

Lehrstuhl für Informatik 2 – Jahresbericht 2002

(*Programmiersysteme*)

Leiter:

Prof. Dr. Michael Philippsen, ab 01.04.02
Prof. em. Dr. H.-J. Schneider, bis 30.09.02

Mitarbeiter:

Allendorf, Helmut, Dipl.-Ing. (FH)	Programmierer
Barthelmann, Volker, Dipl.-Inf.	wiss. Mitarbeiter
Brütting, Agnes	Sekretärin - halbtags
Dormeyer, Ricarda, Dipl.-Inf.	wiss. Mitarbeiterin, bis 30.09.02
Fischer, Ingrid, Dr.-Ing.	wiss. Mitarbeiterin
Forman, Franz	teilabgeordnet vom Dietrich-Bonhoeffer-Gymnasium, Oberasbach
Glaser, Günter, Dipl.-Inf.	wiss. Mitarbeiter, bis 31.05.02
Gröbner, Matthias, Dipl.-Inf.	wiss. Mitarbeiter
König, Roman, Dr.-Ing.	wiss. Mitarbeiter, bis 31.12.02
Minas, Mark, PD Dr.-Ing.	Akad. Oberrat, beurlaubt ab 01.08.02
Puhlmann, Hermann, Dr.	teilabgeordnet vom Ohm-Gymnasium, Erlangen
Schnitzer, Bernd, Dipl.-Inf.	wiss. Mitarbeiter, vom 01.09.02 bis 31.12.02
Schörmal, Elfriede	Sekretärin - halbtags
Strüber, Frank, Dipl.-Inf.	wiss. Mitarbeiter, ab 01.09.02
Uebler, Manfred	Programmierer
Veldema, Ronald, Dr.-Ing.	wiss. Mitarbeiter, ab 15.10.02
Weißel, Andreas, Dipl.-Inf.	wiss. Mitarbeiter, ab 01.12.02
Wilke, Peter, PD Dr.-Ing. habil.	Akad. Oberrat, beurlaubt ab 15.10.02

Stipendiaten:

Kókai, Gabriella, Dr.-Ing.	Bayer. Wissenschaftsministerium
Nagorniy, Stanislav	Graduiertenkolleg, bis 30.09.02
Tóth, Zoltán	DAAD - Siemens
Ványi, Róbert	DAAD – Siemens, bis 30.09.02
Zvada, Silvia	Bayer. Wissenschaftsministerium
Haworth, Brigid Mary, Dr.-Ing.	Univ.-Erlangen, bes. frauenfördernde Maßnahmen, bis 28.02.02
Weber-Russell, Prof. Sylvia, PhD.	Fulbright-Kommission, bis 28.02.02

Lehrbeauftragte:

Hindel, Bernd, Honorar-Prof. Dr.-Ing.	Fa. method park Software AG, Erlangen, bis 31.03.02
Kips, Detlef, Honorar-Prof. Dr.-Ing.	Fa. BASYS GmbH, Erlangen, bis 31.03.02
Kreisel, Klaus, Dr.rer.nat.	Emmy-Noether-Gymnasium, Erlangen
Nilson, Jörg, Dr.-Ing.	Fa. 3SOFT GmbH, Erlangen

Kooperationspartner:

- Arbeitsgemeinschaft Software-Qualität Franken e.V. (Prof. Dr. Hindel)
- ESPRIT-Programme APPLIGRAPH (Applications of Graph Transformation)
- EU - TMR Network GETGRATS (General Theory of Graph Transformations)
- Universität Szeged, Ungarn (Prof. Dr. Csirik)
- Technische Universität Berlin (Prof. Dr. Ehrig)
- Universität Bremen (Dr. Hoffmann)
- Universität Darmstadt (Prof. Dr. Schürr)

- Universität Kaiserslautern (Prof. Dr. Litz, Dr. Frey)
- Universität Krakau, Polen (Prof. Dr. Grabska)
- Universität Rom (Prof. Dr. Bottoni)
- Universität Umeå, Schweden (Dr. Drewes)
- Tripos Inc, San Francisco, USA (Dr. Berthold)
- University of New Hampshire, USA (Prof. Dr. Weber-Russell)
- University of Western Australia (Prof. Dr. Bräunl)
- Astrum GmbH, Erlangen
- Audi AG, Ingolstadt
- Basys GmbH, Erlangen (Prof. Dr. Kips)
- Fachhochschule Nürnberg (Prof. Dr. Rieckeheer)
- Fraunhofer Institut für Integrierte Schaltungen, IIS-A
- imbus GmbH, Erlangen
- INA Wälzlager Schaeffler KG, Herzogenaurach (Dr. Lessen)
- Infoteam GmbH, Bubenreuth (Hr. John)
- LION bioscience AG, Heidelberg (Hr. Kirsch)
- SWIPe GmbH, Nürnberg (Hr. Schacht)
- 3SOFT GmbH, Erlangen

Forschungsberichte:

Das Jahr 2002 war für den seit 1972 bestehenden Lehrstuhl Informatik 2 ein Jahr des Umbruchs. Prof. Dr. H.-J. Schneider, der den Lehrstuhl aufgebaut und zu einer international beachteten Arbeitsgruppe gemacht hat, wurde Ende September emeritiert. Dank einer Übergangsfinanzierung war es möglich, dass der Nachfolger, Prof. Dr. Michael Philippsen, schon vorzeitig ab April sein Amt in Erlangen aufnehmen konnte, was einen gleitenden Übergang erlaubte.

Während Graphgrammatiken und –transformationen, computerlinguistische Fragen sowie neuartige Programmier- und Spezifikationssprachen bis hin zu Werkzeugen des Softcomputing lange Jahre den Schwerpunkt der Arbeiten von Lehrstuhl 2 gebildet haben und natürlich in noch laufenden erfolgreichen Projekten weiter fortgesetzt werden, wird sich längerfristig die fachliche Ausrichtung auf das effiziente Programmieren von Nicht-Desktop-Architekturen konzentrieren.

Sowohl auf eingebetteten Systemen als auch auf Rechnerbündeln wird derzeit noch sehr maschinen-nah programmiert. Hier wünscht man sich zwar ein höheres Abstraktionsniveau, um die Programmierproduktivität und Wartbarkeit der erstellten Software zu steigern, kann dies aber heutzutage noch nicht erreichen, weil der von bisherigen Übersetzern erzeugte Code zu voluminös und/oder zu ineffizient ist. Der Lehrstuhl wird versuchen, Optimierungstechniken zu entwickeln, die diese Lücke schließen. Mit ersten Arbeiten in diese Richtung wurde bereits begonnen.

Der Wechsel in der Lehrstuhlleitung bedeutet aber nicht nur Neuerung und Wandel, sondern auch Kontinuität. Die gewohnten Räumlichkeiten sind inzwischen renoviert - der Lehrstuhl erstrahlt in neuem Glanz. Viele liebgegewonnene Traditionen werden fortgesetzt. Langjährige geschätzte Mitarbeiter haben die Chance, den Wandel auch nach ihren Vorstellungen mitzugestalten. Und auch inhaltlich knüpfen zukünftige Arbeitsgebiete im Bereich der Multiprozessorsysteme und der verteilten Programmiersprachen an Themengebiete an, die den Lehrstuhl schon früher bewegt haben.

1. Übersetzung für eingebettete Systeme

(Volker Barthelmann, Frank Strüber)

Ob im Auto oder zur Steuerung von Maschinen, in vielen elektronischen Geräten werden eingebettete Systeme zunehmend ein unverzichtbarer Bestandteil. Mit zunehmender Komplexität der Anwendungen ergibt sich die Anforderung, Software für eingebettete Systeme in modernen Programmiersprachen zu entwickeln. Bei eingebetteten Systemen ist die Sprache C dominierend (80% im Jahr 2000). Am Lehrstuhl wird insbesondere an folgenden zwei Teilgebieten gearbeitet:

- Bedingt durch den großen Erfolg von Java bei Desktop- und Serveranwendungen, stellt sich die Frage, ob Java nicht auch im Bereich eingebetteter Systeme angewendet werden könnte. Desktop

Java eignet sich aufgrund des sehr hohen Ressourcenverbrauchs allerdings nicht für Anwendungen in eingebetteten Systemen, wo typischerweise Speicherplatz, Rechenzeit oder auch Batterielaufzeit kritische und teure Faktoren darstellen. Deshalb wird gerade erforscht, wie Java geeignet für den Einsatz in eingebetteten Systemen verändert werden kann, ohne die Vorteile von Java wie Portabilität einzubüßen.

- Außerdem wird untersucht, inwieweit im Bereich gewisser eingebetteter Systeme eine engere Zusammenarbeit von Übersetzern und statischen Betriebssystemen Vorteile bei Effizienz und Sicherheit bringt. Das Interesse liegt hier speziell auf kleinen eingebetteten Systemen, die in Hochsprachen wie C programmiert werden und statische Echtzeitbetriebssysteme benutzen. Solche Systeme werden z.B. in der Automobilindustrie verstärkt eingesetzt. Durch die Stückzahlen im Millionenbereich müssen die Kosten so klein wie möglich gehalten werden. Einsparungen von einigen Dutzend Bytes können hierbei Kosten in Millionenhöhe einsparen. Eine Studie, die den Overhead eines C-Übersetzers gegenüber direkt in Assembler entwickelten Programmen auf 100 - 1000% schätzt, zeigt das noch große Einsparungspotential durch Optimierungen.

Im Jahr 2002 wurde mit den Arbeiten an Java für eingebettete Systeme begonnen. Es wurden neue Verfahren zur verbesserten Übersetzung für statische Betriebssysteme entwickelt, wobei das Hauptaugenmerk hierbei vor allem auf Einsparung von RAM lag. Auf der LCTES & SCOPES Joint Conference wurde eine Arbeit über Inter-Task Registervergabe für statische Betriebssysteme vorgestellt.

2. Cluster Computing

(Ronald Veldema)

Das Programmieren von Rechnerbündeln befindet sich derzeit noch in einer sehr maschinennahen Phase. Als besonders dringend erscheint die Behandlung der non-uniformen Speicherzugriffshierarchie, die in verteilten Systemen neben Registern, Cache und Hauptspeicher um eine weitere Leistungsebene vertieft ist. Wie im folgenden skizziert, erfordern gute Lösungen Arbeiten auf allen Systemschichten.

Betriebssysteminteraktion: Welche Leistungsengpässe bestehen und wie können diese beseitigt werden? Wie kann der Kommunikationskanal - auch durch das Betriebssystem hindurch - weiter verschlankt werden? Sowohl für Methodenaufrufe, die in der Regel gute Latenzzeiten verlangen, als auch für Datenstromübertragungen, bei denen es auf hohe Bandbreite ankommt?

Übersetzerunterstützung, Laufzeitsystem: Können verteilte Systeme der Tradition von Einprozessorsystemen folgend auch weiterhin mit der Illusion eines transparenten Zugriffs programmiert werden? Kann man durch statische Analyse nebenläufigen objekt-orientierten Codes Informationen über bestehende Lokalitätsbeziehungen extrahieren? Welche Optimierungen sind dann möglich? Profitieren statische Analyse und Laufzeitmechanismen voneinander? Wie können durch Programmanalyse Pre-Fetch- und Post-Store-Kommandos im generierten Code erzeugt werden, durch die Kommunikation und Berechnung überdeckt und dadurch Wartezeiten vermieden werden?

Programmiersprache, Systementwurf: Programmierer haben in der Regel das anwendungsspezifische Hinterrundwissen über bestehende Lokalitätszusammenhänge. Außer in datenparallelen Sprachen ist bisher unklar, wie dieser Aspekt programmiersprachlich sinnvoll ausgedrückt werden kann. Sinnvoll bedeutet dabei, dass es für den Programmierer ausreichend bequem aber auch verständlich sein soll, derartige Zusammenhänge zu codieren. Sinnvoll bedeutet aber auch, dass das codierte Wissen aussagekräftig genug sein muss, um vom Übersetzer oder dem Laufzeitsystem effizienzsteigernd genutzt werden zu können. Schließlich bedeutet sinnvoll, dass sich Sprachmittel für den Aspekt der Lokalitätszusammenhänge nahtlos in die Grundsprache einfügen müssen, ohne dass es z.B. zu Vererbungsanomalien kommt. Noch sehr schlecht verstanden ist generell, wie man verteilte Systeme von vorneherein so entwirft, dass Lokalität per Design vorhanden ist.

Im Jahr 2002 wurden die Arbeiten an JavaParty und Manta fortgesetzt, indem zunächst untersucht wurde, wie JavaParty effizient in der Manta-Umgebung übersetzt werden kann. Zukünftige Arbeiten konzentrieren sich auf neue Optimierungstechniken im Übersetzer, um einige der genannten Probleme zu lösen.

3. Graphgrammatiken und Graphtransformationen

(Ingrid Fischer, Mark Minas)

Das Gebiet der Graphgrammatiken und Graphtransformationen kombiniert Ideen aus den Bereichen Graphentheorie, Algebra, Logik und Kategorientheorie. Die breite Anwendbarkeit in verschiedenen Teilgebieten der Informatik resultiert aus der Tatsache, dass Graphen an vielen Stellen als intuitives Hilfsmittel zur Verdeutlichung komplizierter Sachverhalte verwendet werden, z.B. als Daten- bzw. Kontrollflussdiagramme, als Entity-Relationship-Diagramme, Petri-Netze oder zur Visualisierung sowohl von Software- als auch von Hardware-Architekturen. Eine Graphgrammatik kann benutzt werden, um eine Menge syntaktisch korrekter Diagramme zu definieren, d.h. Diagramme, die nach den Regeln eines bestimmten Anwendungsgebietes aufgebaut sind. Graphtransformationen erlauben dynamische Veränderungen derartiger Darstellungen und somit die Beschreibung der Entwicklung von Strukturen. Die zugrundeliegende Theorie ist ein attraktives Hilfsmittel, äußerst unterschiedliche Strukturen in einer einheitlichen Weise zu beschreiben.

Im Jahre 2002 wurden die Arbeiten zur einheitlichen Beschreibung unterschiedlicher Modelle für asynchrone Prozesse fortgesetzt. Petri-Netze basieren auf "gewöhnlichen" Graphen, Statecharts verwenden hierarchische Graphen, die parallele logische Programmierung kann mit Hilfe sogenannter Dschungel graphentheoretisch interpretiert werden, und die Aktorsysteme lassen sich als Graphen darstellen, deren Markierungsalphabet eine Menge von Termgraphen ist. Im Zusammenhang mit der Wiederaufnahme der Vorlesung über Graphtransformationssysteme wurden gewisse Nebenläufigkeitsaussagen, z.B. die Erzeugung aller möglichen "Traces" eines Programms oder die Vertauschbarkeit von Schritten, so allgemein formuliert, dass die Sätze auf alle genannten Modelle in gleicher Weise anwendbar sind. Die Vorlesung über funktionale Programmierung war Anlass, die früher für Miranda und Java geschriebenen Implementierungen kategorieller Konstruktionen nach Haskell zu übertragen. Entsprechende Berichte sind in Vorbereitung

4. Automatische Analyse natürlicher Sprache

(Ingrid Fischer, Ricarda Dormeyer, Sylvia Russell)

Die interessantesten und schwierigsten idiomatischen Beispiele enthalten lexikalische und syntaktische Modifikationen (z.B. "Er schloss die eiternde Wunde, in die er Salz gestreut hat"). Aber auch unveränderliche Idiome (z.B. "Er hat ins Gras gebissen") dürfen bei der automatische Sprachanalyse nicht vernachlässigt werden. Der verfolgte Ansatz zur Idiomanalyse beruht auf zwei Vermutungen. Erstens können Idiome analysierbar sein, das heißt, die einzelnen Komponenten können bei der Analyse des Idioms eine Rolle spielen, müssen es aber nicht. Zweitens spielen bei vielen Idiomen Metaphern eine aktive Rolle im Verstehensprozess, besonders wenn die Idiome analysierbar und modifizierbar sind. Diese Vermutungen werden an verschiedenen Sprachverarbeitungssystemen getestet. Im syntaktischen Bereich wird auf dem Gebiet der Dependenzgrammatiken gearbeitet. Dieser Grammatikformalismus ist besser zur Darstellung freier Wortstellung geeignet als Phrasenstrukturgrammatiken. Die syntaktische Analyse eines Satzes besteht darin, binäre Relationen zwischen den Wörtern des Satzes zu identifizieren. Anders als in Chomskys Theorie müssen zwei Wörter nicht benachbart sein, damit eine solche Relation zwischen ihnen bestehen kann.

Im Jahr 2002 wurden verschiedene existierende Dependenzparser in ein neues allgemeines Schema eingeordnet und so miteinander verglichen. Dies ist besonders interessant, da viele Publikationen verschiedener Dependenzparser sich nicht mit verwandten Arbeiten beschäftigen. Das Schema wurde weiterhin zur Entwicklung neuer Parser genutzt. Gleichzeitig wurde ein Grammatikformat entworfen, das eine neue Darstellung der Wortstellung und diskontinuierlicher Konstituenten erlaubt. Ein aus der Literatur bekannter inkrementeller Chartparser wurde implementiert und an dieses Schema angepasst. Grammatikfragmente für das Deutsche, Lateinische und Englische wurden implementiert. Dabei zeigte sich, dass das entwickelte Grammatikformat eine ebenso einfache Beschreibung von Sprachen mit fester Wortstellung erlaubt wie die Phrasenstrukturgrammatik und dass die Beschreibung von Sprachen mit freier Wortstellung noch einfacher als in der Phrasenstrukturgrammatik ist. Zur Darstellung der Semantik und der Metaphern wurde im Jahr 2002 eine Theorie zur Analyse von metaphorisch gebrauchten Verben untersucht, die ursprünglich auf der Conceptual-Dependency-Theorie aufbaute. Dieser Ansatz wurde auf Nomen, Adjektive und andere semantische Formalismen verallgemeinert. So kann er zur Darstellung der metaphorischen Grundlage von Idiomen und ihrer Modifikationen ver-

wendet werden. Typisch ist, dass der metaphorische Teil einzelner Wörter parallel zum nichtmetaphorischen Anteil dargestellt wird. Beide Teile werden mit demselben Formalismus realisiert. Eine exemplarische Implementation in einem unifikationsbasierten Phrasenstrukturformalismus wurde entwickelt. An der Zusammenführung der syntaktischen und semantischen Arbeiten wird aktuell gearbeitet.

5. Spezifikation und Generierung graphischer Diagrammeditoren

(Mark Minas, Frank Strüber)

Diagramme werden überall verwendet, wo komplizierte Sachverhalte verständlich dargestellt werden sollen. Diagrammeditoren sind graphische Editoren, die auf eine bestimmte Diagrammsprache zugeschnitten sind. Von gebräuchlichen Zeichenprogrammen unterscheiden sie sich dadurch, dass sie einerseits die erstellten Diagramme in gewissem Sinne "verstehen", zum anderen nur Diagramme zu zeichnen erlauben, die den Regeln des Anwendungsbereiches entsprechen. In diesem Projekt wird das Software-Werkzeug DiaGen entwickelt, mit dem beliebige Diagrammsprachen formal spezifiziert und Diagrammeditoren aus solchen Spezifikationen weitgehend automatisch generiert werden können. Der für die Realisierung eines Diagrammeditors nötige Aufwand wird dadurch drastisch reduziert. DiaGen basiert auf der Modellierung von Diagrammen mit Hypergraphen und der Spezifikation der Diagrammsprache durch eine Hypergraphgrammatik. Ein mit dem Editor frei erstelltes Diagramm wird durch ein äquivalentes Hypergraphmodell repräsentiert. Der Editor kann dieses auf Korrektheit überprüfen und seine syntaktische Struktur mit Hilfe eines Hypergraphparsers ermitteln. Der zusätzlich zur Verfügung stehende Modus des strukturierten Edierens stellt Operationen bereit, die das Hypergraphmodell unmittelbar verändern und die mit programmierten Hypergraphtransformationen spezifiziert werden. DiaGen wurde erfolgreich zur Generierung von unterschiedlichen Diagrammeditoren eingesetzt. Die bisherigen Anwendungen umfassen Editoren u.a. für UML-Klassendiagramme, Flussdiagramme, Struktogramme, Petrinetze und als Beispiel aus anderen Disziplinen die Kontaktpläne der Steuerungstechnik.

Im Jahr 2002 wurde ein grundsätzliches Problem beim Einsatz von graphischen Editoren gelöst, wenn sie als Komponenten von Software-Systemen Datenstrukturen visualisieren sollen, die von anderen Komponenten bereitgestellt werden. Dies setzt die Fähigkeit des Editors voraus, beliebige Datenstrukturen in Diagramme zu übersetzen. Die generische Lösung dieses Problems basiert auf der Darstellung der zu übersetzenden Datenstrukturen mit XML und der Spezifikation des Übersetzungsvorgangs mit XSLT. DiaGen wurde ferner um ein visuelles Spezifikationswerkzeug erweitert, das die Spezifikation und Generierung von Diagrammeditoren mit DiaGen deutlich erleichtert. Während der Benutzer zuvor noch über viel Wissen über Interna von DiaGen verfügen musste, werden nun größere Teile von Diagrammeditoren automatisch generiert. Die graphische Benutzerschnittstelle des Spezifikationswerkzeugs fragt zudem gezielt Informationen vom Benutzer ab. Ausserdem wurde ein Verfahren entwickelt, um dieses Werkzeug auch bei späteren Änderungen von DiaGen mit noch verhältnismäßig geringem Aufwand an diese Änderungen anzupassen.

6. Visuelle, regelbasierte Programmierung

(Mark Minas)

Vielen Anwendungsproblemen liegen Graphen zu Grunde oder lassen sich in natürlicher Weise mit Graphen modellieren. Auf Graphtransformationen beruhende Berechnungsmodelle bieten sich deshalb unmittelbar für ihre Bearbeitung an. Allerdings haben sie den Nachteil, dass Graphen im allgemeinen von den Benutzern nicht als alleinige Datenstruktur akzeptiert werden. Anwendungsspezifische Visualisierungen (z.B. mit Diagrammen) würden die Akzeptanz solcher Programmiersprachen verbessern. Ziel dieses Forschungsprojekts ist deshalb eine visuelle Programmiersprache, in der Datenstrukturen beliebige Diagramme sind, die intern mit Graphen dargestellt und mit Graphtransformationen regelbasiert manipuliert werden. Ausgangspunkt dieser gemeinsamen Arbeiten ist ein Berechnungsmodell, das auf einem Graphtransformationsansatz mit hierarchischen Hypergraphen beruht. Um anwendungsspezifische Visualisierungen erstellen zu können, wird das Spezifikations- und Generierungswerkzeug DiaGen genutzt, das damit im Rahmen dieses Projektes auch weiterentwickelt wird. Langfristiges Ziel ist, in Zukunft anwendungsspezifische Programmiersprachen zur Verfügung zu stellen, die mit der Darstellung ihrer Datenstrukturen und Bedienungsumgebungen an den jeweiligen Anwendungsfall anpassen

bar und deshalb besser anwendbar sind, als dies bei den heute vorherrschenden textuellen Programmiersprachen der Fall ist.

Im Jahr 2002 wurde das DiaPlan zugrundeliegende Typsystem erweitert, das wie bei vielen anderen Programmiersprachen vorrangig dazu dient, Programmierfehler schon zur Übersetzungszeit zu erkennen. Typen von Graphen sind in DiaPlan jeweils eine bestimmte Graphklasse, beispielsweise alle Graphen, die verkettete Listen repräsentieren. Definiert werden solche Typen durch Grammatiken, die alle Graphen der so definierten Klasse erzeugen können. Damit die Typprüfung schon zur Übersetzungszeit, und nicht erst zur Ausführungszeit erfolgen kann, müssen bestimmte Erhaltungseigenschaften gelten. Während die Definition von Typen deshalb bislang auf kontextfreie Grammatiken beschränkt war, konnten wir in diesem Jahr eine echte Erweiterung kontextfreier Grammatiken identifizieren, die eine statische Typprüfung erlauben. Damit kann nun für viele praktisch relevante Graphentypen die Typprüfung statisch durchgeführt werden, für die dies zuvor nicht möglich war. Ein grundsätzliches Problem bei der Programmierung mit Graphtransformationen ist, die linke Seite einer Regel als Teilgraph des Graphen zu finden, auf den die Regel angewandt werden soll. Um die Ausdrucksfähigkeit einzelner Regeln und damit den Programmierkomfort zu erhöhen, dürfen Transformationsregeln in DiaPlan Variablen enthalten. Allerdings wird das Graph-Matching-Problem dadurch komplizierter. In diesem Jahr konnten wir Klassen von Graphtransformationen mit Variablen identifizieren, die effizient handhabbar bleiben.

7. Entwicklung eines softwaregestützten Systems zum Risiko-Management

(Peter Wilke)

Aufgrund gesetzlicher Vorschriften (KonTraG) sind die Vorstände von Aktiengesellschaften persönlich für das Risiko-Management verantwortlich. Durch die Komplexität der Unternehmen ist eine manuelle Ermittlung der Risiken kaum möglich, und es besteht Bedarf für software-basierte Systeme. In diesem Forschungsprojekt wurde untersucht, welche Verfahren sich zu einem integrierten Risiko-Management-System verbinden lassen und wie es in das Unternehmen eingebunden werden kann. Grundlegendes Konzept ist der Zyklus aus Risikoidentifikation, Risikobewertung, Durchführung geeigneter Maßnahmen, Kontrolle und Qualitätssicherung. Dabei wurden in den einzelnen Phasen neben konventionellen Techniken auch neuere Ansätze verfolgt. Beispiele für solche unkonventionellen Ansätze sind Softwareagenten, die im Internet nach Informationen suchen, aber auch Prognosesysteme auf der Basis neuronaler Netze und genetische Algorithmen zur Optimierung.

Im Jahr 2002 wurden die Themen Online-Verfolgung von Prozessen, Optimierung der Basisdaten und Verfolgung kritischer Pfade bearbeitet. Die wesentlichen Ergebnisse waren die Erstellung einer Datenbank zur Erfassung von Datenquellen, die Übertragung von Techniken aus dem Projekt-Management auf das Risiko-Management, u.a. kritische Pfade, kritische Ereignisse und Ressourcen-Management, sowie die Bewertung von Methoden zur Online-Prozessverfolgung. Das System befindet sich bereits bei einigen Pilotkunden im Einsatz und wird vom Industriepartner vermarktet. Das Projekt wurde inzwischen abgeschlossen.

8. Algorithmen für Zeitplanungsprobleme

(Matthias Gröbner, Peter Wilke)

Das Problem der Zeitplanerstellung (Timetabling) besitzt zahlreiche Anwendungsbereiche, so zum Beispiel die Schulstundenplanung, die Personaleinsatzplanung oder die Stundenplanung an Universitäten. Aufgrund des großen Aufwands, den die manuelle Erstellung solcher Zeitpläne erfordert, besteht Bedarf nach einer automatisierten Vorgehensweise. Daher wurden in den letzten Jahren verschiedene computergestützte Verfahren getestet und auch angewendet. Beispiele für solche Verfahren sind Genetische Algorithmen, Tabu Search, Constraint-Logische Programmierung oder diverse andere Heuristiken. Jedoch unterscheiden sich alle bisherigen Implementierungen der Algorithmen für Timetabling-Probleme in ihren Datenstrukturen und damit auch in ihrer Problemlösung, die an die jeweilige Datenstruktur angepasst werden muss.

Um den Beschreibungs- und Implementierungsaufwand für neue Timetabling-Probleme zu reduzieren sowie eine größere Vergleichbarkeit unterschiedlicher Timetabling-Probleme zu gewährleisten, wurde daher im Jahr 2002 zunächst eine formale Beschreibung des Timetabling-Problems vorgenommen. Daraus wurde eine allgemeine Timetabling-Sprache und ein Timetabling-Framework entwickelt, mit

der beliebige Timetabling-Probleme beschrieben werden können. Für einen konkreten Planungszeitraum kann dann mit Hilfe eines Genetischen Algorithmus oder eines einfachen Suchverfahrens ein Zeitplan erstellt werden. Nächstes Forschungsziel wird der Vergleich unterschiedlicher Optimierungsverfahren sein, mit dem Ziel, die Eignung der Verfahren zur Erstellung von Zeitplänen zu untersuchen. Weiterhin ist die Untersuchung der Struktur von Zeitplanungsproblemen von Interesse, um Rückschlüsse auf die Lösbarkeit von Zeitplanungsproblemen unterschiedlicher Komplexität ziehen zu können.

9. Entwicklung und Anwendung von Evolutionären Algorithmen

(Gabi Kókai, Zoltan Tóth)

Das genetisch-logische Programmiersystem GeLog kombiniert Methoden genetischer Algorithmen und induktiv logischer Programmierung. Mit dem automatischen Lernsystem können aus einer induktiv logisch formulierten Aufgabenstellung logische Programme erzeugt werden, die die gegebene Aufgabe lösen. Zum Lernen verwendet GeLog einen genetischen Algorithmus. GeLog verfügt über automatische Parameteradaptation und Metaevolution und vermeidet dadurch die bekannten Nachteile genetischer Algorithmen, wie die große Anzahl von Einstellmöglichkeiten für Evolutionsparameter sowie das Hängenbleiben in lokalen Optima. Im Jahr 2002 wurde das System um Methoden erweitert, die das Auftreten von Overfitting verhindern, also das Erlernen von falschen (bzw. zu speziellen) Theorien aus falsch klassifizierten Beispieldaten. In GeLog wurde ein Filterverfahren gewählt und implementiert, das auf der beim Top-down-Pruning verwendeten Korrelationsheuristik basiert. Zusätzlich erkennt GeLog nun Building Blocks, also im Laufe der Evolution in verschiedenen Individuen vorkommende Konstrukte, die sich positiv auf die Fitness auswirken. Nach der Analyse verschiedener Ansätze wurde das so genannte Linkage Learning als ein geeigneter Mechanismus ausgewählt und realisiert. Genetische Algorithmen werden in Kooperationsprojekten mit der Industrie eingesetzt:

- Genetische Algorithmen zur Generierung redundanzreduzierter FIR-Filter für digitale Schaltkreise: Digitale Filter werden - zumeist als FIR-Filter (Finite Impulse Responde) realisiert - in vielen technischen Bereichen eingesetzt, z.B. in der Kommunikations- und Nachrichtentechnik. Rasche Änderungen der Marktanforderungen erfordern eine einfache und schnelle Generierung von FIR-Filtern. Das dazu am Fraunhofer-Institut für Integrierte Schaltungen IIS-A entwickelt System red-FIR setzt zur automatischen Optimierung der generierten Filter und damit zur Reduktion der Herstellungskosten genetische Algorithmen ein.
- Genetische Algorithmen zur Optimierung von optischen Abbildungssystemen für die Lithographiesimulation: Bei der Herstellung von Halbleiterbauelementen wird die optischen Lithographie angewendet, um kleine Strukturen (Größenordnung 100nm) auf Siliziumscheiben zu erzeugen. Dabei wird eine Maske mit Hilfe eines Abbildungsapparates auf einen mit einem Photolack beschichteten Wafer projiziert. Der belichtete Photolack kann aufgrund geänderter chemischer Eigenschaften leicht abgelöst werden. Das Fraunhofer-Institut für Integrierte Schaltungen IIS-A entwickelt gemeinsam mit der Industrie ein Simulationsprogramm zur Beschreibung dieser mikroelektronischen Fertigungsschritte. Die Abbildung immer kleinerer Strukturen auf den Wafer erfordert immer bessere Methoden (Optical Proximity Corrections) zur Sicherstellung der Exaktheit der erzeugten Strukturen. Seit 2002 wird untersucht, ob und wie mit Hilfe eines genetischen Algorithmus das Design von Masken für beliebige Strukturen optimiert werden kann.
- Strukturierte Bildbeschreibung mit genetischen Algorithmen: Diabetes schädigt schleichend die kleinen Blutgefäße der Retina und kann dadurch zur Erblindung führen. Zur Frühdiagnostik ist eine regelmäßige Augenuntersuchung erforderlich. Ausgehend von Fundusbildern soll das von uns entwickelte System GREDEA den Arzt durch Entwicklung patientenspezifischer Überwachungsprogramme für die Überprüfung der Blutzirkulation unterstützen. Da diese Bilder leider oft von schlechter Qualität sind, mussten im Jahr 2002 verschiedene Bildverarbeitungsmethoden für die Vorverarbeitung und Aufbereitung der Bilder implementiert werden.

10. Erforschung kombinierter Lernverfahren an praktischen Beispielen

(Gabi Kókai, Ingrid Fischer, Szilvia Zvada, Zóltan Tóth, Róbert Ványi)

Das Hauptaugenmerk in diesem PPP-Projekt (Projektbezogenes Personenaustausch-Programm) des DAAD liegt auf der Künstlichen Intelligenz und dort auf der Kombination verschiedener Lernverfah-

ren (insbesondere des Inductive Logic Programming, ILP, und der Genetischen Algorithmen, GA) zur Entwicklung effizienter Algorithmen, die die Vorteile der einzelnen Verfahrensgruppen in sich vereinigen, gleichzeitig aber die Nachteile der Einzelverfahren vermeiden. An verschiedenen praktischen Beispielen werden die überragenden Eigenschaften der neu entwickelten Methoden quantitativ nachgewiesen. Die wesentlichen Ergebnisse des im Jahr 2002 abgeschlossenen Projektes lassen sich wie folgt zusammenfassen:

- Das am Lehrstuhl unter Leitung von Frau Dr. Kókai entwickelte GeLog-System vereinigt induktive Verfahren mit evolutionären Algorithmen. Im Vergleich mit verschiedenen Lernsystemen (FOIL, C4,5 und GEA) lieferte GeLog an fehlererhaltenen Daten die besten Klassifizierungsergebnisse.
- Das GEA-System, eine von Zoltán Tóth entwickelte, einfach anwendbare und flexible Bibliothek für evolutionäre Programmierung, wurde um eine grafische Oberfläche (GraphGEA) erweitert, die die Einstellung der Parameter erleichtert und die Ergebnisse des Evolutionsprozesses (Entwicklung der Fitnesswerten, die besten Individuen, usw.) visualisiert. In der Literatur findet man kein vergleichbares System, das eine interaktive oder eine Offline Evolution ermöglicht.
- Im Teilprojekt "Structural Image Description with Genetic Programming" (SID-GP) evolvierte Róbert Ványi Strukturen (z.B. Kettenkode oder Lindenmayer-Systeme) mit genetischer Programmierung und erzielte damit die genaueste Beschreibung von binären (schwarz-weiß) Bildern. Wie gut diese Beschreibung ist, wird von einer problemspezifischen Fitnessfunktion bestimmt, die das von der evolvierten Struktur erzeugte Bild mit dem Original vergleicht. Diese Idee wurde noch nie verfolgt.
- Die Forschung von Szilvia Zvada zielte auf die Entwicklung von völlig neuen, induktiven Lernverfahren ab, die das Erlernen nicht vollständig angegebener attributierter Grammatiken ermöglichen, denen semantische und eventuell auch syntaktische Regeln fehlen. Im Rahmen des Projektes wurde die neue LAG-Methode (Learning of Attribute Grammars) entwickelt und ein auf LAG basierendes Lernsystem implementiert.

Außerdem wurde im Rahmen des Projektes die Weiterqualifikation des wissenschaftlichen Nachwuchses gefördert: Frau Fischer hat an der Universität Szeged einen Kompaktkurs für PhD-Studenten mit dem Titel "Parsing Natural Languages" gehalten. Frau Fischer, Frau Kókai und Frau Zvada haben sich mit Vorlesungen aktiv am Sommerstudium "Informatica Feminale" der Universität Bremen beteiligt

11. Effiziente Datenanalyse

(Roman König)

Bei der Analyse großer Datenbestände versucht man, aussagekräftige Beziehungen (= Assoziationsregeln) zwischen den Attributen der gegebenen Datenbasis zu finden. Dabei muss zunächst für jede Attributmenge festgestellt werden, wie viele Gegenstände (= Transaktionen) alle diese Attribute besitzen. Die Menge aller Beziehungen zwischen den Attributen mit ihren Zuverlässigkeiten kann dann daraus bestimmt werden. Diese weist aber im Allgemeinen eine hohe Redundanz auf. Daher ist es notwendig und sinnvoll, nach redundanzfreien Erzeugendensystemen für die Assoziationsregeln zu suchen. Die formale Begriffsanalyse bietet die mathematischen Hilfsmittel, um diese Fragestellungen mit fundierten Methoden zu untersuchen. Das Studium der endlichen Hüllsysteme ist die Grundlage für alle Anwendungen auf diesem Gebiet.

Im Jahr 2002 ist es gelungen, den Zusammenhang zwischen den endlichen Hüllsystemen (= Begriffsverbänden) und den Implikationenmengen, die in ihnen erfüllt sind, zu studieren, indem ein passender Kontext gefunden wurde, der diesen Zusammenhang beschreibt. Die Konstruktion dieses Kontexts lässt sich durch ein rekursives Verfahren beschreiben, weshalb es möglich ist, die Suche nach den interessanten Attributmengen und Beziehungen nebeneinander auszuführen. Die Ausnutzung der rekursiven Struktur macht die Verfahren wesentlich effizienter als die bisher üblichen "Apriori"-Verfahren. Die ersten Ergebnisse sind in dem Artikel "Endliche Hüllsysteme und ihre Implikationenbasen" zur Begutachtung eingereicht.

Besonderes Engagement in der Lehre:

SIGNAL - Sofortprogramm Informatik am Gymnasium - Nachqualifikation von Lehrkräften (Franz Forman, Klaus Kreisel, Hermann Puhmann, Bernd Schnitzer)

Deutschland will im Bereich Informationsgesellschaft einen Spitzenplatz in Europa einnehmen. Um dieses Ziel zu erreichen, müssen junge Menschen bereits in der Schule eine fundierte Ausbildung in Informatik erhalten. Konsequenterweise führt Bayern im Schuljahr 2004/05 das Pflichtfach Informatik an allen Gymnasien ein. Jedes der 400 bayerischen Gymnasien sollte ab diesem Zeitpunkt über mindestens eine Lehrkraft mit der Fakultas Informatik verfügen.

Der immense Bedarf an Lehrkräften kann nicht über die traditionelle Lehramtsausbildung an den Universitäten gedeckt werden. Aus diesem Grund wurde im Jahr 2000 das Projekt Nelli (Netzgestützter Lehrverbund zur Lehrerausbildung in Informatik) gestartet und ein elektronisches Angebot von speziell zugeschnittenen Studien- und Unterrichtsmaterialien entwickelt. Der Lehrstuhl Informatik 2 hat hierzu das Algorithmik-Modul beigetragen. Im September 2001 nahmen bayernweit im Rahmen von Pilotkursen 75 Lehrkräfte das Studium der Informatik auf, davon 15 in Erlangen. Im ersten Jahr lernen die Kursteilnehmer überwiegend zu Hause mit einer Kurs-CD, die das Unterrichtsmaterial enthält. Wöchentlich liefern sie auf elektronischem Wege ihre Übungen ab, die vom Tutor des Kurses korrigiert werden. Monatlich finden Treffen statt, die dem Erfahrungsaustausch dienen und auf denen der "Vorlesungsstoff" vertieft wird. Das zweite Studienjahr wird über wöchentliche Präsenzveranstaltungen an der Universität abgedeckt.

Der erfolgreiche Verlauf der Pilotkurse ermutigte zum Folgeprojekt SIGNAL (Sofortprogramm Informatik am Gymnasium - Nachqualifikation von Lehrkräften), mit dem im Jahr 2002 begonnen wurde. An den Universitäten Erlangen, München, Passau und Würzburg starteten 2002 weitere Kurse nach dem erprobten Verfahren, speziell an der FAU ein Kurs mit 25 Teilnehmern. Ein weiterer wird im Herbst 2003 folgen.

Im Vergleich mit den anderen Bundesländern wird Bayern nach Abschluss dieses Programms einen Spitzenplatz in Bezug auf den Stellenwert der Informatik an den Schulen einnehmen. Bisher hatten schon Nordrhein-Westfalen und Thüringen Informatik als Abiturfach angeboten.

Dissertationen:

Billing, Gunnar: „*Integrierte Hardware- und Softwareplanung flexibler Fertigungssysteme*“

Haworth, Brigid: "*Structural Coverage Criteria for Testing Object-Oriented Software*"

Studienarbeiten:

Büttcher, Stefan: „*Entwurf und Implementierung eines Genetischen Algorithmus zur Erstellung von Zeitplänen*“

Handl, Julia: „*Visualisierung Internet Queries using Ant-Based Heuristics*“

Otto, Stephan: „*Entwicklung und Vergleich genetischer Algorithmen zur Generierung redundanzreduzierter FIR-Filter für digitale Schaltkreise*“

Sander, Oliver: „*Kollokationen in medizinisch/biologischen Texten unter Einbeziehung syntaktischer Annotationen*“

Schlötzer, Stephan: „*Entwurf und Implementierung eines Navigationsmoduls für ein Offboard-Navigationssystem*“

Diplomarbeiten:

Birhälmer, Christina: „*Ausarbeitung und Implementierung der im Bildverarbeitungsteil des GREDEA-Systems notwendigen Algorithmen*“

Fühner, Tim: „*Untersuchung von Seeding-Mechanismen und der Entstehung von Building Blocks, um die Effizienz des GeLog-System zu verbessern*“

Landorff, Benjamin: „Auswahl und Implementierung von Methoden, um den Overfitting-Effect im GeLog-System zu vermeiden“

Mend, Volker: „Konzeption von modular und hierarchisch aufgebauten SPS-Programmen zur Steuerung von Platinenschneideanlagen“

Meyer, Markus: „Konfigurierbare graphische Benutzeroberflächen für Embedded System in Java“

Milovanovic, Igor: „Entwurf und Implementierung eines Persistenz-Schicht-Prototypen für eine Human Resource Software Applikation mit dem Microsoft .NET Framework“

Oster, Norbert: „Implementierung eines evolutionären Verfahrens zur Risikoabschätzung“

Reiber, Andreas: „Visuelle Spezifikation von Diagrammeditoren mit DIAGEN“

Seydel, Thomas: „Entwurf und Implementierung einer grafischen Benutzeroberfläche für einen Betriebssystemsimulator“

Strüber, Frank: „Unparsing von Diagrammen in DiaGen“

Tóth, Zóltan: „Visualization of Evolutionary Algorithms“

Ukis, Vladyslav: „Konzept für eine systemweite Konfiguration medizinischer Applikationen auf Basis der Syngo-Plattform“

Ványi, Róbert: „Evolutionary Design with Graph Grammers“

Winter, Frank: „Requirements Engineering im E-Commerce“

Veröffentlichungen:

V. Barthelmann. Inter-Task Register-Allocation for Static Operating Systems. In *Proc. Joint Conf. on Languages, Compilers and Tools for Embedded Systems & Software and Compilers for Embedded Systems*, pp. 149-154, June 2002.

F. Drewes, B. Hoffmann, M. Minas. Constructing shapely nested graph transformations. In *Proc. AGT'2002 (APPLIGRAPH Workshop on Applied Graph Transformation), Satellite Event to ETAPS 2002*, pp. 107-118, Grenoble, France, 2002.

F. Drewes, B. Hoffmann, M. Minas. Context-exploiting shapes for diagram transformations. In *Proc. 1st European Workshop on Diagrammatics and Design (D&D 2002)*, pp. 39-40, Bielsko-Biala, Poland, September 23-24, 2002.

M. Eitschberger, D. Leuchtenberger, I. Fischer. *Mädchen + Technik Praktikum 2001, Technisch-Naturwissenschaftliches Schnupperpraktikum für Schülerinnen*, Fraunhofer IRB Verlag, Stuttgart, ISBN 3-8167-6087-2, 2002.

T. Fühner, G. Kókai: Incorporating Linkage Learning into the GeLog Framework. In *Proc. 3rd Conf. of PhD Students in Computer Science CS²*, p. 35, Szeged, Hungary, July 1-4, 2002.

M. Gröbner, P. Wilke. A General View on Timetabling Problems. In E. Burke, P. De Causmaecker, editors, *Proc. of the 4th Int. Conf. on the Practice And Theory of Automated Timetabling, PATAT 2002*, pp. 221-227, Gent, Belgium, August 2002.

O. Köth, M. Minas. Structure, abstraction and direct manipulation in diagram editors. In M. Hegarty, B. Meyer, N. Narayanan, editors, *Diagrammatic Representation and Inference (Proc. 2nd Int. Conf., Diagrams 2002, Callaway Gardens, Georgia, USA)*, volume 2317 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag, pp. 290-304, April 2002.

G. Kókai. Expansion of the GeLog system by automatic adaptation of parameters. In *Proc. 2nd WSEAS Int. Conf. on Scientific Computation and Soft Computing*, Rethymna Beach, Rethymnon, Crete, July 7-14, 2002.

G. Kókai, B. Landorff. Ausarbeitung und Implementierung von Methoden um den Overfitting-Effekt im GeLog-System zu vermeiden. *FGML 2002, Treffen der Fachgruppe maschinelles Lernen der Gesellschaft für Informatik*, pp. 68-76, Hannover, 7.-9. Oktober 2002.

- G. Kókai, Z. Tóth, Sz. Zvada. An Experimental Comparison of Genetic and Classical Concept Learning Methods. In *Proc. Genetic and Evolutionary Computation Conf.*, New York, July 9-13, 2002.
- M. Minas. Concepts and realization of a diagram editor generator based on hypergraph transformation. *Science of Computer Programming*, 44(2):157-180, 2002.
- M. Minas. Specifying graph-like diagrams with DiaGen. In T. Mens, A. Schürr, G. Taentzer, editors, *Electronic Notes in Theoretical Computer Science*, volume 72. Elsevier Science Publishers. In *Proc. Int. Workshop on Graph-Based Tools (GraBaTs'02)*. Barcelona, October 7-8, 2002.
- M. Minas, F. Strüber. Unparsing of diagrams with DiaGen. In A. Corradini, H. Ehrig, H.-J. Kreowski, G. Rozenberg, editors, *Graph Transformation - Proc. 1st Int. Conf. on Graph Transformation, ICGT 2002*, volume 2505 of *Lecture Notes in Computer Science*. Springer-Verlag, pp. 302-316, Barcelona, October 7-12, 2002.
- M. Minas, G. Frey. Visual PLC-programming using signal interpreted Petri nets. In *Proc. American Control Conf. 2002 (ACC2002)*, pp. 5019-5024, Anchorage, Alaska, May 2002.
- L. Prechelt, B. Unger, M. Philippsen, W. F. Tichy. Two controlled experiments assessing the usefulness of design pattern documentation during program maintenance. *IEEE Transactions on Software Engineering*, 28(6):595-606, June 2002.
- L. Prechelt, G. Malpohl, M. Philippsen. Finding plagiarism among a set of programs with JPlag. *Journal of Universal Computer Science*, 8(11):1016-1038, November 2002.
- M. Philippsen. Internetwahlen: Demokratische Wahlen über das Internet? *Informatik-Spektrum*, 25(2):138-150, April 2002.
- O. Sander, H. Kirsch, I. Fischer. Extracting Collocations from Syntactically Annotated Corpora. *FGML 2002, Treffen der Fachgruppe maschinelles Lernen der Gesellschaft für Informatik*, Hannover, 7.-9. Oktober 2002.
- Z. Tóth. A Graphical User Interface for Evolutionary Algorithms. In *Volume of extended abstracts of the Conf. of PhD Students in Computer Science, CS²*, Szeged, Hungary, July 1-4, 2002.
- R. Ványi. Efficiently computable fitness functions for binary image evolution. In *Applications of Evolutionary Computing, Proc. of EvoWorkshops2002: EvoCOP, EvoIASP, EvoSTim*, Kinsale, Ireland, volume 2279 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 277-288, April 3-4, 2002.
- R. Ványi. Structural Image Description: An Evolutionary Approach. In *Volume of extended abstracts of the Conf. of PhD Students in Computer Science, CS²*, Szeged, Hungary, July 1-4, 2002.
- R. Veldema, C. J. H. Jacobs, R. F. H. Hofman, H. E. Bal: Object Combining: A new aggressive optimization for Object Intensive Programs, *ACM JavaGrande ISCOPE 2002 Conf.*, pp. 165-174, Seattle, USA, Nov. 2002.
- P. Wilke et al. Abschlussbericht Software für ein Risiko-Management-System. Technical Report, Computer Science Dept. 2, University Erlangen-Nuremberg, 2002.
- P. Wilke, M. Gröbner, N. Oster. A Hybrid Genetic Algorithm for School Timetabling. In B. McKay and J. Slaney, editors, *AI 2002: Advances in Artificial Intelligence*, volume 2557 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag, pp. 455-464, Canberra, Australia, December 2002.
- Sz. Zvada. LAG System: Attribute Grammar Specification via Inductive Learning. *FGML 2002 Treffen der Fachgruppe maschinelles Lernen der Gesellschaft für Informatik*, Hannover, 7.-9. Oktober 2002.

Gäste:

Prof. Dr. János Csirik, Universität Szeged, Ungarn

Prof. Sylvia Weber-Russell, PhD, University of New Hampshire, USA

Gastvorträge:

Prof. Dr. Bernd Hindel (Method Park AG, Erlangen): „*Methodisches zwischen Sprachen und Systemen*“, 28.06.02

Prof. Dr. Christian Jacob (Univ. of Calgary): „*Aus Graphen wird man klug*“, 28.06.02

Prof. Dr. Christian Jacob (Univ. of Calgary): „*Evolutionäres Design zum Schwärmen*“, 01.07.02

Prof. Dr. Benno Fuchssteiner (Univ. Paderborn): „*Computeralgebrasysteme: Weder zur Lehre noch zur Forschung tauglich?*“, 14.10.02

Vorträge (außer Workshop- und Konferenzvorträge):

M. Minas: „*Visuelle Programmiersprachen und Rapid Prototyping visueller Programmierumgebungen*“, Informatik-Kolloquium der Universität Kassel, 14.02.02

R. König: „*Berechnung von Implikationen*“, Theorettag Lutherstadt Wittenberg, 25.02.02

R. König: „*Reticoli di concetti*“, Folge von Vorträgen im Rahmen des Seminars für Lehrerfortbildung am Dipartimento di Matematica, Bologna, 15.03.02 bis 05.04.02

R. König: „*Concetti e implicazioni*“, Dipartimento di Matematica, Bologna, 20.03.02

R. König: „*I reticoli di concetti: un fondamento teorico per l'analisi di dati*“, Università di Cesena, 21.03.02

I. Fischer: „*Emmy-Noether: Historische und aktuelle Perspektiven*“, Emmy-Noether Festkolloquium, Emmy-Noether Gymnasium Erlangen, Teilnahme an der Podiumsdiskussion, 23.03.02

R. König: „*Reticoli di concetti*“, Politecnico di Milano, 27.03.02

M. Minas: „*Vergangenheit, Gegenwart und Zukunft von DiaGen*“, TU Berlin, 02.05.02

I. Fischer: „*Parsing Natural Languages*“, Vorlesung + Übung für Promotionsstudenten der Abteilung für Informatik, Department of Computer Science, Fakultät der Wissenschaften (Faculty of Science), University Szeged, im Rahmen des PPP-Programms, 21.05.-31.05.02

I. Fischer: „*Hilfe, mein Computer versteht mich nicht – Eine Einführung in die Verarbeitung natürlicher Sprache*“, Vorlesung + Übung auf der Informatica Feminale, 5. Sommerstudium für Frauen in der Informatik, Bremen, 09.09.02 bis 20.09.02

F. Forman: „*Was tut sich neues in der Informatik am Gymnasium – Unterricht und Weiterbildung*“, Universität Erlangen-Nürnberg, Erlangen, 14.10.02

M. Minas: „*Spezifikation und Generierung von Diagrammeditoren mit DiaGen*“, Informatik-Kolloquium der Universität Koblenz-Landau, 15.11.02

I. Fischer: „*Ada Lovelace - die erste Programmiererin der Welt?*“, Beitrag zur Veranstaltungsreihe Leitfossilien der Logik und Informatik: Vom Abakus zum Quantencomputer I, Bildungszentrum der Stadt Nürnberg, Cauchy-Forum-Nürnberg (CFN) e.V., 21.11.2002

M. Minas: „*Visuelle Sprachen und Rapid Prototyping visueller Programmier- und Modellierungsumgebungen*“, Informatik-Kolloquium der Universität der Bundeswehr, München, 22.11.02

M. Minas: „*Werkzeuge zur visuellen Planung und Programmierung*“, Informatik-Kolloquium der Universität der Bundeswehr, München, 09.12.02