.

Examiner:

Peter Fritzson.

Schedule:

September - November.

Examination:

Written examination.
Implementation project gives extra 1-2 points.

Credit:

5 (+1 - 2) points.

Generalized knowledge representation systems

Recommended for: **Lectures: 16-24h**

Students of Computer Science and Computational Linguistics.

The course last ran:

New course.

Goals:

(1) To identify and compare trends in different, though related areas of knowledge representation, such as the (typed) feature structure formalisms used in natural-language processing, the definite clause grammars used in logic programming, and the description logics used in AI; (2) To give an overview of research and systems that attempt to generalize results across one or more of these areas; (3) To provide a forum for discussion of current research at IDA on the mentioned topics.

Prerequisites:

A thorough knowledge of one or more of the following areas: logic programming, knowledge representation, logic, unification-based grammars.

Organization:

The course is organized in seminar form. Each participant is expected to present (or discuss) some material of relevance to the course, e.g. some chapter of the course book, some articles or some topic related to his/her own research.

Contents:

Partly to be decided by participants. For a start some topics covered by the course book: types and inheritance; feature structures, unification and subsumption; recursive and relational type constraints. Possible extensions include role structures, negation, defaults. Applications to feature-based grammars, definite clause programming and description logics.

Literature:

Bob Carpenter: The Logic of Typed Feature Structures. Cambridge Tracts in Theoretical Computer Science 32. Further literature to be decided on later.

Teachers:

Participants.

Examiner:

Lars Ahrenberg.

Schedule:

October onwards.

Examination:

Two contributions to the course in the form of a presentation or a prepared comment on some article(s).

Credit:

3 points.

Non-Monotonic Reasoning

Recommended for:	***Lectures:***	24 hours
Computer Science and Systems students.	Practical exercises:	6 hours

The course last run:
Fall 1991.

Goals:
The purpose of the course is to present both practical and theoretical aspects of non-monotonic reasoning and to explain its relationship to computer science, particularly AI, deductive databases and logic programming.

Prerequisites:
Introduction to standard logic. Familiarity with modal logic and three-valued logic is recommended but not necessary.

Organization:
12 two-hour lectures, twice a week. Independently, 6 hours of the laboratory work involving practical exercises.

Contents:
1. Foundations of non-monotonic reasoning.
2. Approaches to non-monotonic reasoning.
3. Non-monotonic formalisms:
 (i) Modal non-monotonic logics
 (ii) Default logic
 (iii)Circumscriptive logics
 (iv)Closed-World Assumption formalisms
 (v) Preferential entailment approach

Literature:
Lukaszewicz, W.: *Non-Monotonic Reasoning. A Formalization of Comonsense Reasoning*, Ellis Horwood , 1990.

Teacher:
Witold Lukaszewicz.

Examiner:
Witold Lukaszewicz.

Schedule:
September - October.

Examination:
There will be a set of obligatory exercises. Each student must submit written solutions to these exercises and be prepared to orally present his/hers solutions.

Credit:
4 points

Machine Learning

Recommended for: **_Lectures:_ 15 h**
Graduate students in Computer Science and Systems.

The course last ran:
1990.

Goals:
The course concentrates on inductive learning techniques and draws on experience from the automated knowledge aquisition domain. The inductive learning paradigm is examined with regard to the different approaches to concept learning, practical work using Quilan’s ID3 system and the problems associated with the approach.

Prerequisites:
Some experience with AI programming would be helpful.

Organization:
An intensive course: 5 lectures, 3 hours each. Moreover some laboratory exercises.

Contents:
- Introduction to machine learning
- Overview of inductive learning methods
- Attribute-based learning systems
- Practical work with Assistant Professional Evaluations of programmer’s performance
- Logic-based knowledge representations
- Relation learning systems - Inductive logic programming
- The role of inductive learning in knowledge acquisition

Literature:
A set of articles will be distributed to the participants of the course.

Teacher:
Nada Lavrac.

Examiner:
Nada Lavrac.

Schedule:
November 8, 9, 10, 16 and 17.

Examination:
Assignments requiring some practical experience with the ASSISTANT machine learning system.

Credit:
3 points

Introduction to Research Methodology in Computer Science

Recommended for: ***Lectures:*****16 h**

New graduate students.

The course last ran:

Fall 1992.

Goals:

To give an introduction to the philosophy of science, the special characteristics of computer science research and to discuss practical aspects of graduate studies and scientific activities.

Prerequisites:

None.

Organization:

Lectures and seminars.

Contents:

Introduction to the philosophy of sience. Scientific writing, publication and information retrieval. Science and technology, methodological issues. Survival strategies for young scientists. Project management.
Research paradigms in computer science.

Literature:

Chalmers: *What is this thing called science*?
Sindermann: *Survival strategies for young scientists.*
Lecture Notes.

Teachers:

Sture Hägglund, et al.

Examiner:

Sture Hägglund.

Schedule:

September - November.

Examination:

Written examination and seminar activity.

Credit:

2 points

Documents: Subject Analysis and Description

Recommended for: ***Lectures: 24 h***
Graduate students.

The course last ran:
New Course.

Goals:
To give an understanding of the issues in analyzing and describing the subject/content of “documents”, e.g by indexing and classification, and current approaches to solving them.

Prerequisites:
None.

Organization:
12 two-hour lectures discussing material previously assigned for study.

Contents:
Introduction: subjects, content, aboutness.
The subjects of different types of “documents”.
Analysis and description, approaches, methods and tools.
Depth of analysis and extent of description.
Approaches to automating the analysis.
Organizing subjects.
Controlled vocabularies vs. other approaches, e.g. “free text”, clustering, links.
Tools for vocabulary control: Classifications, subject heading lists, thesauri.
Review of common systems for knowledge organization.

Literature:
Articles and reports handed out during the course.

Teachers:
Roland Hjerppe, possibly some invited lecturers.

Examiner:
Roland Hjerppe.

Schedule:
October - December.

Examination:
A written report on a chosen topic.

Credit:
3 points.

Cognitive Science

Recommended for: ***Lectures:*** **28 h**

Graduate students from IDA, Tema, IPP and other
Cognitive Science oriented departments.

The course last ran:

New Course.

Goals:

To give a historical background of Cognitive Science as the interdisciplinary study of cogniton. To present some current developments and applications.

Prerequisites:

Basic knowledge in one sub-discipline in cognitive science; Cognitive psychology, AI, philosophy of mind, linguistics, anthropology, and neuroscience.

Organization:

An introductory part giving an overview and context, followed by seminars where course participants are active in analyzing current developments and applications.

Contents:

Historical background
Overview of Philosophy of mind, Cognitive psychology, Artificial Intelligence, Linguistics, Anthropology, Neuroscience in relation to the overreaching aims of cognitive science.
Emerging new paradigms; e.g. situated cognition, and sub-symbolic cognition.
Applied cognitive science.

Literature:

Howard Gardner: The Mind’s New Science, A History of the Cognitive Revolution. New York: Basic Books, 1985.
Selected papers

Teachers:

Yvonne Waern, Nils Dahlbäck

Examiner:

Yvonne Waern, Nils Dahlbäck

Schedule:

September - December.

Examination:

Seminar presentation, written term paper

Credit:

2 + 1 points

Ekonomisk styrning

Recommended for: **Lectures:** **ca 35 h**

Forskare med baskunskaper i redovisning, kostnadsintäktsanalys och organistion.

The course last ran:
Ny kurs.

Goals:
Gemensamt utforska ekonomisk informations potential som styrmedel internt i organisationer, med särskild betoning på målstyrning och decentralisering.

Prerequisites:
Inga formella.

Organization:
5 heldagars undervisning under IDA/EIS:s forskarveckor.

Contents:
Ekonomisk information som styrmedel i förhållande till organsationens struktur och uppgift. Lönsamhets- och prestationsmätning. Mål: krav på ekonomiska resultat. Planeringsprocesser och internmarknad. Organisation av den ekonomiska styrningen: controllers, stödsystem.

Literature:
Böcker av Johansson-Östman, Samuelson (red) och (prel.) Anthony-Dearden-Govindarajan; artiklar och enstaka kapitel ur andra skrifter, delvis i form av individuella referatuppdrag.

Teacher:
Nils-Göran Olve.

Examiner:
Birger Rapp.

Schedule:
Augusti - december.

Examination:
Presentationer och uppsatser under kursens lopp.

Credit:
5 poäng.

Kunskapsutveckling - teori, projektering, genomförande

Recommended for: ***Lectures:*** **24 h**
Ekonomiska informationssystem, systemutveckling; samt andra
med intresse för kunskapsteori och forskningsmetodik.

The course last ran:
92/93. Kursen har tidigare givits under namn ”Kunskapsutveckling om informationssystem”.

Goals:
Ge översikt över kunskapsteoretiska och forskningsmetodologiska grunder för empiriska studier, samt öka förmågan att planera och genomföra sådana studier.

Prerequisites:
Inga särskilda krav.

Organization:
Delvis samläsning med kursen Samhällsvetenskaplig metodik, SVL3. Del 1 av kursen innebär utförande av kunskapsprojektering för forskningsuppgift. Frivillig del 2 kan genomföras som mindre empirisk studie.

Contents:
Kursen fokuserar forskningsarbete som kunskapsutveckling. Kunskapsteori; olika kunskapsformer, olika kunskapsteoretiska skolor. Metodik för kunskapsprojektering (planering av forskningsarbete). Orientering om olika samhällsvetenskapliga empiriska metoder (intervju, observation, källgranskning). Väsentligt moment i kursen är inlärning och tillämpning av en metod för strukturerad och reflekterad forskningsplanering (kunskapsprojektering).

Literature:
Patel, Tebelius: Grundbok i forskningsmetodik, Studentlitteratur.
Goldkuhl: Kunskapande, kompendium.
Artikelkompendium i vetenskapsteori.

Teacher:
Göran Goldkuhl.

Examiner:
Göran Goldkuhl.

Schedule:
September-november 93.

Examination:
Skriftliga rapporter. Seminarium.

Credit:
3 poäng (del 1) + ca 3 poäng (del 2).

Metodik för verksamhetsanalys & informationsanalys

Recommended for: ***Lectures:*** **24 h**

Ekonomiska informationssystem, systemutveckling.
Forskarstuderande i andra ämnen som önskar orientering om systemutveckling och systemutvecklingsmetodik.

The course last ran:

91/92. Del av kursen har tidigare givits under namn ”Verksamhets- & informationsbehovsanalys”.

Goals:

Ge introduktion till verksamhetsinriktad teori och metodik för systemutveckling.

Prerequisites:

Inga särskilda krav. Grundkursen kan ej läsas av studerande som läst motsvarande kursmoment på grundutbildningsnivå (SVL, ADB). Fördjupningsdelen är dock öppen även för de som läst ”Verksamhets- & informationsbehovsanalys” på doktorand- eller grundutbildningsnivå.

Organization:

Koncentrerade föreläsningar. Eget arbete med tillämpningsuppgifter. Seminarier.

Contents:

Grundkurs: Informationssystem i verksamheter. Verksamhetsinriktad systemutveckling. Betydelse av systemutvecklingsmetodik. Metodik för verksamhets- & informationsbehovsanalys (VIBA/SIM-metoden som exempel på en Verksamhetsinriktad metod). Verksamhetsanalys, analys och beskrivning av informationssystem ur verksamhetsperspektiv, informationsstrukturering, behandlingsanalys, effektanalys.
Fördjupningsdel: Översikt och orientering om olika metoder för verksamhetsanalys och informationsanalys. Komparativ analys. Teoretisering och praktisk tillämpning.

Literature:

Goldkuhl: Datasystem och verksamhetsutveckling.
Goldkuhl: Kontextuell verksamhetsanalys med handlingsgrafer.
Andersen: Systemutveckling, Studentlitteratur.
Artiklar.

Teacher/Examiner:

Göran Goldkuhl

Schedule:

November - december.

Examination:

Skriftliga rapporter. Seminarium.

Credit

3 p (grundkurs) + ca 3 p (fördjupningsdel).

Vetenskaplig dialog om informationssystem

Recommended for: ***Seminars:*** **24 h**

Kursen vänder sig till forskarstuderande med inriktning mot informationssystem.

The course last ran:

Detta är en ny kurs.

Goals:

Seminariekurs som syftar till att stödja avhandlingsprojekt (lic eller dr) eller andra större forskningsarbeten. Kursen har fokus på presentation, granskning och kommunikation avseende forskningsprocesser.

Prerequisites:

Kursen är i första hand avsedd för forskarstuderande, som arbetar med lic.- eller dr.-avhandling.

Organization/Examination:

Kursen består av återkommande seminarier och eget arbete. Skriftlig redovisning (i form av PM eller liknande) av pågående avhandlingsarbete (kan vara i olika faser; inledande planering, aktivt genomförande, avslutande skrivfas). (Respondentroll). Kritisk granskning av respondentrapport. Skriftlig kommentering. (Opponentroll). Dialog på seminarium för att uppnå kunskapsutvecklande synteser. Flera respondentrapporter kan framställas under seminariekursen.

Contents:

- Skriftlig/muntlig presentation av planerade/pågående forskningsarbeten.
- Granskning av forskningsarbeten; kritisk analys, skriftlig kommentering.
- Vetenskapligt samtal och argumentation utifrån ideal om genuin kommunikation.

Seminariekursen kommer att ha insprängda teoriavsnitt avseende argumentation, dialog, kommunikationsideal, kritik, vetenskaplig självreflektion, diskursiv och rekonstruktiv rationalitet.

Literature:

Utvalda artiklar.

Teacher:

Göran Goldkuhl.

Examiner:

Göran Goldkuhl.

Schedule:

Börjar september -93; pågår hela läsåret.

Credit:

Kurspoäng ges i princip ej för skrivande av respondentrapporter (ingår i avhandlingsarbete). Poäng ges för kritisk granskning (opponentrapport) och deltagande på seminarier. 1 - 3 p beroende på insats (opponent); 1 p för aktiv närvaro (observatörsroll).

General Graduate Courses 93/94

Real-Time Systems

Recommended for: ***Lectures:*** **24 h**

Computer science and Systems graduate students..

The course last ran:

This course last run fall 1991.

Goals:

The course will give a survey of different research issues and state of the art in the design of real time systems.

Prerequisites:

Processprogrammering TDDA 21.
Basic knowledge in control teory and operating systems.

Organization:

The course will consist of 5 lectures and 7 seminars. The participants are expected to give presentations during the seminars. .

Contents:

- Specification and verification – inclusion of time metrics
- Real-time scheduling theory
- Real-time operating system
- Distributed real-time databases
- Real-time symbolic systems
- Architectures and models issues

Literature:

S-T. Levi, A.k. Agrawala: Real Time systems design.

Teacher:

Anders Törne.

Examiner:

Anders Törne.

Schedule:

Spring -94.

Examination:

Seminar presentation, written summaries.
Term paper on chosen subject gives extra points

Credit:

3 p (+2 p)

Principles of Modern Database Systems

Recommended for: ***Lectures:*** **32 h**

Graduate students.

The course last ran:

Fall -92.

Goals:

To present the fundamental concepts, theories and realizations of modern database technology and systems, with a concentration on modern Object-Oriented (OO) database systems, Query Processing, and Active Databases.

Prerequisites:

Undergraduate courses in computer science. Basic database course preferred, but not required.

Organization:

Lectures and seminars covering the core material of the course. Extra points for programming project or seminar papers.

Contents:

- Overview Traditional DBMSs.
- DBMS architectures.
- Data Models (e.g. relational, OO, functional).
- Object Data Management Concepts.
- Query processing (OO and relational).
- Concurrency and Recovery.
- Versioning.
- Database Performance and Benchmarks.
- Deductive Databases.
- Active Databases.
- Temporal Databases.
- Main-memory Databases.

Literature:

R.G.G. Catell: Object Data Management (prel.) + handouts.

Teacher:

Tore Risch

Examiner:

Tore Risch

Schedule:

Spring -94

Examination:

Written exam, Seminar Papers, and Small Programming Project

Credit:

4 + 2 points

Principles of Automated Action Planning

Recommended for:
Graduate students.

Lectures: **24 - 28 h Part I**
6- 8 h Part II

The course last ran:
New course. Replaces Knowledge Based Planning, D Driankov, J Maleki, spring 1991.

Goals:
Knowledge of the major approaches to automated action planning and their underlying theoretical principles.

Prerequisites:
Discrete math and some knowledge of logic and complexity theory.

Organization:
Divided into two parts:
Part I: Lectures + homework (incl. some laborations).
Part II: Assignments to groups of 2-3 students w. presentations.

Contents:
Focus on search-based planning (STRIPS school) rather than logic-based.
Focus on foundations and theoretical principles rather than on implemented systems.

Literature:
Articles and probably a compendium functioning as a ’skeleton’ for the articles.

Teachers:
Christer Bäckström (+ possibly others).

Examiner:
Christer Bäckström.

Schedule:
Spring -94.

Examination:
Part I: Obligatory homework.
Part II: Written + oral presentation by groups of 2-3 students.

Credit:
3 +1 p (maybe 3+2 p).

Applied Rewriting

Recommended for: ***Lectures:*** **28 h**

Graduate students in Computer Science and Systems.

The course last ran:
New course.

Goals:
To discuss relations between different rewriting formalisms.

Prerequisites:
Formal languages.

Organization:
Lectures and homework.

Contents:

- Chomsky hierarchy
- Two-level Grammars
- Attribute Grammars
- Lambda calculus
- Combinatory logic
- Term-rewriting
- Axiomatic systems (proof theory, type theory etc)

Literature:
Lecture notes (to be developed).

Teachers:
Jan Maluszynski and Ulf Nilsson.

Examiner:
Jan Maluszynski.

Schedule:
Late spring -94.

Examination:
Homework and presentation.

Credit: **4 points.**

Natural Language Semantics

Recommended for: ***Lectures:*** **36 h**
All interested parties.

The course last ran:
New Course.

Goals:
Mastery of the basic techniques of conceptual and semantic analysis for the processing (understanding and generation) of texts in natural language.

Prerequisites:
None

Organization:
Lectures and Discussions covering the course literature.

Contents:
Lexical Semantics: Word meanings - definitions, classifications, semantic fields, synonomy, antymony, metaphor, reference, predication, etc.
Sentence Semantics: Truth-conditional semantics, presupposition, conjunctions, modalities, tense, aspect, moods, etc.
Discourse Semantics:Discourse Relations, Rhetorical Relations, Coherence, Cohesion, Relevance, Discourse Genre, etc.

Literature:
Allwood, J. & Andersson L-G. (1992) GULING I Semantik
Hurford, J. & Heasley, B. *(1983) Semantics a Coursebook*
Lyons, J. (1977) *Semantics* Vols I & II (or something equivalent)
Articles and selected readings

Teachers:
Richard Hirsch
Ivan Rankin

Examiner:
Richard Hirsch

Schedule:
Spring term 1994: January - May

Examination:
Take home essay exam for 4 points with extra 1 point for course paper.

Credit:
4 + 1 point

Architectures of Intelligent Autonomous Agents

Recommended for:
Computer Science and Systems students.

Lectures: 24h

The course last run:
Fall 1991.

Goals:
To give a survey of previous and current approaches to Intelligent Autonomous Agent design. In particular, I am going to discuss the “classical” paradigm: SENSE-THINK-ACT, with its variations, and contrast it to the “emergent functionality” paradigm originating from Brooks’s subsumption architecture.

Prerequisites:
Introduction to AI (TDDA 58).
Knowledge Representation (TDDA 16).
Automata Theory (at basic level).

Organization:
A series of 8 lectures presenting general ideas, followed by 4 seminars for discussing more detailed topics chosen by participants.

Contents:
1. SENSE-THINK-ACT paradigm;
 - description of several “flagships” of this approach;
 - detailed analysis of SOAR;
 - analysis of complexities involved;
 - possible solutions.
2. EMERGENT FUNCTIONALITY paradigm;
 - description of several subsumption systems;
 - methods for combining primitive behaviors into complex ones.

Literature:
Architectures for Intelligence, Kurt van Lehn (ed.), Lawrence Erlbaum Associates, 1991.

Teacher:
Jacek Malec.

Examiner:
Jacek Malec.

Schedule:
Spring 1994.

Examination:
Term report.

Credit:
3 points.

Compiler Construction – Advanced Course

Recommended for:

For graduate students.

Part 1, Lectures	8 h
Part 2, Lectures	30 h
Project lab work	10-20 h

The course last ran:

Spring 1991

Goals:

The goal of this course is to give the student a thorough insight in compiler construction. The course includes topics not included in undergraduate courses or advanced research topics.

Prerequisites:

The undergraduate compiler courses Compiler Construction (TDDA37) or Compilers and Interpreters (TDDA28), or equivalent. Pteori I is also recommended but not required.

Contents:

Part 1:

- **Attribute grammars, Denotational semantics.**

(Only for students who have not taken Pteori I (TDDA43) or equivalent).

Part 2:

- **Code generation:**

Code generator generators, register allocation, register colouring, code generation by pattern matching.

- **Data flow analysis:**

Basic blocks, flow graphs, Intraprocedural data flow analysis, Interprocedural data flow analysis, reducible flow graphs, iterative data flow algorithms, monotone data flow analysis frameworks.

- **Optimization:**

Code-improving transformations, Alias analysis, Peephole optimization.

- **Compilation methods for Very High Level Languages:**

Data type transformations, Finite differencing.

- Incremental Compilation Techniques.
- Compiler Generation Techniques.
- Project: Practical programming exercises on solving data flow analysis equations.

Literature:

Aho, Sethi, Ullman: Compilers - Principles, Techniques and Tools, Addison-Wesley.
(M.S. Hecht: Flow Analysis of Computer Programs, North-Holland).
Articles and reports.

Teacher:

Mariam Kamkar.

Schedule:

Spring 1994.

Examiner:
Mariam Kamkar.

Examination:
Part 1: written examination.
Part 2: exercises and written examination, or alternatively: exercises and a written summary of the course material.

Credit: **6 - 7 p**
Part 1 gives 1 point for students who have not taken Pteori I or equivalent.
Part 2 gives 4 points for the theoretical part, and 1-2 points for the project part.

Formal Description Techniques for Distributed and Communicating Systems

Recommended for:
Graduate students in Computer Science.

Lectures: **28 h + seminars and homework**

The course last ran:
New course.

Goals:
An introduction to and a comparative study of the standard formalisms used for description of distributed and communicating systems: ESTELLE, LOTOS, (ISO standards) and SDL (CCITT standard).

Prerequisites:
Basic notions of discrete math, otherwise self-contained course.

Organization:
An intensive course including lectures and seminars.

Contents:
- underlying formal models,
- expressiveness and mechanisms of abstraction,
- parallelism, synchronization and communication,
- time constraints and implementation dependence,
- application context,
- supporting tools.

Literature:
K. J. Turner (ed.) *Using Formal Description Techniques: An Introduction to ESTELLE, LOTOS and SDL*, Wiley 1993.

Teachers:
Piotr Dembinski, and possibly invited seminar speakers from Sweden and Finland.

Examiners:
Piotr Dembinski, one of th designers of ESTELLE during his employment in France 1985 - 1989.
Jan Maluszynski.

Schedule:
Intensive course during 4-5 weeks in January/February, 1994.

Examination:
Obligatory homework.

Credit:
5 p.

Information Needs and Uses

Recommended for: **_Lectures:_ 24 h**
Graduate students.

The course last ran:
New course.

Goals:
Provide an introduction to the extensive literature on information needs and uses.

Prerequisites:
None

Organization:
Reading assignment will be specified or distributed one week before each meeting. Students will take turns in presenting the contents of the reading assignment and in leading the discussions of issues arising from it.

Contents:
Defining information needs and uses
Dimensions of needs and uses, e.g. current awareness - retrospective, facts - ideas
Measuring needs and uses
Results from selected studies
Interpreting the results
Impact on system design

Literature:
Articles and reports handed out during the course.

Teachers:
Roland Hjerppe, possibly some invited lecturers.

Examiner:
Roland Hjerppe

Schedule:
Spring 1994

Examination:
Attendance at seminars and term paper.

Credit:
3 p.

Datorisering och samhällsutveckling

Recommended for: ***Seminars:*** **24 h**

Kursen vänder sig till forskarstuderande som vill arbeta med att sätta sina ämnesinriktade studier och kunskaper i ett större - samhälleligt - sammanhang.

The course last ran:

Ny kurs.

Goals:

Göra deltagarna medvetna om rådande, motstridiga uppfattningar om innebörder av modern informationsteknik. Ge en översikt över teman i debatterna kring datorisering och samhällsutveckling. Reflektera över vilka innebörder man själv vill ge interaktionen mellan samhällsutveckling och spridning av ny informationsteknik.

Prerequisites:

Inga särskilda krav utöver att vara forskarstuderande eller att vilja ta en kurs på nivån 61-poäng i sitt huvudämne.

Organization/Examination:

Kursen innefattar återkommande seminarier och eget arbete.

Examinationen omfattar en allmän del i form av aktivit seminariedeltagande, en uppsats och presentation av den. Den utvidgade delen omfattar därutöver ännu en rapport. Denna ska även ta upp material som deltagaren själv letat fram.

Contents:

Kursen kan, styrd av deltagarnas intressen, ta upp några eller alla följande teman:

- Drömmar om informationstekniska utopier.
- Ekonomiska och organistoriska aspekter på datorisering.
- Datorisering och arbetslivet.
- Sociala relationer i ”elektroniska samhället”.
- Social kontroll och den enskildes integritet.
- Säkerhet och pålitlighet.
- Etiska aspekter och experternas yrkesansvar.

Literature:

C. Dunlop and R. Kling Computerization and Controversy Value Conflicts and Social Choices, Academic Press, Harcourt Brace Jovanovich, Boston, 1991, 758 sidor Artiklar.

Teacher:

Hans-Erik Nissen.

Examiners:

Göran Goldkuhl, Hans-Erik Nissen.

Schedule:

Vårterminen 1994.

Credit:

3 poäng för allmän redovisning av kursarbete + ca 3 poäng för utvidgad redovisning.

General Graduate Courses 93/94

Graduate Courses in the School of Engineering 93/94

Vetenskapsteori, Ingemar Nordin, Tema H, tel. 28 22 20

Presentationsteknik (in English), Lars-Gunnar Pettersson, IFM, tel. 28 12 49

Teknikhistoria, Lars Alm, tel. 28 10 12 for information

Graduate Courses in the School of Engineering 93/94

Recommended Master Courses

C3-courses

TDDA11	ADA and Programming Languages, 3p
TDDA14	AI Programming, 5 p
TDDA21	Concurrent Programming, 4 p
TDDA37	Compiler Construction, 3,5 p
TDDA38	Database Technology, 2,5 p
TDDA41	Logic Programming, 4,5 p
TDDA43	Programming Theory, 4 p
TDDA99	Psychology of Communication, 4 p
TDDB60	Methodology of Program Development and Programming Development Project

C4-courses

TDDA12	System Development, 3,5 p
TDDA16	Representation of Knowledge in AI, 2,5 p
TDDA18	Natural-Language Processing, 3,5 p
TDDA30	Programming Theory II, 3 p
TDDA32	Design and Analysis of Algorithms, 3,5 p
TDDA33	Object-Oriented Programming, 2 p
TDDA66	Expert Systems-Methods and Tools, 2 p
TDDA67	Distributed Systems, 1,5 p
TDDB02	Software Quality, 3 p
TDDB10	Human-Computer Interaction, 2,5 p
TDDB15	Computer Aided Software Engineering for Development and Maintenance, 2 p
TDTS41	Computer Networks, 3,5 p
TDTS51	Advanced Computer Architecture, 1,5 p

SVL-courses

Teorier och strategier för informationssystem, 3 p
CASE-verktyg i systemutveckling, 5 p
Design av användbarhet, 5 p
Metodmodellering och CASE-skal, 5 p

Recommended Master Courses

Laboratory-Oriented Courses and Activities 93/94

Like the graduate courses these are open for all graduate students at the department , but they are organized so as to have a direct link to activities in each laboratory. Additional course activities may be announced during the year.

The Laboratory for Complexity of Algorithms (ACTLAB)
Fixed time for lab meetings: **Wednesdays between 13-15.**

The Laboratory for Application Systems (ASLAB)
Seminar Series joint with FOA: *Simulation in Command and Control Systems*.

Fixed time for lab meetings: **Thursdays between 13-15**.

The Laboratory for Computer Aided-Design of Digital Systems (CADLAB)
Fixed time for lab meetings: **Thursdays between 13-15.**

The Laboratory for Computer Assistance in Engineering (CAELAB)
Fixed time for lab meetings: **Wednesdays between 15-17**.

Economic Information Systems (EIS)
These courses are planned for the spring:
Styrning av verkstadsflöden
Doktrinhistoria
Företagsekonomiska teorier

No fixed time for lab meetings.

The Laboratory for Library and Information Science (LIBLAB)
Fixed time for lab meetings:**Wednesdays between 13-15.**

The Laboratory for Intelligent Information Systems (IISLAB)
Fixed time for lab meetings: **Wednesdays between 13-15.**

The Laboratory for Logic Programming (LOGPRO)
Fixed time for lab meetings: **Wednesdays between 15-17.**

People, Computers and Work (MDA)
Fixed time for lab meetings:**Wednesdays between 13 - 15**.

The Laboratory for Natural Language Processing (NLPLAB)
Fixed time for lab meeting**s: Wednesdays between 15-17.**

Laboratory-Oriented Courses and Activities 93/94

The Laboratory for Programming Environments (PELAB)
Fixed time for lab meeting**s: Wednesdays between 15 - 17.**

The Laboratory for Representation of Knowledge in Logic (RKLLAB)
Fixed time for lab meetings: **Wednesdays between 13-15.**

Information Systems and Work Contexts (VITS)
Fixed time for lab meetings: **Some Mondays between 10-12.**

Research Organization and Laboratories

The department is led by a Department Board (institutionsstyrelse). The Department Chairman ("prefekt") is Anders Haraldsson, as from July 1, 1990. Vice Chairman is Erik Sandewall.

The two main areas of activity are reflected in two subordinate committees,:

- The Undergraduate Teaching Committee (IDUN), whose Chairman is Olle Willen.

- The Research Committee (FND), whose Chairman is Sture Hägglund.

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

The responsibility for managing the graduate education program rests with the Research Secretariat, made up of Sture Hägglund, Chariman of the Research Committee, Per-Olof Fjällström, Director of Graduate Studies (forskarstudierektor) and Lillemor Wallgren, Administrator of Graduate Studies.

The graduate research at IDA is organized into the following laboratories:

ACTLAB – Laboratory for Complexity of Algorithms

Per-Olof Fjällström

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

ASLAB – Application Systems Laboratory

Sture Hägglund

ASLAB is oriented towards the study of knowledge-based approaches to software development, including certain aspects of human-computer interaction. Major projects in the lab concern engineering environments for generic knowledge systems and expert critiquing approaches. Joint projects involve cooperation with industry in the knowledge transfer programme, and with several other research groups. Associated with ASLAB is the *Industrial Software Technology project,* studying software engineering issues in an industrial context.

CADLAB – Laboratory for Computer-Aided Design of Digital Systems

Krzysztof Kuchcinski

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

Research Organization and Laboratories

CAELAB – Laboratory for Computer Assistance in Engineering

Anders Törne, Tore Risch

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

EIS – Economic Information Systems

Birger Rapp

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

IISLAB – Laboratory for Intelligent Information Systems

Lin Padgham

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

LIBLAB – Laboratory for Library and Information Science

Roland Hjerppe

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

LOGPRO – Laboratory for Logic Programming

Jan Maluszynski

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

MDA – People, Computers and Work

Toomas Timpka

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

Research Organization and Laboratories

NLPLAB – Natural Language Processing Laboratory

Lars Ahrenberg

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

PELAB – Programming Environments Laboratory

Peter Fritzson

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

RKLLAB – Laboratory for Representation of Knowledge in Logic

Erik Sandewall

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

VITS - Development of Information Systems and Work Contexts

Göran Goldkuhl

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

Research Organization and Laboratories

Faculty

Faculty Engaged in Graduate Study Program

Lars Ahrenberg, Ph. D., Uppsala 1987. Associate professor (*docent, högskolelektor*) of computational linguistics. Group leader, NLPLAB. Previous affiliations Uppsala and Göteborg.

Syntax, semantics and pragmatics of natural language; natural language understanding; natural language interfaces; text generation.

Christer Bäckström, Ph. D., Linköping 1992. Research associate (*forskarassistent*).

Planning and reasoning about plans, algorithms and complexity for AI problems, representation and reasoning about knowledge.

Douglas Busch, Ph. D., Rockefeller 1973. Associate professor (*högskolelektor*) of logic and theoretical computer science. Previous affiliation Mcquarie Univ., Sydney, Australia. On partial leave.

Application of theories from formal logic to problems in theoretical computer science and artificial intelligence; semantics of logic programs; philosophical questions in artificial intelligence.

Nils Dahlbäck, Ph. D., Linköping 1992. Assistant professor (*högskolelektor*), cognitive Science.

Natural language processing, especially empirically based computational models of discourse. Cognitive aspects of discourse coherence in man and machine. Philosophy of mind and its consequences for empirical theories in cognitive science.

Faculty

Patrick Doherty, Ph. D., Linköping 1991. Assistant professor (*högskolelekto*r), logic and theoretical computer science.

Logical approaches to knowledge representation; reasoning with incomplete information, non-monotonic reasoning; reasoning with uncertainty, fuzzy logic; multi-valued and partial logics.

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

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

Johan Fagerström, Ph. D., Linköping 1988. Assistant professor (*högskolelektor*), computer science. Director of undergraduate studies in System Architecture.

Distributed systems, parallel systems, operating systems.

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

Computational geometry, analysis of algorithms, data structures.

Faculty

Peter Fritzson, Ph. D., Linköping 1984. Associate professor (*docent, högskolelektor*), computer science. Group leader, PELAB. Previous affiliations Sun MicroSystems USA 1985-86, Linköping Univ. Hospital 1976-79.

Programming environments, scientific computing, debugging tools, incremental compilation technology, compiler generation, compilers for parallel hardware.

Göran Goldkuhl, Ph. D., Stockholm 1980. Associate professor (*docent, högskolelektor*), administrative data processing. Group leader VITS. Previous affiliations Stockholm, Göteborg.

Theories/methods on problem formulation, activity analysis, IS design and evaluation; ISD methods and customization of CASE tools; Humanistic science traditions and qualitative research methods.

Anders Haraldsson, Ph. D., Linköping 1977. Associate professor (*högskolelektor*), computer science. Department chairman. Previous affiliation Uppsala.

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

Richard Hirsch, Ph. D., Gothenburg 1989. Lecturer (*docent, högskoleadjunkt*) of general and computational linguistics. Thesis supervision in the laboratory for natural language processing. Previous affiliation Gothenburg.

Syntax, semantics, and pragmatics of natural languages; discourse analysis; argumentation theory; philosophy of language.

Faculty

Roland Hjerppe. Researcher (*forskare*). Group leader, LIBLAB. Previous affiliation KTH, DFI and expert mission Tanzania. Visiting Distinguished Scholar at Office of Research, OCLC Inc. in Columbus, Ohio, 1988-89.

Library science and systems, hypertext and -media, knowledge organization and information retrieval, citation analysis and bibliometrics, computer support for personal and cooperative activities, virtual environments.

Sture Hägglund, Ph. D., Linköping 1980. Professor of knowledge-based systems. Research Committee chairman. Group leader, ASLAB. Previous affiliation Uppsala.

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

Krzysztof Kuchcinski, Ph. D., Gdansk 1984. Acting professor of computer systems. Group leader, CADLAB. Previous affiliation Technical Univ. of Gdansk, Poland.

Computer architecture, computer-aided design of digital systems, VLSI, test generation methods.

Jonas Löwgren, Ph. D., Linköping 1991. Assistant professor *(högskolelektor*), human computer interaction.

Human Computer Interaction; User Interface Management Systems; Usability issues in information systems development; Expert Critiquing Systems.

Faculty

Jacek Malec, Ph. D., Wroclaw 1987. Research associate (*forskarassistent*). Previous affiliation Technical Univ. of Wroclaw, Poland.

Artificial Intelligence: knowledge representation, planning, reactive systems, autonomous systems architecture, dynamic scene description.

Jan Maluszynski, Ph. D., Warszawa 1973. Professor of programming theory. Several previous affiliations. Group leader, LOGPRO.

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

Ulf Nilsson, Ph. D., Linköping 1992. Assistant professor (*högskolelektor*), logic programming. On leave 1991–1992 to SUNY at Stony Brook, USA.

Logic programming and deductive databases; Evaluation strategies for query processing; program transformation and abstract interpretation.

Anders G. Nilsson, Econ. Dr., Stockholm 1991. Acting professor of Economic Information Systems. On leave from the Institute for Business Process Development (Institute V) and Stockholm School of Economics. Previous affiliations Royal Institute of Technology and Stockholm University.

Business modelling, strategy planning, activity based development, information systems development, maintenance management, application packages, information management.

Faculty

Lin Padgham, Ph. D. , Linköping 1989. Assistant professor *(högskolelektor)*, computer science. Previous affiliation Univ. of Oregon, USA, and Tektronix. Group leader, IISLAB.

Inheritance, default reasoning, taxonomical reasoning, object-oriented systems.

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

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

Birger Rapp, Econ. Dr., Stockholm 1974, Professor of Economic Information Systems. Several previous affiliations. On leave 1992-93.

Accounting systems, economic control, IT and organisation, production, economics.

Tore Risch, Ph. D., Uppsala 1978. Professor of Engineering Databases. Previously at Uppsala University, IBM Almaden Research Lab. (San Jose, CA), Stanford Research Institute, Syntelligence Inc. (Sunnyvale, CA), HP Laboratories (Palo Alto, CA), and Stanford University.

Database support for engineering and scientific applications, e.g., object-oriented databases, heterogeneous data bases, active databases, and real-time databases.

Faculty

Erik Sandewall, Ph. D., Uppsala 1969. Professor of computer science. Vice-Chairman of the department. Research Committee Chairman (1983-1990). Group leader, RKLLAB. Several previous affiliations.

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

Kristian Sandahl, Ph. D., Linköping 1992, Research associate (*forskarassistent).*

Knowledge engineering, industrial software technology.

Toomas Timpka, MD., Stockholm 1983, Ph. D., Linköping 1989. Associate professor (*docent*, *högskolelektor*), computer and information science. Group leader, MDA.

Hypermedia, computers and society, human-computer interaction, systems development.

Anders Törne Ph. D., Uppsala 1980. Associate professor (*högskolelektor)*, computer support in automation. Group leader, CAELAB. Previously at ABB Corporate Research, Västerås. Director of the graduate study programme. CENIIT manager from 1992.

Computer support for generation, transformation, and use of information in manufacturing processes. Architectures for processing control and supervision. Robot programming.

Guest Researchers and Ph.D.'s in Transition Engaged in Graduate Study Program

Roland Bol, Ph. D., Amsterdam 1991. Guest researcher. Previous affiliation CWI, Amsterdam.

Logic programming, non-monotonic reasoning, deductive databases, process algebra.

Staffan Bonnier, Ph. D., Linköping 1992. On leave 1991-92 to Ecole Normale Supérieure, Paris, France.

Logic and functional programming languages, extended unification and types for logic programs. More recently automatizing inductive proofs of program correctness.

Brant Cheikes, Ph. D., Univ. of Pennsylvania, 1991. Guest researcher.

Natural language processing and cooperative dialogue; architectures for response-planning systems, esp. intelligent help systems; simulator-based training systems. Applications of knowledge-based systems technology for operator training in the process industry.

Keith Downing, Ph. D., University of Oregon 1990. Guest researcher.

Model-based diagnosis, qualitative physics, reason maintenance, artificial intelligence in medicine. Applications of model-based simulation and diagnosis to physiological domains.

Faculty

Wlodek Drabent, Ph. D., Warszawa 1985. Assistant professor (*högskolelektor*). On leave from Institute of Computer Science, Polish Academy of Sciences.

Logic programming, programming languages semantics.

Henrik Eriksson, Ph. D., Linköping 1991. On leave at Stanford University, Stanford, California 1991 – 93.

Knowledge-based systems, knowledge acquisition, software development environments, software reuse.

Robert L. Glass, M. Sc., University of Wisconsin 1954. Guest professor. President, Computing Trends. Previous affiliation Software Engineering Institute, Seattle University, The Boeing Company, USA.

Software engineering: software quality, maintenance, design, technical communication, creativity. The importance of the application domain.

Bengt Lennartsson, Ph. D., Göteborg 1974. Associate professor (*högskolelektor*), software engineering. Previous affiliation Luleå. Group leader, PELAB until 1988. On leave 1992 .

Real-time systems, industrial software technology, large scale software development.

Faculty

Witold Lukaszewicz, Ph. D., Warsaw University 1979. Guest professor. On leave from the Institute of Informatics, Warsaw University, Poland.

Knowledge representation, non-monotonic reasoning, programming methodology.

James M. Nyce, Ph. D., Brown 1987. Guest professor, computer and information science. Previous affiliation Brown.

Work and knowledge (medicine and academia); tradition, innovation and technology; hypertext and visual resource development paths.

Jukka Paakki, Ph. D., University of Helsinki, 1991. Associate professor of computer science (University of Jyväskylä, Finland). Guest researcher.

Programming paradigms, language design and implementation, attribute grammars, logic programming.

Ralph Rönnquist, Ph. D., Linköping 1992. Lecturer (*högskoleadjunkt*) computer science.

Formal representation techniques and advanced computation methods applied for information systems.

Nahid Shahmehri, Ph. D., Linköping 1991. Lecturer (*högskoleadjunkt*), computer science.

Programming theory, programming languages, debugging tools, compiling technology.

Faculty