Inhalt

Vorwort

Das Jahr 1990 stand für die Fachgruppe Informatik im Zeichen erheblicher personeller Belastungen. Infolge der Stellenzugänge aus dem Hochschulsonderprogramm sowie der Berufung von 2 C3-Kollegen auf Lehrstühle zu anderen Hochschulen waren zunächst 5 von 11 Professorenstellen vakant. Eine willkommene Unterstützung besonders in der Lehre erhielten wir von den Kollegen E. Speckenmeyer, L. Staiger und W. Thomas, die für Vertretungen gewonnen wurden. Es ist der raschen und konstruktiven Mitarbeit in 4 Berufungskommissionen zu verdanken, daß die Kollegen M. Jarke und B. Steffen zum Wintersemester 90/91 an die RWTH berufen werden konnten und mit zwei weiteren Kollegen die Verhandlungen aufgenommen wurden.

Auch die räumliche Situation der Fachgruppe bleibt weiterhin angespannt. Leider ist der geplante Erweiterungsbau wegen der schwierigen Haushaltslage des Landes zunächst dem Rotstift des Finanzministers zum Opfer gefallen. Dennoch darf dankend festgestellt werden, daß die Hochschule zur Unterbringung der beiden neuen Lehrstühle eine Fläche von 700 qm im Erweiterungsbau E2 bereitgestellt hat. So konnte insbesondere dem Wunsch nach räumlicher Nähe zu den übrigen Informatikeinheiten entsprochen werden. Sicherlich dürften damit noch nicht alle Raumwünsche der Fachgruppe befriedigt worden sein. Gehen wir davon aus, daß der Ausbau des Fachs Informatik noch nicht abgeschlossen ist und daß die neuen Lehrstühle das erfolgreiche Einwerben von Drittmittelprojekten fortsetzen, ergibt sich offensichtlich ein zusätzlicher Raumbedarf.

Als eine der jüngsten Disziplinen an unserer Hochschule hat die Informatik aufgrund ihrer rasanten Entwicklung und ihrer vielfältigen Anwendungsmöglichkeiten für nahezu alle an der RWTH vertretenen Fächer zunehmende Bedeutung als Strukturwissenschaft erlangt. Daher stellt ihre Integration, vor allem mit den Ingenieurwissenschaften, ein zentrales Anliegen unserer Fachgruppe dar. Die traditionell ohnehin engen Beziehungen zur Fakultät für Elektrotechnik konnten weiter verstärkt werden. Dies kommt auch darin zum Ausdruck, daß Herr Kollege Nagl an der Informatikgrundausbildung im Studiengang Elektrotechnik beteiligt wurde und daß Herr Kollege Spaniol und der Verfasser Zweitsitze in der dortigen Fakultät erhielten.

Darüber hinaus galt das Interesse der Fachgruppe auch dem industriellen Umfeld. Im Informatik-Kolloquium wurde eine Vortragsserie über berufliche Anwendungen eingerichtet. Hier stellen Vertreter der ortsansässigen Firmen verschiedene Arbeitsfelder der Informatik vor. Außerdem hielt Herr Dr. Hruschka von der Firma GEI im SS 90 die Vorlesung *"Methoden in Requirements-Engineering"*. Ferner wird die Fachgruppe bei der Pflege der Beziehungen zwischen Hochschule und Industrie durch den regionalen Industrie-Club Aachen (REGINA) unterstützt.

Der vorliegende Bericht zeigt, daß sich die Arbeit der Fachgruppe vor allem in einer Vielzahl von Forschungsprojekten manifestiert. Angesichts der raschen Entwicklung der jungen Wissenschaft Informatik wird deutlich, daß eine aktuelle Ausbildung unserer Studenten nur in Verbindung mit intensiver Forschung gewährleistet werden kann.

K. Indermark

(Sprecher der Fachgruppe Informatik)

Forschungsberichte der Lehreinheiten

Lehrstuhl für Informatik I

1 Personal des Lehrstuhls

- Leiter:

  Professor Dr. Jürgen Merkwitz

- Sekretariat:

  Helga Jussen
  Ulrike Simon

- Mitarbeiter:

  Dipl.–Inform. Christian Engel (bis 30. 9. 1990)
  Dipl.–Inform. Christian Franke (über Drittmittel)
  Dipl.–Inform. Ulrich Fröhlings (ab 3. 9. 1990 als wiss. HK)
  Dr. Michael Sonnenschein
  Dipl.–Inform. Carsten Weise

- Technischer Angestellter:

  Karl–Heinz Thevis

- Mathematisch-Technische Assistent(inn)en zur Ausbildung:

  Hien Pham
  Brigitte Schrey (bis 30. 8. 1990)
  Bozena Skorupa (ab 3. 9. 1990)
  Margarete Tilch
  Susan Tomaschewski (ab 3. 9. 1990)
  Iris van Nieuwenhofen

2 Projekte

2.1 Objektorientierte, parallele Programmiersprachen

Beteiligt: **C. Weise, U. Fröhlings (Dipl.)**

Wesentliches Ziel des Projektes ist die Entwicklung geeigneter Methoden zur Parallelisierung sequentieller, objektorientierter Programme. Dazu wurde von Herrn Fröhlings im Rahmen seiner Diplomarbeit ein paralleles Ausführungsmodell für C++ (pAC++) entwickelt.

Innerhalb des Projektes soll aufgrund der dabei gesammelten Erfahrung eine Prototyp-Implementierung auf einem Transputer–System durchgeführt werden. Die Prototyp-Implementierung besteht aus zwei Teilen: zum einen aus dem Compiler, der das sequentielle Programm in ein paralleles übersetzt, und zum anderen aus einem Laufzeitsystem für die parallele Abarbeitung des Programms mittels des Konzepts der virtuellen Zeit.

Für die Prototyp-Implementierung wurde eine einfache, objektbasierte Sprache definiert, bei der – um den Parser einfach zu halten – auf jede Art von „Syntactic Sugar“ verzichtet wurde. Außerdem wurden die in C++ als problematisch erkannten Konstrukte ausgeschlossen. Zu diesen zählen in der Hauptsache Konstrukte, die von einem zugrundeliegenden globalen Speicher ausgehen (wie z.B. die Zeigerarithmetik).

Hauptaufgabe des Compilers ist es, durch eine geeignete (statische) Datenflußanalyse zur Laufzeit mögliche Konflikte zwischen parallel arbeitenden Objekten zu erkennen und durch entsprechende Maßnahmen so weit wie möglich zu vermeiden.

Das Laufzeitsystem besteht hauptsächlich aus einer Implementierung des Konzepts der virtuellen Zeit (nach D. R. Jefferson). Hierzu braucht jedes Objekt einen Objektverwalter, der u.a. alte Zustände des Objekts speichert und falls nötig eine Rückkehr in alte Zustände vornimmt. Weiterhin müssen versendete und empfangene Nachrichten auf eine spezielle Weise verwaltet werden. Ihre Behandlung hängt von ihnen zugeordneten Zeitmarken ab. Für den Transport und das Routing der Nachrichten soll auf den Transputern das Betriebssystem Helios genutzt werden.

Die Implementierung des Prototyps soll 1991 im Rahmen mehrerer Diplomarbeiten durchgeführt werden.

2.2 Verifikation verteilter Echtzeitsysteme

Beteiligt: **C. Weise (Dipl.)**

Innerhalb des Projektes geht es um die Untersuchung von Methoden zur Verifikation verteilter Echtzeitanwendungen. Ziel ist dabei, diese Verifikation automatisch durchführen zu können.

Für eine solche Verifikation braucht man eine Beschreibung des Verhaltens des Echtzeitsystems sowie seiner Umgebung und eine Spezifikation der zeitlichen Anforderungen an das System. Für eine automatische Verifikation muß diese Spezifikation in einer entscheidbaren Echtzeitlogik angegeben werden. Die meisten bestehenden Echzeitlogiken haben jedoch das Problem, i.a. nicht entscheidbar zu sein. Eine der wenigen Ausnahmen ist RTTL (Real Time Temporal Logic) von Alur und Henzinger.

Im wesentlichen bestanden die bisherigen Arbeiten innerhalb des Projektes darin, eine geeignete Modellierung für Echtzeitanwendungen zu finden, die eine Verwendung entsprechender automatischer Verifikationsmethoden von Echtzeitlogiken zuläßt. Der momentane Stand dieser Bemühungen wurde am Tag der Informatik 1990 vorgetragen. Eine Zusammenfassung des Vortrags findet sich ebenfalls in dieser Schrift.

2.3 GINA: Eine objektorientierte Petrinetz–Sprache

Beteiligt: **M. Sonnenschein, K. Augustyniak (Dipl.), S. Staas (Dipl.), S. van Hauten (Dipl.)**

Mit GINA wurde ein Konzept paralleler Sprachen entwickelt, das im wesentlichen Eigenschaften objektorientierter, funktionaler und prozeßorientierter Sprachkonzepte so kombiniert, daß eine effiziente Implementierung der GINA–Sprache auf einem Multicomputer möglich wird. Im Rahmen des Projekts wurde zunächst eine operationelle Semantik für GINA entwickelt, die im wesentlichen auf interpretierten Petrinetzen mit Prioritätendefinitionen basiert. Eine solche Semantik beinhaltet einen sehr hohen Grad an implizitem Parallelismus, ermöglicht andererseits aber auch die Definition nichtdeterminierter Programme durch das Konzept der Stellen, auf die verschiedene Transitionen „parallel“ zugreifen können und damit in Konflikt geraten.

Eine solche Nichtdeterminierung ist auch in vielen anderen Sprachen mit asynchronem Parallelismus — beispielsweise durch „Guarded Commands“ — möglich. Sie ist jedoch insbesondere dann nicht erwünscht, wenn Programme, die unter sonst gleichen äußeren Bedingungen auf verschiedenen Prozessornetzen ausgeführt werden, unterschiedliche Resultate erzielen. Ein zentraler Aspekt des Projekts war daher die Angabe von Kriterien, mit deren Hilfe ein Programmierer diejenigen Stellen eines GINA–Programms ermitteln kann, an denen es sich nicht determiniert verhält.

Entscheidbare Kriterien für die Determinierung können auf der Grundlage einer abstrakten Interpretation der Netze, die auch zur Übersetzungs– und Analysezeit des dargestellten Programms verfügbar ist, angegeben werden. Diese Kriterien wurden zunächst allgemein für Netze mit Prioritätendefinition entwickelt und dann auf Objektnetze und Programmnetze, wie sie in GINA–Programmen vorkommen, übertragen. Hierbei zeigte sich, daß Prioritätendefinitionen in Netzen insbesondere für eine sinnvolle abstrakte Interpretation nur eingeschränkt verwendet werden dürfen.

Ein GINA–Programm kann zur Laufzeit auf einen sogenannten Objektspeicher zugreifen, der persistente Objekte beeinhaltet. Ausgehend von den genannten Determinierungskriterien konnte gezeigt werden, daß die Determinierung von GINA–Programmen unter den angegebenen Kriterien auch bei dynamisch modifizierten Programmnetzen und bei der Verwendung eines Objektspeichers erhalten bleibt.

Die Spezifikation der Semantik und die Behandlung der Determinierungskriterien für GINA wurde 1990 abgeschlossen. Da die operationelle GINA–Semantik bereits auf einem implementierungsnahen Niveau definiert wurde, kann hieraus eine Prototyp–Implementierung abgeleitet werden. Eine solche Implementierung wurde auf einem Transputer–System unter dem Betriebssystem Helios begonnen. Die Implementierung umfaßt einen Compiler, der zunächst unter UNIX erstellt wird, um dann auf Helios portiert zu werden. Ferner wird ein Laufzeitsystem entwickelt, das im wesentlichen die Funktionen des Erzeugens und Löschens von Objekten, der Speicher– und Prozessorverwaltung auf den einzelnen Rechnerknoten und der Kommunikation der Objekte untereinander realisiert. Der Objektspeicher wird im Prototyp als zentraler Server installiert. Die Implementierung erfolgt im wesentlichen im Rahmen von Diplomarbeiten.

2.4 Debugging von verteilten Systemen

Beteiligt: **C. Franke, D. Dieke (Dipl.)**

Parallelrechner auf der Basis des Transputers unter OCCAM finden derzeit immer größere Verbreitung. Der einfache Aufbau und damit relativ günstige Preis dieser Systeme täuscht jedoch oft darüber hinweg, daß die zusätzlichen Probleme beim Programmieren dieser Systeme nicht unerheblich sind.

Gegenüber sequentiellen Programmsystemen ergeben sich bei parallelen Systemen zusätzliche Schwierigkeiten bei der Fehlersuche. Das interne Verhalten (lokale Variable, Kontrollfluß) und die Interaktion (Nachrichtenaustausch) von mehreren parallel ablaufenden sequentiellen Prozessen sollten beobachtbar sein. Die bei sequentiellen Programmen übliche Methode, ein Programm mehrfach ablaufen zu lassen, um so den Fehler einzukreisen, ist nicht mehr praktikabel. Das Protokoll des Nachrichtenaustauschs führt dazu, daß ein Programm bei einem erneuten Ablauf trotz identischer Eingabedaten ein anderes Verhalten zeigen kann.

Im Rahmen dieses Projektes wurde ein Präprozessor für Programme der Programmiersprache OCCAM entworfen und implementiert, der die Beobachtung der logischen Kanäle erlaubt. Dieser Präprozessor erzeugt aus einem gegebenen OCCAM–Programm ein neues Programm, das gegenüber dem ursprünglichen Programm so erweitert ist, daß jeder logische Kanal zwischen den Prozessen durch einen zusätzlichen Monitorprozeß überwacht wird. Die Monitorprozesse sind dann in der Lage, die auf dem jeweils zugeordneten Kanal übertragenen Daten dem Programmierer zur Beobachtung zugänglich zu machen.

Der Präprozessor wurde von Mitte 1988 bis Anfang 1990 auf einem UNIX System in der Programmiersprache C implementiert. Parallel zu der Implementierung des

Präprozessors wurden theoretische Untersuchungen durchgeführt, die Aussagen über die prinzipiellen Grenzen der Beobachtung von OCCAM–Programmen liefern sollen. Da OCCAM im wesentlichen auf CSP basiert, wurde CSP als Ausgangspunkt der Überlegungen gewählt.

Eine *Monitorfunktion* ist eine Funktion auf CSP–Programmen, die diese so umformt, daß eine Beobachtung von internen Ereignissen von außen möglich wird. Dazu werden zu jedem internen Ereignis von der Monitorfunktion neue *Monitorereignisse* eingeführt, deren Auftreten in engem Zusammenhang mit den internen Ereignissen stehen müssen. In der Praxis werden die Monitorereignisse für den Programmierer sichtbar gemacht, beispielsweise durch Ausgaben auf einem Bildschirm. Hierin begründet sich die Zusatzbedingung an Monitorfunktionen, daß die einzelnen Monitorereignisse nur sequentiell hintereinander ausgeführt werden dürfen.

Eine Monitorfunktion heißt *korrekt* bezüglich einer CSP–Semantik, wenn die Semantik des ursprünglichen Programms gleich der Semantik des auf die ursprünglichen Ereignisse eingeschränkten erweiterten Programms ist. Für klassische CSP-Semantiken, beispielsweise die Trace/Failure Sets von Roscoe/Brookes, konnten korrekte Monitorfunktionen angegeben werden. Die Korrektheitseigenschaft geht jedoch für Semantiken, die das Zeitverhalten von Programmen berücksichtigen, verloren.

In diesem Zusammenhang ist zu untersuchen, wie gravierend das Programmverhalten von einer solchen Monitorfunktion verändert wird. Damit könnte geklärt werden, für welche Teilklassen von CSP-Programmen solche Monitorfunktionen noch brauchbar sind.

2.5 Sicherheit von Personal Computern

Beteiligt: **C. Franke, G. Hohpe (Dipl.), M. Steinberg (Stud. HK)**

Personal Computer unter dem Betriebssystem MS-DOS wurden ursprünglich für den reinen Einbenutzerbetrieb konzipiert. Es sind keinerlei Maßnahmen vorgesehen, die verhindern, daß Unbefugte die auf dem PC gespeicherten Daten und Programme lesen oder modifizieren. Der inzwischen sehr breite Einsatz von Personal Computern auch in sicherheitskritischen Bereichen, insbesondere bei den Geld– und Kreditinstituten im beleglosen Zahlungsverkehr, macht geeignete Schutzmaßnahmen dringend erforderlich.

Im Herbst 1989 wurde mit der Konzeption einer Erweiterung für das Betriebssystem MS-DOS begonnen. Dabei wurden die IT-Sicherheitskriterien berücksichtigt. Von der Sicherheitssoftware werden im wesentlichen folgende Sicherheitsfunktionen zur Verfügung gestellt:

- Zugangskontrolle: Identifizierung und Authentisierung von Benutzern.
- Zugriffskontrolle: Verwaltung und Prüfung von Zugriffsrechten auf Datenobjekte.

  - Benutzerbestimmbare Zugriffskontrolle: Die Zugriffsrechte werden vom Eigentümer der Datenobjekte vergeben.
  - Festgelegte Zugriffskontrolle: Jedem Datenobjekt ist eine Sicherheitsstufe zugeordnet. Alle Sicherheitsstufen bilden eine Halbordnung. Das System sorgt dafür, daß ein Informationsfluß nur von einer niedrigeren zu einer höheren oder zwischen gleichen Sicherheitsstufen möglich ist.
- Datenverschlüsselung: Alle Daten werden auf Magnetplatte in verschlüsselter Form abgelegt. Ein Diebstahl der Magnetplatte ist zwecklos. Als Verfahren wird derzeit DES eingesetzt.
- Protokollierung: Alle sicherheitsrelevanten Ereignisse können vom Betriebssystem protokolliert werden.

Die Implementierung der MS-DOS-Erweiterung wurde Mitte 1990 begonnen und wird voraussichtlich bis Mai 1991 abgeschlossen sein.

Reine Softwarelösungen können zwar im Hinblick auf die *Funktionalität* der Sicherheitsmaßnahmen ausreichen, jedoch lassen sich alle Softwarekomponenten leicht durch direkten Zugriff auf die Hardware umgehen. Um dies zu verhindern und damit einen hohen Grad der *Qualität* der Sicherheitsmaßnahmen zu erreichen, ist Anfang 1990 ein *Sicherheits-PC* konzipiert worden, der im wesentlichen folgende Eigenschaften besitzt:

- Getrennte 80386 Prozessoren für Betriebssystem– und Anwendersoftware mit jeweils eigenem Hauptspeicher. Der Betriebssystemprozessor besitzt die Kontrolle über die vom Anwenderprozessor erreichbaren Adreßbereiche.
- Hardwareunterstützung für die DES Verschlüsselung.
- Sicherheitsgehäuse

Im Laufe des Jahres 1991 soll ein Prototyp des Sicherheits-PC in Zusammenarbeit mit der Firma Van den Berg Elektronik GmbH in Herzogenrath realisiert werden.

2.6 AWKward: Eine Erweiterung der Programmiersprache AWK

Beteiligt: **C. Franke, M. Schopen (Dipl.)**

AWK ist eine von A. **A**ho, P. **W**einberger und B.W. **K**ernighan entwickelte *pattern matching and processing*-Sprache. Die Sprache entstand aus der Bestrebung, die Unix-Tools *grep*, *lex*, *sed* sowie diverse C-Sprachelemente miteinander zu verbinden.

AWK verarbeitet als Eingabedateien allerdings nur ASCII-codierte Textdateien. Jede Zeile wird als ein Datensatz aufgefaßt, welcher durch ein vorgegebenes Trennzeichen in Felder zerlegt wird. Im Rahmen des Projektes AWKward (**AWK** **w**ith **a**dvanced **r**ecord **d**escriptions) wurde AWK so erweitert, daß auch anders codierte und strukturierte Daten bearbeitet werden können.

Es wurde eine Format-Spezifikations-Sprache entwickelt, mittels der die Struktur der einzelnen Datensätze und -felder in einer der Programmiersprache C ähnlichen Form beschrieben werden kann. Sowohl das Format der Eingabe-Dateien als auch das der Ausgabe-Dateien kann beschrieben werden.

Zu jedem Datenfeld wird ein Codierungstyp angegeben, so daß verschiedene Codierungen verarbeitet werden können. Neben fest implementierten Standardcodierungen (EBCDIC, BCD) besteht die Möglichkeit, eigene Codierungen durch Angabe der erforderlichen Eingabe- und Ausgabe-Konvertierroutinen zu spezifizieren.

Die Implementierung erfolgte von Mitte bis Ende 1990 in der Programmiersprache C auf einem UNIX-System. Dabei wurde die GNU Version des AWK als Grundlage verwendet.

Das Projekt wurde von der Firma Van den Berg Elektronik GmbH in Herzogenrath unterstützt.

3 Diplomarbeiten, Veröffentlichungen und Vorträge

3.1 Diplomarbeiten

Beißmann, Dieter: *Ein konfigurierbares Programm zur Variantenanalyse komplexer Produkte,* (Zweitgutachten)

Dieke, Detlev: *Ein Präprozessor zur Beobachtung der Kommunikation über die logischen Kanäle der Programmiersprache OCCAM*

Fröhlings, Ulrich: *Entwicklung eines parallelen Ausführungsmodells für C++*

van Hauten, Swen: *Dynamisches Mapping von gewichteten Objektbäumen auf Prozessornetze*

Klersy, Winfried: *Mathematische Optimierungsverfahren bei der Bestimmung von Zeitkalkulationsformeln für die Herstellung von Gußstücken,* (Zweitgutachten)

Kurth, Hermann–Josef: *Entwicklung eines Programms zur Prozeßoptimierung mittels der Evolutionsstrategie,* (Zweitgutachten, Prof. Dr. W. Michaeli)

Mehta, Peter: *Konzipierung und Erstellung eines Softwarepakets zur Ansteuerung einer hydraulischen Stempeleinheit an einem Innenmischer zur diskontinuierlichen Aufbereitung von Kautschukmischungen,* (Zweitgutachten)

Reineke, Raymond : *Untersuchung des Einflusses von Freiheitsgraden in der Leistung eines mechanischen Parksystems bei variabler Gestaltung der Zielfunktion,* (Zweitgutachten, Referent Prof. Dr. Pleßmann)

Reiter, Matthias: *Erstellung eines Programms zur Simulation von Risikoprofilen bei partieller Information,* (Zweitgutachten)

Schopen, Manfred: *Eine Beschreibungssprache für Dateiformate als Erweiterung für die Programmiersprache AWK*

Simon, Sabine: *Terminplanung von Fertigungsaufträgen unter besonderer Berücksichtigung der zeitlichen Verfügbarkeit und der begrenzten Kapazität von Ressourcen,* (Zweitgutachten)

3.2 Veröffentlichungen und Vorträge

- M. Sonnenschein: Ein objektorientiertes und datengesteuertes Programmierkonzept für Multicomputer auf der Grundlage dynamischer Petrinetze. Interner Bericht des Lehrstuhls für Informatik I, Juni 1990
- M. Sonnenschein: Gina: An object oriented, parallel language based on Petri nets. Schriften zur Informatik und angewandten Mathematik, Nr. 144, Oktober 1990
- M. Sonnenschein: Ein objektorientiertes und datengesteuertes Programmierkonzept für Multicomputer auf der Grundlage dynamischer Petrinetze. Vortrag im
  - Informatik Kolloquium der RWTH Aachen am 5. April 1990
  - Informatik Kolloquium der Universität Oldenburg am 11. Juli 1990
  - Informatik Kolloquium der TU München am 24. Juli 1990
  - Mathematischen Kolloqium der Ludwig Maximilians Universität München am 25. Januar 1991
- C. Weise: Verifikation periodischer Prozesse in statischen, verteilten Echtzeitsystemen
  Vortrag an der RWTH Aachen zum Tag der Informatik 1990

Lehrstuhl für Informatik II

Personal

- Dozenten:

  Prof. Dr. rer. nat. Klaus Indermark (Lehrstuhlinhaber)
  Priv.-Doz. Dr. rer. nat. Ewald Speckenmeyer
  (Vertretung des Lehr- und Forschungsgebiets Informatik im SS 90)
  Prof. Dr. rer. nat. Bernhard Steffen
  (Lehr- und Forschungsgebiet Informatik, ab 14. 09. 90)
  Prof. Dr. rer. nat. Wolfgang Thomas (Lehrauftrag im WS 89/90)
  Priv.-Doz. Dr. Heiko Vogler (ab 24.10.90)

- Sekretariat:

  Frau Ruth Jansen
  Frau Carla Meckler

- Wissenschaftliche Mitarbeiter:

  Akademischer Rat z. A. Dr. Heiko Vogler
  (beurlaubt ab 01. 10. 90 zur Vetretung einer C3-Professur an der Universität Ulm)
  Dipl.-Inform. Heinz Faßbender (DFG, ab 01. 06. 90)
  Dr. Thilo Hafer (DFG, bis 31. 03. 90)
  Dipl.-Inform. Frank Herrmann (DFG, bis 30. 09. 90)
  Dr. Uschi Heuter (DFG, bis 31. 03. 90)
  Dr. Friedhelm Hinz (bis 30. 06. 90)
  Dr. Herbert Kuchen
  Dipl.-Inform. Detlef Lippert (bis 14. 06. 90)
  Dr. Rita Loogen
  Dipl.-Inform. Andreas Maaßen
  Dipl.-Inform. Andreas Potthoff (bis 30. 04. 90)
  Dipl.-Inform. Udo Wermuth (DFG, bis 30. 04. 90)
  Dipl.-Inform. Katia Gladitz (Wiss. Hilfskraft, ab 15. 10. 90)
  Ldo. Mat. Yolanda Ortega Mallén (Wiss. Hilfskraft, 15. 01. – 15. 03. 90)

- Mathematisch-Technischer Assistent:

  Herr Arnd Gehrmann

Allgemeines

Die Berufung von W. Thomas auf den Lehrstuhl für Theoretische Informatik an der Universität Kiel ergab für unseren Lehrstuhl eine Reihe von persönlichen Veränderungen. Erfreulicherweise konnte das Lehr- und Forschungsgebiet Informatik bereits zum 14. 09. 1990 durch B. Steffen wiederbesetzt werden, nachdem W. Thomas und E. Speckenmeyer durch Vertretungen in den beiden vorhergehenden Semestern für einen nahtlosen Übergang gesorgt hatten.

Für die Gruppe der wissenschaftlichen Mitarbeiter darf ich zunächst den erfolgreichen Abschluß des Habilitationsverfahrens von H. Vogler erwähnen. Der Erfolg seiner wissenschaftlichen Arbeit wird auch sichtbar an der Bewilligung des von ihm beantragten DFG-Projekts, für das H. Faßbender eingestellt wurde, sowie an seiner Vertretung einer C3-Professur an der Universität Ulm.

Demgegenüber hatten der Wechsel im Lehr- und Forschungsgebiet Informatik sowie das Auslaufen von DFG-Projekten den erheblichen Schwund von 7 Mitarbeitern zur Folge.

Als Gastforscher waren Frau Y. Ortega Mallén aus Madrid und Frau K. Gladitz aus Dresden tätig.

Entsprechende Veränderungen sind bei den Forschungsprojekten zu verzeichnen. Besondere Erwähnung verdient die im 3. Jahr vom DAAD im Rahmen der "Acciones Integradas" geförderte Kooperation mit der Madrider Forschungsgruppe von Mario Rodríguez Artalejo. Die Integration logischer und funktionaler Programmierung konnte durch Entwicklung und Implementierung der Sprache BABEL wesentlich verbessert werden. Zum Abschluß der "Acción Integrada" wurde ein Workshop beider Arbeitsgruppen mit weiteren Gästen in Granada veranstaltet.

Durch die Berufung von B. Steffen wird ein neuer Forschungsschwerpunkt entstehen. Dabei sollen Methoden zur Entwicklung, Verifikation, Analyse und Transformation sequentieller Programme und verteilter Systeme untersucht werden.

K. Indermark

Forschungsprojekte:

Projekt:

Integration funktionaler und logischer Programmierung in BABEL

Beteiligt: **H. Faßbender, K. Indermark, H. Kuchen, R. Loogen, W. Thomas, H. Vogler**

(in Zusammenarbeit mit den Arbeitsgruppen von D. de Frutos Escrig und M. Rodríguez Artalejo, Universidad Complutense de Madrid, Madrid, Spanien, sowie J.-J. Moreno Navarro, Universidad Politécnica de Madrid, Madrid, Spanien)

Förderung: **DAAD**

Für die funktional-logische Programmiersprache BABEL wurde eine OR-parallele Implementierung auf einem Transputer-Mehrprozessorsystem durchgeführt. Außerdem wurde eine abstrakte Maschine für eine AND-parallele BABEL-Implementierung auf einem lose gekoppelten Mehrprozessorsystem entwickelt. Letztere erlaubt mehr Parallelität als der übliche eingeschränkte AND-parallele Ansatz von Hermenegildo.

Zur Realisierung einer 'lazy narrowing'-Strategie wurde eine Transformation in sogenannte uniforme BABEL-Programme entwickelt. Uniforme BABEL-Programme erlauben eine Einteilung von Funktionsparametern in benötigte und nicht-benötigte Argumente und somit eine natürliche Erweiterung von 'innermost' auf 'lazy' Implementierungen funktional-logischer Programme.

Die Untersuchungen über BABEL mit Einschränkung auf primitive Rekursion wurden fortgesetzt und führten zu einem neuen DFG-Projekt, über das an späterer Stelle berichtet wird.

Der Typinferenz-Algorithmus zur Ableitung regulärer Typen für Prädikate eines Logik-Programmes konnte verbessert werden. Dazu wird ein Logik-Programm in ein reguläres Logik-Programm übersetzt, welches zu einem Top-down-Automaten äquivalent ist und reguläre Typen als reguläre Baumsprachen beschreibt.

Forschungsaufenthalte spanischer Kollegen in Aachen:

Yolanda Ortega Mallén	15. 01. 90 – 15. 03. 90
David de Frutos Escrig	02. 03. 90 – 08. 03. 90
Juan Carlos Gonzalez Moreno	17. 06. 90 – 30. 06. 90
Maria Teresa Hortala Gonzalez	17. 06. 90 – 30. 06. 90
Juan José Moreno Navarro	17. 06. 90 – 30. 06. 90
Mario Rodríguez Artalejo	17. 06. 90 – 30. 06. 90

Workshop in Granada:

Zum Abschluß dieser "Acción Integrada" wurde vom 23. 09. 90 – 29. 09. 90 ein Workshop in Granada (Spanien) veranstaltet, an dem außer den beteiligten Arbeitsgruppen aus Madrid und Aachen weitere Wissenschaftler aus Deutschland und Spanien mit Vorträgen teilnahmen. Das Programm findet sich mit Kurzfassungen der Vorträge in dem Abschnitt über besondere Veranstaltungen.

Fertiggestellte Arbeiten:

H. Kuchen, R. Loogen, J. J. Moreno Navarro, M. Rodríguez Artalejo: *Graph-Based Implementation of a Functional Logic Language,* Proceedings European Symposium on Programming (ESOP), Copenhagen, Springer LNCS, Vol. 432 (1990), 271 – 290

R. Loogen: *Stack-based Implementation of Narrowing,* Aachener Informatik-Bericht Nr. 90-4, RWTH Aachen, 1990

Y. Ortega Mallén, D. de Frutos Escrig: *A Complete Proof System for Timed Observations,* Aachener Informatik-Bericht Nr. 90-20, RWTH Aachen, 1990

J. J. Moreno Navarro, H. Kuchen, R. Loogen, M. Rodríguez Artalejo: *Lazy Narrowing in a Graph Machine,* Proceedings 2nd International Conference on Algebraic and Logic Programming, Nancy, Frankreich, Springer LNCS, Vol. 463 (1990), 298 – 317, ebenso erschienen als Aachener Informatik-Bericht Nr. 90-11, RWTH Aachen, 1990

W. Hans: *Entwurf einer abstrakten Maschine für eine UND-parallele Implementierung der funktionalen Logik-Programmiersprache BABEL auf einem lose gekoppelten Multicomputer-System,* Diplomarbeit

A. Schleicher: *OR-parallele Implementierung der funktional-logischen Sprache BABEL,* Diplomarbeit

Projekt:

Funktionale Programmierung mit primitiver Rekursion

Beteiligt: **H. Faßbender, K. Gladitz, F. Herrmann, K. Indermark, H. Vogler**

(in Zusammenarbeit mit Z. Fülöp und S. Vagvölgyi, Akademie der Wissenschaften, Szeged (Ungarn))

Förderung: **DFG**

In diesem Projekt werden funktionale Programme mit primitiver Rekursion untersucht. Es wird das Ziel verfolgt, die spezielle Rekursionsstruktur für eine effiziente Implementierung zu nutzen. Hierbei ist der enge Zusammenhang zwischen primitiv rekursiven Funktionsdefinitionen, baumübersetzenden Systemen (***tree transducer***) und attributierten Grammatiken von besonderer Bedeutung.

Im Berichtsjahr wurde für SRDL*, eine funktionale Modellsprache mit primitiver Rekursion, eine iterative Reduktionssemantik entwickelt [Vog]. Für den Fall simultaner Rekursion können dadurch erhebliche Effizienzsteigerungen erzielt werden. Sind außerdem die nicht rekursiven Funktionsargumente konstant, so läßt sich durch eine boolesche Tabellensteuerung die Berechnung noch stärker optimieren [IndKlaVog].

Aus theoretischen Resultaten zur Auflösung von simultaner Rekursion mit Termschachtelung in Parameterpositionen in (einfache) primitive Rekursion wurde ein Compiler entwickelt [Küh]. Die Beschreibung der Übersetzungsfunktionen auf drei Abstraktionsniveaus gibt ein hinreichendes Indiz für die prinzipielle Korrektheit des Compilers.

Weiterhin wurde der formale Zusammenhang zwischen primitiv rekursiven Funktionsdefinitionen und attributierten Grammatiken untersucht [FülHerVagVog]. Die Ergebnisse betreffen die Berechnungsstärke beider Systeme in bezug auf freie Interpretationen. Ferner wurden primitiv rekursive Schemata mit Parametern von höheren funktionalen Level definiert. Es wurde gezeigt, daß auf diese Weise auch nicht primitiv rekursive Baumfunktionen definiert werden können [HerVog].

Funktionsdefinitionen nach dem Konzept der Syntaxsteuerung waren Ausgangspunkt von drei Untersuchungen.

1. Die aus der Automatentheorie bekannte Charakterisierung der call-by-name Auswertung durch ***checking-tree nested-stack machines*** wurde zur Implementierung der call-by-need Auswertung durch ***attributed checking-tree nested-stack machines*** verallgemeinert [FaßVog, Faß]. Es wurde ein Bericht vorbereitet , in dem die ***interpreter-based*** Implementierung auf ***checking-tree nested-stack transducers*** mit einer ***compiler-based*** Implementierung auf ***runtime stack*** Maschinen verglichen wird.

2. Für die Teilklasse der sogenannten L-primitiv-rekursiven Schemata wurde auf der Grundlage von zwei theoretischen Resultaten eine ***interpreter-based*** Implementierung formalisiert und realisiert, die auf einer Attributauswertungstechnik beruht [Gli].

3. Zur Auswertung spezieller Eigenschaften dieser Funktionsdefinitionstechnik wurde eine Implementierung auf einer abstrakten parallelen ***shared-memory*** Maschine formalisiert [Rob]. Hier läßt sich bei geringen Kosten eine Entscheidung über die Notwendigkeit der Auswertung von Parametern treffen.

Fertiggestellte Arbeiten:

[Vog] H. Vogler: *Funktionale Programmierung mit primitiver Rekursion – Formale Modelle zur Reduktionssemantik,* Habilitationsschrift, RWTH Aachen, 1990

[IndKlaVog] K. Indermark, H. A. Klaeren, H. Vogler: *Computational Aspects of Structural Recursion,* eingeladener Vortrag, STACS '90, Rouen, Frankreich, 1990

[Küh] A. Kühnemann: *Transformation strukturell-rekursiver Programme in Normalform,* Diplomarbeit

[FülHerVagVog] Z. Fülöp, F. Herrmann, S. Vagvölgy, H. Vogler: *Tree Transduces with External Functions,* Aachener Informatik-Bericht Nr. 90-27, RWTH Aachen, 1990

[HerVog] F. Herrmann, H. Vogler: *Primitive Recursive Programs for Defining Tree Functions,* Aachener Informatik-Bericht Nr. 90-26, RWTH Aachen, 1990

[FaßVog] H. Faßbender, H. Vogler: *A Call-by-need Implementation of Syntax Directed Functional Programming,* Aachener Informatik-Bericht Nr. 90-22, RWTH Aachen, 1990

[Faß] H. Faßbender: *Implementierung der call-by-need Auswertungsstrategien für Macro Tree-to-String Transducer auf Nested-Stack Maschinen mit geschachtelten Ausgabebändern,* Diplomarbeit

[Gli] L. Glimm: *Implementierung von primitiv rekursiven Schemata auf attributierten Pushdown-Maschinen,* Diplomarbeit

[Rob] H. Robens: *Abstrakte Graphreduktionsmaschinen zur Implementierung funktionaler Programmiersprachen mit Striktheitsanalyse,* Diplomarbeit

Projekt:

Effiziente Implementierung einer funktional logischen Programmiersprache bei Einschränkung auf primitive Rekursion

Beteiligt: **H. Faßbender, H. Vogler**

Förderung: **DFG, DAAD**

In diesem Projekt wird die funktional logische Programmiersprache prim-rek BABEL definiert. Diese Sprache erhält man, indem man die Rekursion im funktionalen Anteil der funktional logischen Programmiersprache BABEL auf primitive Rekursion über frei erzeugten Datenstrukturen einschränkt. Im Rahmen des Projekts werden die Phänomene der Narrowing-Semantik von prim-rek BABEL studiert. Ferner soll untersucht werden, ob man durch die eingeschränkte Form der Rekursion effizientere Implementierungen als für BABEL angeben kann. Die beschriebene Einschränkung ist legitim, da in vielen Anwendungsfällen nur primitiv-rekursive Programme geschrieben werden.

Bei diesem Projekt wird von einer Implementierung der primitiv rekursiven funktionalen Sprache $sd - SRDL^*$ auf einer abstrakten Maschine ausgegangen; bei $sd -$

$SRDL^*$ ist die Rekursion noch stärker eingeschränkt und erlaubt nur die syntaxgesteuerte Übersetzung von Objekten. Diese Implementierung wurde als erste Arbeit dieses Projekts in einem technischen Bericht formalisiert (siehe hierzu [FaßVog]). In einem weiteren Schritt wird nun diese abstrakte Maschine so um logische Komponenten angereichert, daß auf dieser Maschine dann die um logische Aspekte erweiterte Programmiersprache $sd - SRDL^*$ implementiert werden kann (siehe hierzu [RodVog]). Schließlich wird untersucht, wie man diese Implementierung geeignet für prim-rek BABEL modifizieren kann und wie sich diese Implementierung zu bereits existierenden Implementierungen von BABEL verhält.

Fertiggestellte Arbeiten:

[RodVog] M. Rodríguez Artalejo, H. Vogler: *Note on A Narrowing Machine for Syntax Directed BABEL,* Aachener Informatik-Bericht Nr. 89-19, RWTH Aachen, 1989

[FaßVog] H. Faßbender, H. Vogler: *A Call-by-need Implementation of Syntax Directed Functional Programming,* Aachener Informatik-Bericht Nr. 90-22, RWTH Aachen, 1990

Projekt:

Parallele Implementierung funktionaler Sprachen

Beteiligt: **H. Kuchen, R. Loogen**

In diesem Projekt zur Implementierung einer Miranda-ähnlichen Sprache auf einem Transputer-Multicomputer-System wurde im wesentlichen der bestehende Compiler erweitert und verbessert. Fertiggestellt wurden der Scanner und Parser für Miranda, die Assemblerimplementierung der abstrakten Maschine und ein Optimierer. Weiterhin wurden verteilte applikative Arrays in die Sprache integriert und verschiedene Lastverteilungsstrategien untersucht. Die erweiterte Technik der 'evaluation transformer' zur Steuerung der Auswertung von Datenstrukturen wurde in die OCCAM-Implementierung integriert.

Fertiggestellte Arbeiten:

H. Kuchen, A. Wagener: *Comparison of Dynamic Load Balancing,* Aachener Informatik-Bericht Nr. 90-5, RWTH Aachen, 1990
erscheint auch in: Workshop on Parallel and Distributed Processing, Sofia, 1990, Elsevier Publishers

H. Kuchen, G. Geiler: *Distributed Applicative Arrays,* Proceedings 2nd International Workshop on Implementation of Functional Languages on Parallel Architectures, Nijmegen, Technical Report (University of Nijmegen) no. 90-16 (1990), 223 – 235

H. Kuchen, R. Loogen: *Parallele Implementierung einer funktionalen Programmiersprache auf einem Transputer-Mehrprozessor-System,* Proceedings Transputer-Anwender-Treffen 1989 (TAT '89), Aachen, Springer IFB 237, 21 – 33

H. Kuchen, A. Wagener: *Lastverteilung auf Multicomputern,* Proceedings Transputer-Anwender-Treffen 1990, Aachen, Abstraktband erschienen als Bericht des Instituts für Physiologie der RWTH Aachen

H. Denker: *Zwischencodeoptimierung bei der parallelen Implementation einer funktionalen Sprache* (Diplomarbeit)

G. Geiler: *Implementierung verteilter applikativer Arrays* (Diplomarbeit)

K. Händeler: *Effiziente Implementierung der parallelen abstrakten Maschine PAM in Transputerassembler* (Diplomarbeit)

G. Hogen: *Extension of the Parallel Abstract Machine (PAM) with Evaluators,* Workshop on the Integration of Functional and Logic Programming, Granada, September 1990

A. Wagener: *Lastverteilungsstrategien und Netzwerktopologien für eine Realisierung der parallelen abstrakten Maschine PAM* (Diplomarbeit)

R. Wischnewski: *Implementierung einer Übersetzung von MIRANDA in einen getypten λ-Kalkül als Frontend eines parallelisierenden Compilers* (Diplomarbeit)

Projekt:

Stack-basierte Implementierung von Narrowing

Beteiligt: **R. Loogen**

Durch Erweiterung einer Reduktionsmaschine für funktionale Sprachen um Mechanismen zur Durchführung von Unifikation und Backtracking wurde eine stack-basierte Narrowing-Maschine entwickelt und in Miranda implementiert. Auf der Basis dieser Maschine wurde ein Programmsystem für die funktional-logische Sprache BABEL in Miranda geschrieben. Zur Zeit wird dieses System in C implementiert.

Fertiggestellte Arbeiten:

R. Loogen: *Stack-controlled Implementation of Narrowing,* Aachener Informatik-Bericht Nr. 90-4, RWTH Aachen 1990

St. Winkler: *Implementation of a BABEL Programming Environment,* Workshop on the Integration of Functional and Logic Programming, Granada, September 1990

Projekt:

Methodik der funktionalen Programmierung

Beteiligt: **A. Maaßen**

In diesem Projekt sollen Methoden zur systematischen Erstellung funktionaler Programme entwickelt und die sich ergebenden Auswirkungen auf die Anforderungen an die Pragmatik deklarativer Programmiersprachen untersucht werden.

Für funktionale Programmiersprachen wurden Datenstrukturen und Funktionen höherer Ordnung entwickelt, die die Entwicklung gut verständlicher Programme mit einem hohem Maß an potentieller Parallelität auf dem Datenlevel besser unterstützen als Listen, die heute noch der Standard in funktionalen Sprachen sind.

In einer weiteren Arbeit wurde die Verwendbarkeit funktionaler Programmiersprachen für den Compilerbau untersucht. Im Zuge dieser Arbeit entstand die Aufgabenstellung für ein Software-Praktikum, in dem ein Compiler und ein Laufzeitsystem für eine kleine Programmiersprache in der funktionalen Sprache Miranda[1] implementiert werden. Eine Beschreibung dieses Praktikums wurde als Technischer Bericht veröffentlicht.

Nächstes Ziel des Projektes ist die Entwicklung eines Typkonzeptes für funktionale Sprachen, das Funktionen höheren Typs und Polymorphie mit Vererbung und Überladung verbindet.

Fertiggestellte Arbeiten:

A. Maaßen: *Parallel Programming with Data Structures and Higher Order Functions,* Überarbeitung vom Aachener Informatik-Bericht Nr. 89-18, RWTH Aachen, 1989, zur Veröffentlichung eingereicht

A. Maaßen: *Compilerentwicklung in Miranda – ein Praktikum in funktionaler Programmierung,* Aachener Informatik-Bericht Nr. 90-14, RWTH Aachen, 1990

[1]Miranda ist eingetragenes Warenzeichen von Research Software Ltd.

Projekt:

Axiomatisierung eines semantischen Modells mit Zeit für nebenläufige Prozesse

Beteiligt: **Y. Ortega Mallén, D. de Frutos Escrig**

Für ein semantisches Modell nebenläufiger Prozesse, welches auch für Realzeitsysteme geeignet ist, wurde ein vollständiges Beweissystem entwickelt. Die Modellierung von Zeit erfolgt dadurch, daß Aktionen eine Zeitdauer zugeordnet wird. Nebenläufigkeit wird durch Multimengen von Aktionen repräsentiert. Die Zeit ist global und diskret.

Fertiggestellte Arbeiten:

Y. Ortega Mallén, D. de Frutos Escrig: *A Complete Proof System for Timed Observations,* Aachener Informatik-Bericht, Nr. 90-20, RWTH Aachen 1990

Y. Ortega Mallén: *En Busca del Tiempo Perpido,* Dissertation, Fac. CC. Matemáticas, Universidad Complutense de Madrid, 1990

Projekt:

Vollständige interprozedurale Programmlaufzeitoptimierung

Beteiligt: **J. Knoop, O. Rüthing, B. Steffen**

Förderung: **DFG**

Kennzeichnend für die Situation der globalen Programmlaufzeitoptimierung ist eine Vielzahl vorwiegend heuristischer Verfahren, die unabhängig voneinander spezielle Optimierungszielsetzungen verfolgen. Das hat zwei schwerwiegende Nachteile:

- Interaktionen der Optimierungen werden nicht berücksichtigt,
- Vollständigkeit bzgl. einer zugrundegelegten Programmsemantik wird nicht erreicht.

Vor diesem Hintergrund wurde in der Dissertation von B. Steffen ein intraprozedurales Modellverfahren entwickelt. Es kombiniert die Optimierungszielsetzungen

a) Vorziehen (schleifen-) invarianter Berechnungen und

b) Beseitigen gemeinsamer Teilausdrücke und behandelt sie

c) bzgl. der Herbrand-Interpretation vollständig.

Unser Ziel ist es, dieses Konzept weiterzuverfolgen. Dazu wollen wir das Modellverfahren in 3 Richtungen ausdehnen:

- auf weitere Optimierungszielsetzungen,
- auf spezielle Interpretationen,
- auf Prozedurkonzepte.

Ein greifbares Ziel ist die einheitliche und vollständige Ausdehnung des Modellverfahrens hinsichtlich Konstantenausbreitung und -faltung und Reduktion der Operatorstärke, sowie die vollständige Behandlung von Programmen ohne Prozedurschachtelung.

Fertiggestellte Arbeiten:

J. Knoop, B. Steffen: *Finite Interprocedural Constants,* Submitted for TACS '91

B. Steffen, J. Knoop: *Finite Constants: Characterizations of a New Decidable Set of Constants,* Accepted for TCS

B. Steffen, J. Knoop, O. Rüthing: *The Value Flow Graph: A Program Representation for Optimal Program Transformations,* 3rd ESOP, Copenhagen, Denmark, Springer LNCS, Vol. 432 (1990), 389 - 405 – An extended version is available as: Bericht Nr. 9004, Institut für Informatik und Praktische Mathematik, Christian-Albrechts-Universität Kiel, 1990

B. Steffen, J. Knoop, O. Rüthing: *Efficient Code Motion and an Adaption to Strength Reduktion,* Accepted for TAPSOFT '91

Projekt:

AC/P–AA

Beteiligt: **P. Heusch, E. Speckenmeyer**

Es sind Average Case Analysen zum Exact Satisfiability- und zum Set Partitioning Problem (zusammen mit R. Kemp, Universität Frankfurt, und J. W. Rosenthal, Ithaca/N.Y.) durchgeführt worden. Dabei sind Klassen von Input Instanzen bezüglich der zu ihrer Lösung benötigten mittleren Laufzeit von Backtracking Algorithmen klassifiziert worden; in Klassen, die in Polynomzeit gelöst werden können und solche, die Exponentialzeit benötigen.

Darüberhinaus sind für Klassen aussagenlogischer Hornformeln und für verschiedene Varianten der Resolutionsstrategie probabilistische Untersuchungen über die Tiefe von Ableitungsbäumen vorgenommen worden.

Fertiggestellte Arbeiten:

J. .W. Rosenthal, E. Speckenmeyer, R. Kemp: *Exact-Satisfiability, a Natural Extension of Set Partitioning,* erscheint in: Discrete Applied Math.

E. Speckenmeyer, R. Kemp: *On the Average Time Behavior of Set Partitioning,* Proceedings Computer Science Logic 1989 (CSL '89), Springer LNCS, Vol. 440 (1990), 369 – 381

P. Heusch, E. Speckenmeyer: *Some Aspects of the Probabilistic Behavior of Variants of Resolution,* zur Veröffentlichung eingereicht, Dezember 1990

Projekt:

PARKI

Beteiligt: **M. Böhm, E. Speckenmeyer, H. Stamm (Univ. Bonn)**

(in Zusammenarbeit mit H. Kleine Büning, Duisburg, und A. B. Cremers, Bonn)

Förderung: **Land NRW**

PARKI ist ein Teilprojekt im Forschungsverbund KI des Landes NRW von Prof. A. B. Cremers.
Projektleiter: E. Speckenmeyer
Projektmitarbeiter: H. Stamm
Projektbeginn: August 1990
Hierbei geht es um die Parallelisierung von algorithmischen Paradigmen, die im Bereich KI von Relevanz sind. Zielmaschine ist dabei ein Transputersystem. Im Rahmen einer Projektgruppe an der Universität Dortmund (Projektleiter: E. Speckenmeyer, SS 1989 + WS 1989/90) sind bereits Parallelisierungen zur Projektthematik durchgeführt worden.

Im Berichtszeitraum ist eine Arbeit zur mathematischen Modellierung "Superlinearer Speedup-Effekte im Mittel" bei der Lösung NP-harter Suchprobleme mit Hilfe der Methode Backtracking in einer verteilten Umgebung entstanden (von: E. Speckenmeyer).

In dem Zusammenhang ist eine neue Art, die $\alpha - \beta$ Suche von Spielbäumen in einer verteilten Umgebung zu beschleunigen entstanden(von: M. Böhm und E. Speckenmeyer).

Fertiggestellte Arbeiten:

E. Speckenmeyer: *Is Average Superlinearer Speedup Possible?,* zur Veröffentlichung eingereicht, März 1990

M. Böhm, E. Speckenmeyer: *A Dynamic Processor Tree for Solving Game Trees in Parallel,* in: Methods of Operation Research 63, 1990, 479 – 489

E. Speckenmeyer, H. P. Siemon (Hrsg): *Abschlußbericht der Projektgruppe "PAPT" (Parallelisierung algorithmischer Paradigmen auf Transputernetzwerken),* Interner Bericht des FB Informatik, Universität Dortmund, Dortmund, 1990

Projekt:

PRAM-OCCAM

Beteiligt: **T. Seifert, E. Speckenmeyer**

In diesem Projekt ist eine parallele Programmiersprache, PRAM genannt, für das ***single instruction multiple data shared memory*** (SIMD SM) Multiprozessor Maschinenmodell entwickelt worden. PRAM enthält Rekursion, dynamische Datenstrukturen und die Konzepte globaler (***common memory***) und lokaler (***private memory***) Variablen, wobei das Konzept globaler Variablen dahingehend erweitert ist, daß die ***common memory*** Eigenschaft bezüglich einer Prozessorgruppe gilt.

Es ist ein Übersetzer entwickelt worden, der in PRAM geschriebene Programme nach OCCAM2 übersetzt, wodurch solche Programme unmittelbar auf einem Transputernetzwerk ausführbar sind. Dadurch ergeben sich für den Entwickler paralleler Programme auf einem Transputernetzwerk mehrere Vorteile:

- Das shared memory Konzept von PRAM befreit den Programmierer von der expliziten Organisation des Nachrichtenaustauschs zwischen den einzelnen Prozessoren.
- Die shared memory Sicht von PRAM bietet komfortablere Ausdruckmöglichkeiten beim Entwurf paralleler Algorithmen verglichen mit OCCAM.
- In der Literatur sind mittlerweile eine Vielzahl effizienter Algorithmen basierend auf dem SIMD SM Modell für eine Reihe algorithmischer Probleme entwickelt worden. Diese Algorithmen können sehr einfach in PRAM formuliert werden und sind damit unmittelbar auf Transputernetzwerken ausführbar.

Fertiggestellte Arbeiten:

T. Seifert, E. Speckenmeyer: *Spezifikation einer Sprache zur Simulation von PRAM-Modellen und ihre Übersetzung nach OCCAM,* erscheint in Proceedings Transputer-Anwender-Treffen 1990 (TAT '90, Hauptvortrag)

Vorträge auf Konferenzen und Workshops

Symposium on Theoretical Aspects of Computer Science (STACS '90), Rouen, Frankreich, 22. – 24. 2. 90

K. Indermark: "*Computational Aspects of Structural Recursion*"
W. Thomas : "*Infinite Trees and Automaton Definable Relations over ω-Words*"

Workshop "Term Rewriting: Theory and Application ", Universität Dortmund, 19. – 21. 3. 90

H. Vogler, F. Herrmann: "*Funktionale Programmierung mit primitiver Rekursion — Formale Modelle zur Reduktionssematik*"

Workshop on Parallel and Distributed Processing (WP & DP '90), Sofia, Bulgarien, 27. – 29. 3. 90

H. Kuchen, A. Wagener: "*Comparison of Dynamic Load Balancing Strategies*"

Third European Symposium on Programming (ESOP '90), Copenhagen, Dänemark, 15. – 18. 5. 90

H. Kuchen, R. Loogen, J. J. Moreno Navarro, M. Rodríguez Artalejo: "*Graph-based Implementation of a Functional Logic Language*"

ARIDAM V, Rutgers University, New Brunswick/N.Y., 30. 5 – 8. 6. 90

E. Speckenmeyer: "*On the Average Time Complexity of Set Pertitioning*"

Second International Workshop on Implementations of Functional Languages on Parallel Architectures, Nijmegen, Niederlande, 6. – 8. 6. 90

H. Kuchen, G. Geiler: "*Distributed Applicative Arrays*"
R. Loogen: "*From Reduction Machines to Narrowing Machines*"

Conference on Mathematical Foundations of Computer Science (MFCS '90), Banská Bystrica, Tschechoslowakei, 27. – 31. 8. 90

F. Hinz: "*The Membership Problem for Context-Free Chain Code Picture Languages*"

Dagstuhl-Seminar "Functional Programming: Optimization for Parallelism", Internationales Begegnungs- und Forschungszentrum für Informatik, Schloß Dagstuhl, Saarland, 3. – 8. 9. 90

H. Kuchen, G. Geiler: "*Distributed Applicative Arrays*"
R. Loogen: "*Extending a Graph Reduction Machine for the Implementation of a Functional Logic Language*"

Transputer-Anwender-Treffen (TAT '90), Klinikum der RWTH Aachen, 17. – 18. 9. 90

H. Kuchen, A. Wagener: "*Lastverteilung auf Multicomputern.*"
T. Seifert, E. Speckenmeyer: "*Spezifikation einer Sprache zur Simulation von PRAM-Modellen und ihre Übersetzung nach OCCAM*"

Workshop on Integration of Functional and Logic Programming, Granada, Spanien, 24. – 28. 9. 90

H. Faßbender, H. Vogler: "*A Narrowing Machine for the Functional Logic Language BABEL, Restricted to Primitive Recursion*"
G. Hogen: "*Extension of the Parallel Abstract Machine (PAM) with Evaluators*"
H. Kuchen, W. Hans: "*An AND-parallel Abstract Machine for BABEL*"

R. Loogen: "*From Reduction Machines to Narrowing Machiness*"
M. Sassin: "*Design of an Abstract Shared Memory Machine for And-parallel BABEL with Dependency Information*"
B. Steffen: "*Data Flow Analysis as Model Checking*"
St. Winkler: "*Implementation of a BABEL System in Miranda*"

Second International Conference on Algebraic and Logic Programming, Nancy, Frankreich, 1. – 3. 10. 90

J. J. Moreno Navarro, H. Kuchen, R. Loogen, M. Rodríguez Artalejo: "*Lazy Narrowing in a Graph-Machine*"

Fourth Workshop on Computer Science Logic (CLS '90), Heidelberg, Deutschland, 1. – 5. 10. 90

E. Speckenmeyer: "*Some Aspekts of the Probabilistic Behavior of Variants of Resolution*"

Weitere Vorträge:

E. Speckenmeyer: "*Können k Prozessoren im Mittel mehr als k mal schneller arbeiten als einer?*", Informatik-Kolloquium, RWTH Aachen, 2. 5. 90

K. Indermark: "*Effiziente Implementierung simultaner Rekursion*", Universität Bonn, 14. 05. 90, eingeladener Vortrag, Festkolloquium zum 60. Geburtstag von K.H. Böhling

H. Kuchen: "*Dynamische Lastverteilungsstrategien*", EWH Koblenz, 23. 05. 90

H. Vogler: "*Effiziente Implementierung von funktionalen Programmen mit primitiver Rekursion*", Universität Ulm, 29. 05. 90

E. Speckenmeyer: "*Können k Prozessoren im Mittel mehr als k mal schneller arbeiten als einer?*", Informatik-Kolloquium, EWH Koblenz, 13. 6. 90

H. Kuchen: "*Dynamische Lastverteilungsstrategien*", Universität Hildesheim, 19. 06. 90

E. Speckenmeyer: "*Über das erwartete Laufzeitverhalten von Backtracking zur Lösung des Set Partitioning Problems*", Informatik-Kolloquium, Universität Frankfurt/M., 25. 6. 90

H. Kuchen: "*Parallele Implementierung Funktionaler Sprachen*", Siemens AG, München, 27. 06. 90

A. Maaßen: "*Parallel Programming with Data Structures and Higher Order Functions*", University of Kent at Canterbury, (Forschungsaufenthalt 16. 9. - 20. 9. 90)

K. Indermark: "*Berechnungsstrategien für strukturell-rekursive Programme*", Informatik-Kolloquium, Universität Kiel, 16. 11. 90

Veröffentlichungen 1990

M. Böhm, E. Speckenmeyer: *A Dynamic Processor Tree for Solving Trees in Parallel*, Methods of Operation Research 63 (1990), 479 – 489

R. Cleaveland, B. Steffen: *When is "Partial" Adequate? A Logic-Based Proof Technique Using Partial Specifications*, Proceedings of the 5th Annual Symposium on Logic in Computer Science '90, IEEE, 440 – 450

R. Cleaveland, B. Steffen: *A Preorder for Partial Process Specifications*, Proceedings CONCUR '90, Springer LNCS, Vol. 458 (1990), 141 – 152

J. Dassow, F. Hinz: *Kettenkode-Bildsprachen. Theorie und Anwendungen*, Wissenschaftliche Zeitschrift der TU Magdeburg 33 (1990), 1-12

H. Faßbender, H. Vogler: *A Call-by-need Implementation of Syntax Directed Functional Programming,* Aachener Informatik-Bericht Nr. 90-22, RWTH Aachen, 1990

Z. Fülöp, F. Herrmann, S. Vàgvölgy, H. Vogler: *Tree Transduces with External Functions,* Aachener Informatik-Bericht Nr. 90-27, RWTH Aachen, 1990

R. van Glabbeck, S. Smolka, B. Steffen, Ch. Torts: *Reaktive, Generative and Stratified Models of Probabilistic Processes,* Proceedings of the 5th Annual Symposium on Logic in Computer Science '90, IEEE, 130 – 142

S. Graf, B. Steffen: *Compositional Minimization of Finite State Process,* International Workshop on Computer-Aided Verification '90, June 1990, DIMACS Technical Report 90-31 Volume II, New Jersey, erscheint auch in Springer LNCS

F. Herrmann, H. Vogler: *Primitive Recursive Programs for Defining Tree Functions,* Aachener Informatik-Bericht Nr. 90-26, RWTH Aachen, 1990

P. Heusch, E. Speckenmeyer: *Some Aspects of the Probabilistic Behavior of Variants of Resolution,* zur Veröffentlichung eingereicht, Dezember 1990

F. Hinz: *The Membership Problem for Context-Free Chain Code Picture Languages,* Proceedings MFCS '90, Springer LNCS Vol. 452 (1990), 329-336

V. Jansen, A. Potthoff, W. Thomas, U. Wermuth: *A Short Guide to the AMORE System (computing Automata, Monoids, and Regular Expressions),* Aachener Informatik-Bericht Nr. 90-2, RWTH Aachen, 1990

H. Kuchen, A. Wagener: *Comparison of Dynamic Load Balancing,* Aachener Informatik-Bericht Nr. 90-5, RWTH Aachen, 1990
erscheint auch in: Workshop on Parallel and Distributed Processing, Sofia, 1990, Elsevier Publishers

H. Kuchen, R. Loogen, J. J. Moreno Navarro, M. Rodríguez Artalejo: *Graph-based Implementation of a Functional Logic Language,* Proceedings European Symposium on Programming (ESOP), Copenhagen, Springer LNCS, Vol. 432 (1990), 271 – 290
ausführliche Fassung: Aachener Informatik-Bericht Nr. 89-20, RWTH Aachen, 1989

H. Kuchen, G. Geiler: *Distributed Applicative Arrays,* Proceedings 2nd International Workshop on Implementation of Functional Languages on Parallel Architectures, Nijmegen, Technical Report (University of Nijmegen) no. 90-16 (1990), 223 – 235

H. Kuchen, R. Loogen: *Parallele Implementierung einer funktionalen Programmiersprache auf einem Transputer-Mehrprozessor-System,* Proceedings Transputer-Anwender-Treffen 1989 (TAT '89), Aachen, Springer IFB 237, 21 – 33

H. Kuchen, A. Wagener: *Lastverteilung auf Multicomputern,* Proceedings Transputer-Anwender-Treffen 1990 (TAT '90), Aachen, Abstraktband erschienen als Bericht des Instituts für Physiologie der RWTH Aachen

R. Loogen: *Stack-based Implementation of Narrowing,* Aachener Informatik-Bericht Nr. 90-4, RWTH Aachen, 1990

R. Loogen: *Parallele Implementierung funktionaler Programmiersprachen,* Informatik Fachberichte 232, Springer Verlag, 1990

A. Maaßen: *Compilerentwicklung in Miranda - ein Praktikum in funktionaler Programmierung,* Aachener Informatik-Bericht Nr. 90-14, RWTH Aachen, 1990

J. J. Moreno Navarro, H. Kuchen, R. Loogen, M. Rodríguez Artalejo: *Lazy Narrowing in a Graph Machine,* Proceedings 2nd International Conference on Algebraic and Logic Programming, Nancy, Frankreich, Springer LNCS, Vol. 463 (1990), 298 – 317, ebenso erschienen als Aachener Informatik-Bericht Nr. 90-11, RWTH Aachen, 1990

Y. Ortega Mallén, D. de Frutos Escrig: *A Complete Proof System for Timed Observations,* Aachener Informatik-Bericht Nr. 90-20, RWTH Aachen, 1990

T. Seifert, E. Speckenmeyer: *Spezifikation einer Sprache zur Simulation von PRAM-Modellen und ihre Übersetzung nach OCCAM,* erscheint in: Proceedings Transputer-Anwender-Treffen 1990 (TAT '90)

E. Speckenmeyer: *Is Average Superlinear Speedup Possible?,* zur Veröffentlichung eingereicht, März 1990

E. Speckenmeyer, R. Kemp: *On the Average Time Behavior of Set Partitioning,* Proceedings Computer Science Logic 1989 (CLS '89), Springer LNCS, Vol. 440 (1990), 369 – 381

E. Speckenmeyer, H. P. Siemon (Hrsg.): *Abschlußbericht der Projektgruppe "PAPT (Parallelisierung algorithmischer Paradigmen auf Transputernetzwerken),* Interner Bericht des FB Informatik, Universität Dortmund, Dortmund, 1990

S. Smolka, B. Steffen: *Priority as Extremal Probability,* Proceedings CONCUR '90, Springer LNCS, Vol. 458 (1990), 456 – 466

B. Steffen, J. Knoop, O. Rüthing: *A Program Representation for Optimal Program Transformations,* Proceedings European Symposium on Programming (ESOP '90), Copenhagen, Denmark, Springer LNCS, Vol. 432 (1990), 389 - 405

J. W. Rosenthal, E. Speckenmeye: *Exact-Satisfiable, a Natural Extension of Set Partitioning,* erscheint in: Discrete Applied Math.

Habilitationsschriften und Dissertationen

H. Vogler: *Funktionale Programmierung mit primitiver Rekursion – Formale Modelle zur Reduktionssemantik* (Habilitationsschrift, RWTH Aachen, 24. 10. 90)

F. Hinz: *Erzeugung von Bildsprachen durch Chomsky-Grammatiken – Entscheidbarkeits- und Komplexitätsfragen* (Dissertation, RWTH Aachen, 25. 06. 90)

Diplomarbeiten

Denker, Harald: *Zwischencodeoptimierung bei der parallelen Implementation einer funktionalen Sprache* (Indermark)

Delonge, Franz René: *Eine Beschreibungssprache für Dialogoberflächen* (Indermark)

Dorn, Adrian: *Mehrband- und Mehrkopfautomaten über Wörtern* (Thomas)

Faßbender, Heinz: *Implementierung der call-by-need Auswertungsstrategie für Macro Tree-to-String Transducer auf Nested-Stack Maschinen mit geschachtelten Ausgabebändern* (Indermark)

Geiler, Günther: *Implementierung verteilter applikativer Arrays* (Indermark)

Glimm, Lukas: *Implementierung von primitiv rekursiven Schemata auf attributierten Pushdown-Maschinen* (Indermark)

Händeler, Kai: *Effiziente Implementierung der parallelen abstrakten Maschine PAM in Transputerassembler* (Indermark)

Hans, Werner: *Entwurf einer abstrakten Maschine für eine UND-parallele Implementierung der funktionalen Logik-Programmiersprache BABEL auf einem lose gekoppelten Multi-Computer-System* (Indermark)

Kühnemann, Armin: *Transformation strukturell-rekursiver Programme in Normalform* (Indermark)

Ramamonjishao, Franck: *Typinferenz bei funktionalen Sprachen* (Indermark)

Robens, Hubert: *Abstrakte Graphreduktionsmaschinen zur Implementierung funktionaler Programmiersprachen mit Striktheitsanalyse* (Indermark)

Schleicher, Axel: *OR-parallele Implementierung der funktional-logischen Sprache BABEL* (Indermark)

Seibert, Sebastian: *Quantorenhierarchien über sprachtheoretischen Relationsklassen* (Thomas)

Spieß, Willi: *Eingeschränkte temporale Logiken und ihre automatentheoretische Charakterisierung* (Thomas)

Theus, Anette: *Entwurf eines EDIF-Übersetzers für elektrische Netzwerkbeschreibungen* (Engl/Indermark)

Wagener, Andreas: *Lastverteilungsstrategien und Netzwerktopologien für eine Realisierung der parallelen abstrakten Maschine PAM* (Indermark)

Wilke, Thomas: *Eine Eilenberg-Korrespondenz für ω- Sprachen* (Thomas)

Wischnewski, Romuald: *Implementierung einer Übersetzung von MIRANDA in einen getypten λ-Kalkül als Frontend eines parallelisierenden Compilers* (Indermark)

Lehrstuhl für Informatik III

1 Personal des Lehrstuhls

- Leiter:

  Prof. Dr.-Ing. M. Nagl

- Sekretariat:

  A. Fleck
  M. Hirsch

- Wiss. Mitarbeiter:

  Dipl.-Inform. V. Bacvanski, Gastwissenschaftler (ab 1.10.90)
  Dipl.-Inform. M. von der Beeck (BMFT)
  Dipl.-Inform. J. Börstler
  Dipl.-Inform. M. Eichstädt (BMFT, halbtags)
  Dipl.-Inform. P. Heimann (ab 17.4.90)
  Dipl.-Math. Th. Janning
  Dipl.-Inform. N. Kiesel (BMFT)
  E. Kuikka, M. Sc., Gastwissenschaftlerin (bis 1.8.90)
  Dipl.-Inform. M. Lefering (ab 15.10.90)
  Dipl.-Math. U. Schleef
  Dipl.-Inform. A. Schürr (BMFT)
  Dipl.-Inform. J. Theis (MWF, ab 1.12.90)
  Dipl.-Inform. B. Westfechtel
  Dipl.-Inform. Ch. Wojahn (BMFT, ab 17.12.90)
  Dipl.-Inform. A. Zündorf (DFG, ab 2.5.90)

- Systemprogrammierung:

  R. Breuer (Math.-techn. Ass., halbtags)
  U. Cordts (Math.-techn. Ass., BMFT, halbtags, ab 28.8.90)
  M. Lischewski (Math.-techn. Ass., halbtags)
  C. Weigmann (Math.-techn. Ass., halbtags, ab 29.8.90)

- Drittmittelgeber:

  Deutsche Forschungsgemeinschaft
  Bundesministerium für Forschung und Technologie (ESF)
  Europäische Gemeinschaft (Esprit, Complement)
  Minister für Wissenschaft und Forschung des Landes NRW

Das Lehr- und Forschungsgebiet Informatik des Lehrstuhls ist durch einen eigenen Abschnitt vertreten.

2 Übersicht

Dieser Abschnitt gibt eine kurze *Übersicht* über die Forschungsaktivitäten und die sonstigen Aktivitäten des Lehrstuhls und verweist dabei auf die detaillierteren Ausführungen in den folgenden Abschnitten.

Die am Lehrstuhl betriebenen Forschungsvorhaben fallen im wesentlichen alle unter das *Generalthema* „Konzepte, Sprachen, Methoden und Werkzeuge der Softwaretechnik“. Noch wichtiger als für die Softwareentwicklung ist dabei eine Unterstützung der Softwarewartung. Eine besondere wichtige Randbedingung bei der Erstellung ist die Forderung nach Wiederverwendbarkeit. Zielsetzung ist es, automatisierte und nicht-automatisierte Hilfsmittel für die Softwareerstellung und -wartung zur Verfügung zu stellen, zwei Bereiche, die in der Praxis unter gravierenden Problemen leiden (hohe Kosten und mangelnde Qualität).

Bei den verschiedenen am Lehrstuhl angegangenen Problemstellungen ergeben sich folgende *Betrachtungsebenen*.

- Für einzelne Arbeitsbereiche oder allgemeine, phasenunabhängige Problemstellungen der Softwaretechnik werden neue *Konzepte, Sprachen* oder *Methoden* untersucht. Auf dieser Ebene ist im Jahre 1990 über Aktivitäten in den Bereichen Anforderungstechnik (Requirements Engineering), Programmieren-im-Großen, Projektmanagement, Versions- und Konfigurationskontrolle, verteilte Softwareentwicklung sowie Wiederverwendbarkeit von Software zu berichten.

- Für diese Bereiche sowie für weitere wurden bzw. werden *neue Werkzeuge* untersucht und entwickelt, wobei besonderer Wert auf einheitliche externe Eigenschaften und auf eine einheitliche Bedieneroberfläche der Werkzeuge gelegt wird. Diese Werkzeuge haben die folgenden Eigenschaften: Sie sind strukturbezogen, interaktiv, arbeiten integriert und rollen- bzw. gruppenbezogen.

- Von besonderem wissenschaftlichen Interesse und spezifisch für den Lehrstuhl ist der Lösungsansatz für diese Werkzeuge, die wir mit dem Begriff *Graphentechnik* bezeichnen. Alle Dokumente werden intern als Graphen betrachtet. Diese werden aus Bestandteilen komponiert, Veränderungen auf ihnen werden formal durch Graph-Grammatiken beschrieben. Aus dieser Spezifikation wird die Realisierung der Werkzeuge abgeleitet.

- Eine interessante Frage beim Bau von *integriert* zusammenwirkenden *Werkzeugen* ist, wie diese intern zu strukturieren sind. Zielsetzung ist dabei zum einen, daß die *Architektur* bestimmte Qualitätseigenschaften (wie Adaptabilität und Portabilität) besitzt, und daß Standards für solche Architekturen entwickelt werden, um allgemeingültige Komponenten aufzufinden bzw. um spezifische Komponenten mit geringem Aufwand (z.B. durch Generierungstechniken) zu erzeugen. Diese Strukturüberlegungen gelten nicht nur für Softwareentwicklungs-Umgebungen sondern allgemein für interaktive Systeme.

Über diese Forschungsaktivitäten hinaus, die im folgenden Abschnitt 3 im einzelnen beschrieben werden, hat der Lehrstuhl aktiv bei der Veranstaltung von Tagungen durch Programmkomitee-Mitgliedschaften und durch Organisation mitgewirkt. Diese sind in Abschnitt 4 dokumentiert. Hierunter fallen auch eine Reihe von Demonstrationen der Werkzeuge des IPSEN-Prototyps (vgl. 4.3), die am Lehrstuhl entwickelt wurden. Schließlich waren auch die Ämter des Lehrstuhlinhabers in der akademischen Selbstverwaltung, die Mitarbeit bei der Bündelung der Informatik in der Region Aachen und die Beteiligung in der Informatik-Grundausbildung im vergangenen Jahr eine besondere Belastung.

Im Rahmen des ESF (Eureka Software Factory)-Teilprojekts FERESA (First ESF for Real-Time and Embedded Systems Applications) werden Methoden und Werkzeuge für die Entwicklung von Realzeitsystemen entwickelt. Unsere Arbeiten zu Methoden und Werkzeugen des Requirements Engineering und zu dem Übergang zum Programmieren-im Großen werden mit der Zielsetzung durchgeführt, sie in das FERESA-Projekt einzubringen.

Des weiteren ist der Lehrstuhl an dem Esprit-Projekt COMPLEMENT (*COMP*rehensive *L*arge-scale *E*ngineering *ME*thodologies a*N*d *T*echnology transfer) beteiligt, dessen Zielsetzung insbesondere darin besteht, Methoden und Werkzeuge für den Realzeitbereich zu erforschen und zu entwickeln.

Veröffentlichungen:

M. Nagl: *Characterization of the IPSEN Project,* in N. Madhavji/W. Schäfer/H. Weber (Eds): Proc. 1st Int. Conf. on System Development Environments & Factories, 141–150, London: Pitman (1990).

M. Nagl: *Das Forschungsprojekt IPSEN,* Informatik-Forschung und Entwicklung, 5, 103–105 (1990).

M. Nagl: *Werkzeuge für die Softwareentwicklung,* Beitrag für das Sonderheft der RWTH-Themen zur Informatik (1990).

3 Forschungsvorhaben

3.1 Konzepte, Sprachen,Methoden

a) Anforderungstechnik

a1. Structured Analysis (SA)-Entity-Relationship (ER) - Integration

Beteiligt: S. Coors, Th. Janning

Die grundlegenden Ideen zur Integration von Structured Analysis und einer erweiterten Entity-Relationship-Sprache sind bereits in den vergangenen Jahren entwickelt worden. Generell beruht die Integration auf der Idee, daß das ER-Modell (also eine Hierarchie von ER-Diagrammen und ein Attributlexikon) die Typdefinitionen des Datenlexikons aus SA ersetzt und damit als Grundlage für Objektdeklarationen wie Datenspeicherdeklarationen und Datenflußdeklarationen dient. Allerdings mußte noch an verschiedenen Feinheiten geschliffen werden. Zum einen ergaben die Untersuchungen einer graphischen Benutzerschnittstelle Rückwirkungen auf die formale Beschreibung der Sprache. Zum anderen machte der Bau intelligenter Analysewerkzeuge einige Erweiterungen der Sprache notwendig. Im wesentlichen handelt es sich dabei um die Unterscheidung verschiedener Arten von Datenspeichern und Erweiterungen zur eindeutigen Balancierung von Datenflußdiagrammen.

Veröffentlichung:

S. Coors: *Entwurf einer integrierten Requirements-Engineering-Umgebung,* Diplomarbeit

a2. Analyse und Erweiterung von SADT (Structured Analysis and Design Technique)

Beteiligt: H. Docquier, Th. Janning

SADT ist eine der bekanntesten Analysemethoden. Trotzdem sind Probleme bei der Anwendung dieser Methode die Regel, was zu einem Verlust der Mächtigkeit dieser Methode führt. Eine Gegenüberstellung von SADT und Erweiterungen von SA um ein Kontrollmodell ergab, daß die Grundidee von SADT sehr mächtig ist, daß aber einige Modifikationen für einen sinnvollen Umgang damit notwendig sind. Eine solche Erweiterung von SADT umfaßt Kontextmodell, Informationsmodell, Aktivitätenmodell und Kontrollmodell.

Veröffentlichung:

H. Docquier: *ISADT - Eine überarbeitete strukturierte Analysemethode,* Diplomarbeit

a3. Entwicklung eines Kontrollmodells für das Requirements-Engineering

Beteiligt: M. von der Beeck

Im Berichtszeitraum wurde die Entwicklung eines Kontrollmodelles für Structured Analysis (SA) zur Steuerung der Prozeßaktivität fortgesetzt. Hierbei wurden Erweiterungen von endlichen Automaten untersucht, die Möglichkeiten zur Formulierung von Nebenläufigkeit, Hierarchie, Wiederverwendbarkeit, Zeitbedingungen sowie Erzeugung und Broadcasting interner Signale bieten. Als Beispiele dieser Erweiterungen seien Statecharts und Modecharts genannt.

Als problematisch hat sich die Kopplung der Kontrollmodellkomponenten mit den Datenprozessen von SA erwiesen. Diese Probleme resultieren aus der hierarchischen Struktur einer SA-Spezifikation. Eine Problemlösung kann sich dadurch ergeben, daß an Stelle der Information, die als Ergebnis der funktionalen Dekomposition in SA entsteht und zur Bildung der Datenflußdiagrammhierarchie führt, die Hierarchieinformation der Zustände herangezogen wird, in denen sich das zu spezifizierende System befinden kann. Das Resultat dieser Vorgehensweise ist ein einziger umfassender (hierarchischer) endlicher Automat, also z.B. ein Statechart, das lediglich mit einem flachen"Datenflußdiagramm gekoppelt ist. Dieser Ansatz muß noch weiter untersucht werden.

Nach Abschluß der beschriebenen Integration eines Kontrollmodelles soll eine zusätzliche Erweiterung um ein Informationsmodell nach dem Entity-Relationship-Modell zur Beschreibung von Daten und ihren Abhängigkeiten erfolgen.

b) Übergang Anforderungstechnik-Programmieren im Großen

Beteiligt: T. Bruckhaus, D. Däberitz, Th. Janning, M. Lefering

Es wurde ein Verfahren entwickelt, durch das eine Anforderungsdefinition in eine Vorgabe einer Modularchitektur übersetzt werden kann. In einem ersten Entwicklungsschritt werden dabei hierarchische Entity-Relationship-Diagramme in eine Hierarchie von abstrakten Datentypen übersetzt. Diese Modulhierarchie wird dann um funktionale Module und Ressourcen, die aus atomaren Prozessen resultieren, ergänzt. Schließlich wurde studiert, wie ebenfalls die Hierarchie von Datenflußdiagrammen und neue PiG-Konstrukte wie Teilsysteme und Vererbungsbeziehung berücksichtigt werden können.

Veröffentlichungen:

M. Lefering: *Ein Übergang von einer integrierten Requirements-Engineering-Methode zum Programmieren-im-Großen,* Diplomarbeit

Th. Janning/M. Lefering: *A Transformation from Requirements Engineering into Design: the Method and the Tool,* Proc. of Toulouse '90: 3rd Int. Workshop on Software Engineering & its Applications, 223-237, Nanterre, Frankreich : EC2 (1990)

Th. Janning/M. Lefering: *The Transformation of Requirements Definitions - A First Step for Integrating Two Important Lifecycle Phases,* Proc. of the ESF-Seminar 1990, Berlin: ESF

c) Programmieren im Großen

c1. Modulkonzepte

Beteiligt: J. Börstler, A. Joereßen, Th. Liese, E. Nadarzinski, M. Nagl, F. Philipp

Im Berichtszeitraum erfolgte eine Erweiterung der vorhandenen Sprache für das Programmieren im Großen um die Konzepte Teilsysteme, Vererbung und Generizität. Diese Erweiterungen erlauben eine von der Implementierungssprache weitgehend unabhängige Beschreibung von Software Architekturen. Die neuen Konzepte fördern dabei vor allem die Anpaßbarkeit und Erweiterbarkeit von Software Architekturen, was vor allem deren Wartbarkeit und Wiederverwendbarkeit erheblich verbessert.

Die Integration dieser Konzepte in eine Sprache machte deutlich, wie eng diese Konzepte zusammenhängen und sich gegenseitig beeinflussen. Dadurch konnten verschiedene Situationen herausgearbeitet werden, die Rückschlüsse auf die Qualität von Software Architekturen zulassen. Diese Erkenntnisse sollen in die Entwicklung von Editoren für das Programmieren im Großen eingebracht werden, um Operationen auf hohem semantischen Niveau anbieten zu können.

Veröffentlichungen:

M. Nagl: *Software Architectures: Importance, Notions, Experiences,* in (Hrsg.): Tagungsband Online 91 Konf., auch Aachener Informatik-Bericht Nr. 90-21, RWTH Aachen, 1990

M. Nagl: *Softwaretechnik: Methodisches Programmieren im Großen,* 387 S., Berlin: Springer-Verlag (1990)

c2. Wiederverwendbarkeit

Beteiligt: J. Börstler, D. Hassl

Kernbestandteil vieler Methoden zur Wiederverwendbarkeit von Softwaredokumenten ist eine Klassifikation von Softwaredokumenten. Da uns die aus der Literatur bekannten Klassifikationsschemata zu statisch erschienen, wird zur Zeit eine Sprache mit dazugehörigem Editor entwickelt, welche den systematischen Aufbau von Klassifikationsschemata erlaubt. Der Ansatz kombiniert facettierte und hierarchische Klassifikation mit eigenen Ideen. Dabei ist es möglich , Beziehungen zwischen verschiedenen (Teil-) Schemata herzustellen. Die Definition dieser Beziehungen wird ebenfalls von dem Editor unterstützt. Dieses Vorgehen erlaubt dem/der Benutzer/in Beziehungen zu definieren, welche den speziellen Anforderungen bestimmter Anwendungsbereiche gerecht werden.

d) Revisions- und Konfigurationskontrolle

Beteiligt: B. Westfechtel

Zu diesem Thema konnte eine Dissertation fast fertiggestellt werden. Sie beschäftigt sich mit der Revisionskontrolle, d.h. der Kontrolle der zeitlichen Entwicklung von Softwaredokumenten (Anforderungsdefinitionen, Architekturen, Modulimplementationen etc.) und ihren Querbeziehungen. Im Verlauf ihrer Lebensgeschichte durchlaufen Dokumente bezüglich ihres Inhalts nacheinander verschiedene Zustände, weil Fehler beseitigt werden, Optimierungen vorgenommen werden etc. Solche Zustände werden als Revisionen bezeichnet. Die Revisionskontrolle ist insbesondere bei der Wartung komplexer Softwaresysteme von herausragender Bedeutung.

In dieser Arbeit wird ein formaler Ansatz zur Revisionskontrolle entwickelt, der sich durch folgende Eigenschaften auszeichnet:

- Einheitliches formales Datenmodell. Die Revisionskontrolle beruht auf einem einheitlichen formalen Datenmodell, nämlich gerichteten, markierten und attributierten Graphen. Zur formalen Spezifikation von Werkzeugen zur Revisionskontrolle werden programmierte Graphersetzungssysteme verwendet.

- Strukturbezogenheit. Die Revisionskontrollwerkzeuge selbst arbeiten strukturbezogen und sind mit anderen, ebenfalls strukturbezogenen Werkzeugen integriert.

- Dokumenttypunabhängigkeit. Die Revisionskontrollwerkzeuge werden so weit wie möglich von den Typen der Dokumente unabhängig gehalten, deren Verwaltung unterstützt wird. Dadurch wird erreicht, daß die Werkzeuge sich allgemein einsetzen lassen und nicht jedesmal erweitert werden müssen, wenn ein neuer Dokumenttyp hinzukommt.

- Arbeitsbereichsübergreifende Integration von Aktivitäten. Die Integration von Aktivitäten, die verschiedenen, voneinander abhängigen Arbeitsbereichen (Programmieren im Großen, Programmieren im Kleinen etc.) zuzuordnen sind, beschränkt sich nicht auf Konsistenzanalysen, sondern es werden Vorgaben erzeugt und abhängige Revisionen angepaßt, nachdem bestimmende Revisionen geändert worden sind.

- Hohes Maß an Kontrolle der Konsistenz des Datenbestands. Während die interne Konsistenz von Revisionen (kontextfreie und kontextsensitive Korrektheit) mit Hilfe von Werkzeugen überwacht wird, die nicht Gegenstand dieser Arbeit sind, besteht ein herausragendes Merkmal des hier beschriebenen Ansatzes darin, daß die externe Konsistenz zwischen Revisionen voneinander abhängiger Dokumente kontrolliert wird (z.B. Konsistenz einer Modulimplementation mit der Spezifikation der Modulschnittstelle in der Architektur eines Softwaresystems).

Veröffentlichung:

B. Westfechtel: *Revisionskontrolle in einer integrierten Softwareentwicklungs-Umgebung*, Dissertation, Fertigstellung Anfang '91

e) Arbeitsteilige Softwareentwicklung

Beteiligt: U. Schleef

Moderne Softwaresysteme sind derart komplex geworden, daß sie nicht mehr von einer einzelnen Person, sondern nur noch arbeitsteilig in Mehr-Personen-Projekten entwickelt werden können. Dabei gibt es in diesen Projekten abhängig von der Größe des Entwicklungsvorhabens und damit abhängig von der Größe des Projektteams sehr unterschiedliche Organisationsformen und Vorgehensweisen, die sich natürlich auch auf die Anforderungen an die einzusetzenden Softwareentwicklungswerkzeuge auswirken. Ziel dieses Vorhabens ist es, die bisherigen Ergebnisse des IPSEN-Projekts unter dem Gesichtspunkt der arbeitsteiligen Softwareentwicklung zu betrachten und die IPSEN-Ideenwelt von eng miteinander vernetzten Dokumenten und Werkzeugen mit der Welt der Mehr-Personen-Projekte zusammenzuführen.

Im Berichtszeitraum lag der Schwerpunkt der Arbeit im Bereich der mittelgroßen Mehr-Personen-Projekte. Dazu wurden zunächst verschiedene Projektorganisationsformen und Vorgehensweisen bei der Aufgabenverteilung studiert, um die Anforderungen an die Werkzeugunterstützung für diese Klasse von Mehr-Personen-Projekten festlegen zu können. Dabei hat sich noch viel deutlicher als bei den bisher betrachteten kleinen Mehr-Personen-Projekten gezeigt, wie eng das Problem der arbeitsteiligen Softwareentwicklung mit dem Problem der Softwarekonfigurationsverwaltung verknüpft ist. Um diese Verknüpfung noch näher studieren zu können, wurde die Entwicklung von bisher noch nicht vorhandenen Werkzeugen für die Softwarekonfigurationsverwaltung angegangen. Gleichzeitig wurden verschiedene Modelle des Softwaredokument-Erstellungs- und Konfigurationsprozesses entworfen und anhand verschiedener Scenarien durchgespielt. Dabei beschreiben diese Prozeßmodelle, welche Stadien die Revisionen bzw. Konfigurationen der zu erstellenden Dokumente durchlaufen und welche Personen und Werkzeuge daran wie beteiligt sind.

Die Prozeßmodelle unterscheiden sich von Projektklasse zu Projektklasse. Für kleine Mehr-Personen-Projekte reichen oft zweistufige Modelle aus, bei denen die beteiligten Objekte entweder in Bearbeitung, aber nicht nutzbar oder zur Nutzung freigegeben, aber nicht mehr veränderbar sind. Für größere Mehr-Personen-Projekte sind komplexere Modelle erforderlich, bei denen die Objekte durch die Hände mehrerer Bearbeiter gehen, bevor sie zur Nutzung freigegeben werden. Aber auch innerhalb einer Projektklasse findet man in der Regel Unterschiede, da die Prozeßmodelle von den Gepflogenheiten und individuellen Vorstellungen eines Projektteams abhängig sind. Es wurde deshalb damit begonnen, darüber nachzudenken, wie man diese Prozeßmodelle in einer Softwareentwicklungsumgebung so verkapseln kann, daß sie zumindest vom Hersteller der Umgebung leicht ausgetauscht werden können. Im Idealfall sollten

natürlich die Benutzer der Umgebung das Prozeßmodell leicht ihren Wünschen entsprechend anpassen können. Die Berücksichtigung dieser Anforderung würde jedoch weit über den Rahmen des beschriebenen Vorhabens hinausgehen.

Neben den oben aufgeführten Aktivitäten wurden außerdem verschiedene Mechanismen zur arbeitsteiligen Erstellung großer Dokumente erprobt und erste Konzepte zur Unterstützung von Projekten, die sich aus mehreren Teilprojekten zusammensetzen, entwickelt.

3.2 Werkzeuge für verschiedene Arbeitsbereiche

a) Werkzeuge für das Requirements Engineering (RE)

Beteiligt: S. Coors, Th. Janning

Die Editor- und Analysewerkzeuge für das Requirements Engineering wurden durch Entwicklung einer Basisschicht für kontextsensitive Analysen verbessert. Sie ermöglichen die Überprüfung typischer Import-Export-Beziehungen, die auch in anderen Arbeitsbereichen auftreten.

b) Integrationswerkzeuge zwischen Requirements Engineering und Programmieren im Großen

Beteiligt: St. Coors, Th. Janning, M. Lefering

Nachdem im letzten Jahr ein Transformatorwerkzeug gebaut worden war, das batchartig Anforderungsdefinitionen in Modularchitekturvorgaben übersetzt, wurden in diesem Jahr die Anforderungen an inkrementell arbeitende Transformations- und dokumentübergreifende Analysewerkzeuge definiert, die Voraussetzung für einen praktischen Einsatz der Transformation von Requirements Engineering und Programmieren im Großen sind.

c) Programmieren im Großen (PiG)

c1. HOOD-Editor

Beteiligt: Th. Janning, E. Nadarzinski

Innerhalb des ESF (Eureka Software Factory) - Teilprojekts FERESA (First ESF for Real-Time and Embedded Systems Applications) sollen die Ergebnisse aus dem Requirements Engineering und bzgl. der Transformation zum Programmieren im Großen Anwendung finden. Da innerhalb FERESA die PiG-Sprache HOOD verwendet wird, war es für uns notwendig, ebenfalls Editor- und Analysewerkzeuge für HOOD zu bauen, auf die das FERESA-Transformatorwerkzeug aufsetzen kann.

c2. Ein Editor für eine erweiterte Programmieren-im-Großen-Sprache

Beteiligt: J. Börstler, A. Joereßen, Th. Liese, E. Nadarzinski, F. Philipp

Zu den Überlegungen aus 3.1.c wurden (und werden) Erweiterungen am vorhandenen PiG- Editor vorgenommen. Diese Erweiterungen beinhalten den kontextfreien Teil der neu hinzugekommenen Konzepte, sowie eine Reihe von Analysefunktionen, die den komplexen Zusammenhängen der neuen Architekturkonzepte Rechnung tragen.

c3. Komplexe (semantische) Edieroperationen

Beteiligt: J. Börstler, A. Joereßen

Für den PiG-Editor (siehe c2.) sollen die unter 3.1.c gewonnenen Erkenntnisse über die Anwendung von Programmiersprachen-Konzepten nutzbar gemacht werden. Hierbei sollen die kontextsensitiven Beziehungen innerhalb von Softwaredokumenten und zwischen verschiedenen Softwaredokumenten ausgenutzt werden. Des weiteren soll der Editor den sinnvollen Umgang mit Programmiersprachen-Konzepten unterstützen. Dies führt zur Definition von Edieroperationen, die es den Anwender/inne/n erlauben sich mehr und mehr auf die Konzepte zu konzentrieren, anstatt auf die Frage, wie sie in einer gegebenen Sprache ausgedrückt werden können.

3.3 Graphentechnik

a) Operationales Spezifizieren mit programmierten Graphersetzungssystemen

Beteiligt: A. Schürr, A. Zündorf

Die Bereitstellung von Methoden und Werkzeugen zum "ingenieurmäßigen" Bau komplexer interaktiver Systeme ist eine der Hauptaufgaben des Forschungsbereiches Softwaretechnik. Viele dieser interaktiven Systeme – für Anwendungsbereiche wie den rechnergestützten Entwurf von Konstruktionsplänen oder der Softwareentwicklung selbst – besitzen ein gemeinsames Charakteristikum: Sie unterstützen die Beschreibung, Bearbeitung und Darstellung komplex strukturierter Sachverhalte. Deshalb wurde im Rahmen des Forschungsprojektes IPSEN eine neue Spezifikationsmethode für die formale Beschreibung komplexer Strukturen und der darauf auszuführenden Operationen entwickelt.

Sie beruht auf der Darstellung komplexer Strukturen als Graphen und der operationalen Spezifikation der entsprechenden Zugriffsoperationen mit programmierten Graphersetzungssystemen. Diese Spezifikationen wurden bislang allein auf dem Papier erstellt, von Hand auf ihre Korrektheit hin überprüft und schließlich wiederum von Hand in eine Implementierung umgesetzt. Eine Dissertation zu diesem Themenkomplex wurde nahezu fertiggestellt.

Dieses Dissertationsvorhaben verfolgte die Zielsetzung, den Einsatz programmierter Graphersetzungssysteme durch interaktive, integrierte und inkrementell arbeitende Werkzeuge zu unterstützen, in den folgenden Etappen:

- Der Entwurf einer statisch typisierten Sprache namens PROGRESS, die die Vorzüge PROgrammierter GRaph-Ersetzungs-SySteme bei der Beschreibung von Transformationprozessen mit den Vorzügen moderner Datenbankprogrammiersprachen und attributierter Grammatiken bei der Beschreibung von Datenstrukturen in sich vereint.
- Die Entwicklung eines neuen (programmierten) Graphersetzungskalküls, der nicht nur die operationale Beschreibung graphverändernder Operationen, sondern auch die deklarative Beschreibung der Graphstrukturen selbst mit den Hilfsmitteln der Prädikatenlogik unterstützt.
- Die vollständig formale Beschreibung der Sprache PROGRESS mit den Hilfsmitteln des neu entwickelten Graphersetzungskalküls.
- Die Entwicklung einer graphbasierten Maschinerie für die Generierung syntaxgesteuerter Softwareentwicklungswerkzeuge (vermittels der Kombination bekannter EBNF-orientierter und neuer PROGRESS-orientierter Techniken).
- Der Bau einer integrierten Umgebung, die das syntaxgesteuerte Edieren, Analysieren und Ausführen von PROGRESS-Spezifikationen unterstützen wird, mit Hilfe der neuen graphbasierten Maschinerie, auf die in b) kurz eingegangen wird.

Veröffentlichungen:

G. Engels/C. Lewerentz/M. Nagl/W. Schäfer/A. Schürr: *Experiences in Building Integrated Tools, Part I,* Tool Specification, zur Veröffentlichung eingereicht

G. Engels/C. Lewerentz/M. Nagl/W. Schäfer/A. Schürr: *Experiences in Building Integrated Tools, Part II,* Environment Realization, in Arbeit

A. Schürr: *Spezifizieren mit programmierten Graphersetzungssystemen,* Dissertation, Fertigstellung Anfang '91

A. Schürr: *PROGRESS: A VHL-Language Based on Graph Grammars,* in: Ehrig H., Kreowski H.-J., Rozenberg G. (Eds.): Proc. 4th Int. Workshop on Graph-Grammars and Their Application to Computer Science, erscheint in: LNCS, Springer Verlag (1991)

b) Integration von logischen Dokumenten verschiedener Arbeitsbereiche

Beteiligt: M. Lefering

Bei der Softwareerstellung entstehen zahlreiche logische Dokumente unterschiedlicher Klassen, die im IPSEN-Projekt einheitlich mit Hilfe von Graphen realisiert werden.

Es sollen diese Dokumente und die ihnen zugrundeliegenden Sprachen auf Gemeinsamkeiten hin untersucht werden, um Basismechanismen für die Modellierung sowie den Entwurf und die Realisierung dieser Dokumente abzuleiten. Darauf aufbauend soll ein dokumentklassenunabhängiger, inkrementeller Transformationsansatz (für logische Dokumente) gesucht und ein Mechanismus zur Generierung inkrementell arbeitender Transformatoren entwickelt werden, die diesen Ansatz realisieren.

c) Die PROGRESS-Entwicklungsumgebung

Beteiligt: R. Köther, Th. Liese, A. Schürr, A. Zündorf

Parallel zur Fortentwicklung und vor allem formalen Beschreibung der Sprache PROGRESS wurde auch die Realisierung ihrer Entwicklungsumgebung weiter vorangetrieben. Das betrifft vor allem den syntaxgesteuerten PROGRESS-Editor, der nun den gesamten, aktuellen Sprachumfang unterstützt und einen inkrementellen Analysator für alle Typverträglichkeitsregeln der Sprache (über 350 Regeln) enthält.

Insgesamt umfaßt die PROGRESS-Umgebung damit ungefähr 150.000 Zeilen Modula-2 und C-Code, wovon nur 15.000 Zeilen wirklich PROGRESS-spezifisch sind. Von diesen 15.000 Zeilen wurden wiederum etwa 10.000 Zeilen – der kontextfreie Teil der Editorimplementation – generiert. Hierfür wurden einige Werkzeuge eingesetzt, die im Rahmen der Diplomarbeit (Köther'90) entstanden sind.

Sowohl die PROGRESS-Basisschicht als auch die erwähnte Generierungsmaschinerie wurden inzwischen für die Generierung anderer Editoren aus den Bereichen „Anforderungsanalyse" und „Entwurf von Software-Architekturen" erfolgreich eingesetzt.

Veröffentlichungen:

M. Nagl/A. Schür: *A Graph Grammar Specification Environmen*, Aachener Informatik-Bericht Nr. 90-15, RWTH Aachen, 1990

M. Nagl/A. Schürr: *A Specification Environment for Graph Grammars*, in: Ehrig H., Kreowski H.-J., Rozenberg G. (Eds.): Proc. 4th Int. Workshop on Graph-Grammars and Their Application to Computer Science, erscheint in: LNCS, Springer Verlag (1991)

3.4 Softwaretechnische Überlegungen zu interaktiven Systemen

a) Graphische Editoren

Beteiligt: D. Hassl, S. Vaillant, A. Schürr, M. Eichstädt

Anfang des Jahres wurde der bereits 1989 begonnene Bau eines Prototyps eines syntaxgesteuerten SA-Editors mit graphischer Benutzeroberfläche zu Ende geführt. Im Rahmen dieser Implementierungsarbeiten wurden eine Reihe von Basisschichten für die erweiterten Anforderungen ergänzt.

Der Schwerpunkt der Arbeit lag anschließend in der Verallgemeinerung der Konzepte, um zu beliebigen Sprachen (z.B. Requirementssprache, Entwurfssprache) graphische Editoren erzeugen zu können.

Dazu wurden Datenstrukturen entwickelt (Repräsentationsdokumente), durch die Dokumente auf komplexe Art, d.h. durch eine gemischt textuelle und graphische Darstellung, repräsentiert werden können. Für jede Darstellungsart (z.Z. Texte und graphartige Diagramme) wurden Teilrepräsentationen definiert, die für die jeweilige Darstellungsart spezifische Konstrukte und Layoutalgorithmen zur Verfügung stellen. Ein funktionaler Baustein gewährleistet die Integration in eine gemeinsame Repräsentation.

Als zukünftige Aufgaben wird neben der Fertigstellung der Implementierung obiger Aufgaben die Gestaltung der Benutzerschnittstelle von syntaxgesteuerten, graphischen Editoren betrachtet werden.

b) Revision der IPSEN-Architektur

Beteiligt: M. Eichstädt, P. Heimann, A. Schürr

Eine kritische Betrachtung hat gezeigt, daß die für im Rahmen des IPSEN-Projektes realisierte Werkzeuge verwendete Architektur im Lichte neuer Anforderungen überarbeitet werden sollte.

Eigenheit der IPSEN-Architektur ist die Trennung zwischen der logischen und der Repräsentationsinformation gespeicherter Dokumente. Die Transformationen zwischen diesen Datenstrukturen prägen den Aufbau des Systems. Dabei zeigt sich jedoch, daß vergleichbare Situationen je nach ihrem Auftreten unterschiedlich modelliert sind. Die Transformation ist zum Teil ein eigener Baustein, zum Teil ist sie an die Realisierung einer Datenstruktur angebunden. Diese Uneinheitlichkeit erschwert das Verständnis und bedeutet zusätzlichen Aufwand bei Erweiterungen.

Bei der Implementierung von Werkzeugen mit grafischer Benutzerschnittstelle wird offenkundig, daß eine Erweiterung der Funktionalität der Ein-/Ausgabesteuerung und eine Ergänzung der internen Modellierung von Dokumenten sinnvoll ist.

Darüber hinaus ist zu prüfen, ob und wie die monolithische IPSEN-Umgebung in lose gekoppelte Teile aufgespalten werden kann. Die enge Integration der Werkzeuge soll dabei erhalten bleiben.

c) Basisbausteine für die Verwaltung von Revisionsgeflechten

Beteiligt: G. Metzen, B. Westfechtel

In dieser Arbeit wurde ein Basisbaustein für die Revisionsverwaltung in einer integrierten Software-Entwicklungsumgebung entwickelt. Dieser Basisbaustein dient dazu, Verzeichnisse zu verwalten, in denen sich die Entwicklungsgeschichte der Dokumente, aus denen sich ein Softwaresystem zusammensetzt, sowie die zwischen ihnen bestehenden Querbeziehungen darstellen lassen. Zu jedem Dokument werden Revisionen verwaltet, die jeweils den Entwicklungsstand des entsprechenden Dokuments zu einem bestimmten Zeitpunkt wiedergeben. Auch Querbeziehungen zwischen den Revisionen werden in die Verwaltung einbezogen. Anhand dieser Querbeziehungen läßt sich feststellen, welche Revisionen voneinander abhängiger Dokumente jeweils zusammenpassen und somit zu einer Konfiguration des Softwaresystems zusammengesetzt werden können.

Veröffentlichung:

Gerd Metzen: *Revisionsverwaltung in einer integrierten Software-Entwicklungsumgebung,* Diplomarbeit

d) Softwareentwicklungs-Umgebungen in Arbeitsplatzrechnernetzen

Beteiligt: H. Scharrel, U. Schleef

Softwareentwicklung findet heute in der Regel nicht mehr auf einem Großrechner statt, sondern es werden dafür Arbeitsplatzrechner eingesetzt. Die übliche Vorstellung ist, daß jeder Softwareentwickler über seinen eigenen Arbeitsplatzrechner verfügt und daß diese Arbeitsplatzrechner in Mehr-Personen-Projekten durch ein Kommunikationsmedium mit hohen Leistungsmerkmalen miteinander verbunden sind. Die Problematik der arbeitsteiligen Softwareentwicklung ist daher stets mit der Problematik der Realisierung von Softwareentwicklungsumgebungen auf Rechnernetzen verbunden. Für die im IPSEN-Projekt betrachteten kleinen und mittelgroßen Softwareprojekte sollten diese Probleme durch verteilte Betriebssysteme, die die Daten- und Lastverteilung übernehmen, abgefangen werden. Leider stehen diese Betriebssysteme jedoch noch nicht allgemein zur Verfügung.

Im Rahmen einer Diplomarbeit wurde daher untersucht, welche grundlegenden Mechanismen für den Einsatz einer Softwareentwicklungsumgebung in einem Arbeitsplatzrechnernetz erforderlich sind und wie sich diese Mechanismen mit den am Lehrstuhl vorhandenen Ressourcen realisieren lassen. Dabei ging es vor allen Dingen um Probleme der optimalen Anordnung der Daten eines Softwareprojekts im Rechnernetz, der Verwaltung dieser verteilten Datenhaltung und um Maßnahmen zur Erhaltung eines konsistenten Datenbestandes bei Hard- und Softwareausfällen. Von den erarbeiteten Mechanismen wurden einige unter dem Betriebssystem SunOS implementiert und praktisch erprobt.

Veröffentlichung:

H. Scharrel: *Grundlegende Mechanismen zur Unterstützung geographisch verteilter Software-Entwicklung,* Diplomarbeit

e) Verteilter Objektspeicher

Beteiligt: N. Kiesel

Im Forschungsbereich verteilter Objektspeicher wurden Untersuchungen über die integrierten und verteilten Software-Entwicklungsumgebungen zugrunde liegenden Objekte durchgeführt. Hierbei lag der Schwerpunkt in der Entwicklung geeigneter Mechanismen, um die Interaktion und die Reaktivität zusammengesetzter und abhängiger Objekte effizient zu unterstützen. Diese Mechanismen sollen für die Kopplung persistenter Graphen eingesetzt werden. Da im Forschungsprojekt IPSEN alle Dokumente als solche persistenten Graphen abgelegt werden, bietet dies die Möglichkeit, Dokumentbeziehungen dynamisch auf- und abzubauen und somit reaktive verteilte Systeme zur Laufzeit zu konfigurieren. Mögliche Einsatzgebiete dieser Mechanismen sind die Kopplung von Dokumenten aus unterschiedlichen Bereichen des Softwarelebenszyklus (z.B. Programmieren im Großen und Programmieren-im-Kleinen) oder die Kopplung verschiedener logischer Modelle aus einem Bereich des Softwarelebenszyklus (z.B. Datenflußdiagramme und Entity-Relationship-Diagramme in der Anforderungsdefinition).

Im nächsten Jahr soll eine erste Prototyp-Implementierung dieser Mechanismen im Rahmen von einer oder zwei Diplomarbeiten durchgeführt werden. Weiterhin soll untersucht werden, inwieweit die entwickelten Mechnismen in das SoftwareBus-Konzept aus ESF und die Interprozeß-Kommunikation von PCTE passen.

f) Basismechanismen für die Änderungsverwaltung in einem Objektspeicher

Beteiligt: M. Broekmans, B. Westfechtel

In dieser Diplomarbeit wurde ein Dokumentenverwaltungssystem erweitert, das auf Graphen basiert und in einer strukturbezogenen Software-Entwicklungsumgebung eingesetzt wird. Das Ziel dieser Erweiterung bestand darin, Basismechanismen für die Änderungsverwaltung zur Verfügung zu stellen. Diese Basismechanismen dienen dazu, Transaktionen, Undo/Redo (Zurücksetzen bzw. Wiederholen von Benutzerkommandos) und die effiziente Abspeicherung von Revisionen (Entwicklungszuständen von als Graphen dargestellten Dokumenten) in aufeinander abgestimmter Weise zu unterstützen. Für alle genannten Aufgaben sind Folgen von Änderungsoperationen auf Graphen von zentraler Bedeutung. Intern werden Transaktionen, Undo/Redo und Deltas - bei letzteren handelt es sich um die Unterschiede zwischen verschiedenen Revisionen eines Dokuments - auf einen gemeinsamen Mechanismus, nämlich die Protokollierung von Änderungsoperationen zurückgeführt.

Veröffentlichung:

Michael Broekmans: *Änderungsverwaltung in einem Nichtstandard-Datenbanksystem,* Diplomarbeit

3.5 Weitere Vorhaben

a) Architekturmodellierung von Expertensystemen

Beteiligt: V. Bacvanski

Ziel dieses Vorhabens ist die Untersuchung von Softwarearchitekturen für Expertensysteme, ausgehend von den Schwächen und Grenzen, die Expertensysteme in professionellem Einsatz gezeigt haben. Der Ausgangspunkt dieser Forschung ist die Verwendung moderner Software Engineering Techniken, insbesondere Programmieren-im-Großen-Modellierung und objektorientierten Entwurfs. Das entwickelte Konzept bildet die Basis für die Konstruktion von Expertensystemen aus lose gekoppelten Modulen, die Anwendung von Wiederverwendbarkeit und insbesondere die Unterstützung für multiparadigmatische Systeme, in welchen traditionelle und Expertensystemkomponenten kombiniert sind.

b) Syntaxgesteuerte Textverarbeitungssysteme

Beteiligt: E. Kuikka

Heutige Systeme zur Textverarbeitung haben i.a. den Nachteil, daß den Systemen selbst die Struktur der bearbeiteten Dokumente (weitgehend) unbekannt ist. In diesem Ansatz werden Textdokumente (deklarativ) durch Grammatiken beschrieben und dann mit Hilfe von syntax-gesteuerten / -gestützten Editoren bearbeitet. Diese Editoren besitzen nun Wissen über die aktuellen Dokumente und können somit den Benutzer gezielt unterstützen. Ein in Prolog geschriebener Prototyp benutzt 'Eingabe'-Grammatiken zur Unterstützung der Texterfassung und 'Ausgabe'-Grammatiken zur Erzeugung von Layout-Informationen für einen LaTeX-Formatierer. Ein technischer Bericht über diese Arbeit ist in Vorbereitung.

Veröffentlichung:

E. Kuikka: *A Proposal for a Syntax-Directed Text Processing System,* Technischer Bericht in Vorbereitung

c) Programmiersprachenkonzepte

Beteiligt: M. Nagl

Einige Veröffentlichungen, die Programmiersprachenkonzepte verschiedener Sprachen dargestellt haben, wurden aktualisiert.

Veröffentlichungen:

M. Nagl: *Schlagwörter Ada und Smalltalk,* in P. Mertens u.a. (Hrsg.): Lexikon der Wirtschaftsinformatik, 2. Aufl., 8-9, 386-387 (1990)

M. Nagl: *Ada: Eine Einführung in die Softwaretechnik-Programmiersprache Ada,* 3. Auflage (1991), 345S., Wiesbaden: Vieweg-Verlag

4 Sonstige Aktivitäten

4.1 Softwaretechnik-Projektpraktika

a) Generizität und Objektorientierung für Modula-2 im SS 1990

Beteiligt: J. Börstler, A. Zündorf und Studenten

Generizität und objektorientierte Konzepte sind wichtige Hilfsmittel für die Erstellung von wiederverwendbarer und leicht wartbarer Software. Die Sprache Modula-2, die am Lehrstuhl zur Softwareentwicklung eingesetzt wird, bietet solche Mechanismen jedoch nicht direkt an. Aber durch strikte Programmierdisziplin ist es eingeschränkt möglich, diese Konzepte in Modula-2 zu simulieren.

Aufgabe des Projektpraktikums war es, Software-Werkzeuge zu erstellen, die den Programmierer bei dieser Arbeit unterstützen. Es wurden Erweiterungen des Sprachumfangs von Modula-2 definiert und im Falle der Generizität Präprozessoren entwickelt und implementiert, die solchermaßen erweiterte Modula-2 Programme in Standard Modula-2 umsetzen und dabei umfangreiche Fehleranalysen durchführen. Ein ausführlicher, zusammenfassender Bericht über die verwendeten Mechanismen und Ergebnisse befindet sich in Vorbereitung.

b) Layoutalgorithmen für Diagramme im WS 1990/91

Beteiligt: P. Heimann, M. Eichstädt, C. Wojahn, J. Theis und Studenten

Das Ziel des Projektpraktikums in diesem Semester war die Entwicklung inkrementeller Layoutalgorithmen für graphartige Diagramme. Die Algorithmen sollten innerhalb des graphischen SA-Editors realisiert werden. Besonderes Augenmerk wurde dabei auf die Inkrementalität der Algorithmen gelegt, um den Benutzer bei der interaktiven Arbeit mit dem Editor möglichst gut zu unterstützen.

Entstanden ist zunächst eine ausführliche Anforderungsdefinition, in der insbesondere Verfahren zur halbmanuellen Knotenpositionierung sowie für die automatische Kantenführung beschrieben sind.

Beim Entwurf einer Teilarchitektur bzw. dem Einbau in die Gesamtarchitektur des Prototyps wurden die verschiedenen Algorithmen nach dem Werkzeugkastenprinzip integriert, d.h. verschiedene Algorithmen für die gleiche Aufgabe (z.B. Kantenausrichtung) sind austauschbar. Aufgrund des insgesamt zu großen Erweiterungsaufwandes in der Gesamtarchitektur mußten hier gegenüber der Anforderungsdefinition einige Abstriche gemacht werden. Die Implementierung beschränkte sich auf Realisierung und Einbau je eines Algorithmus' zur Knoten- und zur Kantenpositionierung sowie einer geeigneten Hilfsdatenstruktur.

Durch die Realisierung innerhalb der IPSEN-Umgebung bot das Projektpraktikum für die Studenten und die neuen Mitarbeiter des Lehrstuhls eine gute Gelegenheit, sich in die Ideenwelt des IPSEN-Projektes, insbesondere in die Architektur des Prototyps, einzuarbeiten.

4.2 WG'89

Der Tagungsband des WG '89 on Graphtheoretic Concepts in Computer Science, den der Lehrstuhl im letzten Jahr veranstaltet hat, wurde veröffentlicht.

Veröffentlichung:

M. Nagl (Ed.): *Proc. WG'89 15th Int. Workshop on „Graphtheoretic Concepts in Computer Science"*, Lecture Notes in Computer Science 411, 372 S., Berlin: Springer-Verlag (1990)

4.3 Vorführung des IPSEN-Prototyps

Neben vielen internen Demonstrationen am Lehrstuhl, meist anläßlich des Besuchs auswärtiger Gäste, fanden eine Reihe größerer Demonstrationen des IPSEN-Prototyps statt, die wegen ihres Aufwands hier dokumentiert werden sollen:

08. 02.:	IPSEN-Prototyp (Programmieren im Großen, Programmieren im Kleinen), Vorlesung Softwaretechnik von Prof. Nagl, RWTH Aachen
Präsentation:	M. von der Beeck
08. 02.:	Structured Analysis/Entity-Relationship Editor, Vorlesung Softwaretechnik von Prof. Nagl, RWTH Aachen
Präsentation:	St. Coors
02. 03.:	PROGRESS-Editor, Arbeitstreffen über Methodenintegration im Requirements Engineering, RWTH Aachen
Präsentation:	M. von der Beeck
05. 03.– 06. 03.:	PROGRESS-Editor, 4. Int. Workshop on Graph Grammars and their Applications to Computer Science, Bremen
Präsentation:	M. von der Beeck
04. 04.:	Structured Analysis/Entity-Relationship Editor, Research Focus (ESF-Projekt), Uni Dortmund
Präsentation:	M. von der Beeck
05. 12.– 07. 12.:	Third International Workshop on Software Engineering and its Applications, Toulouse, Frankreich
Präsentation:	M. Lefering

4.4 Erweiterung und Pflege der Technischen Ausstattung

Beteiligt: **P. Heimann, U. Schleef**

Die Zahl der ständigen Benutzer der Rechnerausstattung am Lehrstuhl III ist im Berichtszeitraum auf vierzig angestiegen. Dadurch mußte für die Einweisung und Betreuung erheblich mehr Zeit aufgewendet werden als in den Vorjahren.

Vor allem aber wurde eine Erweiterung der vorhandenen Hard- und Software dringend erforderlich. Die sich daraus ergebenden Aufgaben umfaßten die Einrichtung neuer Arbeitsplätze in den Laborräumen einschließlich aller erforderlichen Anschlüsse, die Beschaffung und Installation neuer Rechner und ihre Einbindung in das vorhandene Netz. Zusätzlich wurden eine Reihe neuer Softwarepakete beschafft und installiert. Die lehrstuhleigene Rechnerausstattung besteht damit aus insgesamt 7 Sun3-Workstations, 16 SPARCstations sowie jeweils einem Sun3-Server und einem SPARC-server (alle Rechner von der Firma Sun Microsystems).

Der durch die Erweiterungen gestiegene Aufwand für Routinearbeiten und für die Behebung von Hard- und Softwareausfällen, konnte durch den Einsatz eines zweiten Mitarbeiters ausgeglichen werden.

Eine zusätzliche Arbeitsbelastung von mehreren Wochen entstand durch die Abwicklung einer umfangreichen Beschaffungsmaßnahme für die Informatik allgemein, zu der insbesondere die Erstausstattung (Hardware, Software, Möbel, Telefone etc.) der Lehrstühle V und VI gehörte.

4.5 Tagungs-, Antrags-, Begutachtungstätigkeit und akad. Ämter

Beteiligt: **die Profs. Eversheim, Michaeli, Nagl, Spaniol, Weck sowie für den Lehrstuhl die Mitarbeiter N. Kiesel und B. Westfechtel**

DFG Forschergruppe SUKITS Software- und Kommunikationsstrukturen in technischen Systemen

Beteiligt: **Dipl.-Math. H. Keutgen und M. Nagl**

Mitorganisation des Workshops „Qualitätssteigerung bei der Softwareerstellung" auf der Ausstellung „Forschungsland Nordrhein-Westfalen"

Beteiligt: **M. Nagl und Mitarbeiter**

WG '90 16th Int. Workshop on „Graphtheoretic Concepts in Computer Science"

Ada '91 Konferenz, Athen, Rezension der Beiträge

3rd Int. Workshop on Software Configuration Control, Trondheim, Mitglied des Programmkomitees und Rezension der Beiträge

Vorsitz: **M. Nagl**

Berufungskommission C3 „Konstruktion von Softwaresystemen“

Bewertung des Instituts F2 der GMD

Beteiligt: **M. Nagl**

Wissenschaftlicher Beirat der GMD

Finanz-, Bau-, Kolloquiumsbeauftragter der Fachgruppe Informatik

Organisation der Sonderreihe des Informatik-Kolloquiums „Firmen der Region stellen sich vor" in Zusammenarbeit mit dem Regionalen Industrie Club und dem Forum Informatik

5 Vorträge im Jahr 1990

J. Börstler: *What is Object Oriented Design ?,* Design Meeting des ESF Teilprojektes FERESA, Toulouse, 5.4.90

Th. Janning: *Integration von Requirements Engineering und Programmieren im Großen,* Workshop „Qualitätssteigerung bei der Softwareerstellung" im Rahmen der Veranstaltung „Forschungsland Nordrhein-Westfalen", 3.4.90 in Bonn

Th. Janning: *A Transformation of Requirements Definitions - A First Step for Integrating Two Important Lifecycle Phase,* ESF-Seminar, 21.11.90 in Berlin

Th. Janning: *The Transformation from Requirements Engineering into Design - The Method and the Tool,* Toulouse '90: 3rd Int. Workshop on Software Engineering & its Applications, 5.12.90 in Toulouse

M. Nagl: *Integrierte Werkzeuge,* Vortrag auf dem gemeinsamen Workshop der GEI, Aachen und des Lehrstuhls Informatik III auf der Ausstellung Forschungsland NRW, 30.4.90

M. Nagl: *An Environment for Graph Grammar Specifications,* Vortrag 4th Int. Workshop on Graph Grammars and Their Applications to Computer Science, 6.3.90

M. Nagl: *Graph Grammars which are Suitable for Applications,* Position Statement, 4th Int. Workshop on Graph Grammars and Their Applications to Computer Science, 7.3.90

M. Nagl: *Seminare Methodisches Programmieren im Großen,* DATEV Nürnberg, und GEI Frankfurt, im Frühjahr und Herbst 1990

M. Nagl: *Ein Ansatz für „intelligente" i^3-Werkzeuge,* Philips-Forschungslabor Aachen, 27.9.90

M. Nagl: *Die Bedeutung der Architekturmodellierung,* Vortrag zur Vorstellung des Buches, Augustinus-Buchhandlung, 29.10.90 in Aachen

M. Nagl: *Integrierte Werkzeuge für die Softwareentwicklung,* Universität GHS Paderborn, 30.10.90

M. Nagl: *Eine integrierte (Software-)Entwicklungs-Umgebung,* Vortrag im Informatik-Kolloquium der Universität Bonn, 9.11.90

M. Nagl: *Eine integrierte (Software-)Entwicklungs-Umgebung,* Vortrag, Universität Kaiserslautern, 13.11.90

A. Schürr: *PROGRESS, A VHL Language Based on Graph Grammar,* Vortrag auf dem 4th Int. Workshop on Graph Grammars and Their Applications to Computer Science, 5.3.90

A. Schürr: *PROGRESS - A Graph Grammar Based Language,* Process Modelling, Workshop, TAT - ESF Headquarter Berlin, 12. 6

6 Gesammelte Liste der Veröffentlichungen

6.1 Diplomarbeiten

M. Broekmans: *Änderungsverwaltung in einem Nichtstandard-Datenbanksystem*

S. Coors: *Entwurf einer integrierten Requirements Engineering-Umgebung*

H. Docquier: *ISADT - Eine überarbeitete strukturierte Analysemethode*

M. Lefering: *Ein Übergang von einer integrierten Requirements-Engineering-Methode zum Programmieren im Großen*

G. Metzen: *Revisionsverwaltung in einer integrierten Software-Entwicklungsumgebung*

H. Scharrel: *Grundlegende Mechanismen zur Unterstützung geographisch verteilter Software-Entwicklung*

6.2 Dissertationen

B. Westfechtel: *Revisionskontrolle in einer integrierten Softwareentwicklungs-Umgebung,* Dissertation, Fertigstellung Anfang '91

A. Schürr: *Spezifizieren mit programmierten Graphersetzungssystemen,* Dissertation, Fertigstellung Anfang '91

6.3 Tagungsbände / Zeitschriften / Technische Berichte

G. Engels/C. Lewerentz/M. Nagl/W. Schäfer/A. Schürr: *Experiences in Building Integrated Tools, Part I: Tool Specification,* zur Veröffentlichung eingereicht

G. Engels/C. Lewerentz/M. Nagl/W. Schäfer/A. Schürr: *Experiences in Building Integrated Tools, Part II: Environment Realization,* in Arbeit

Th. Janning/M. Lefering: *A Transformation from Requirements Engineering into Design: the Method and the Tool,* Proc. of Toulouse '90: 3rd Int. Workshop on Software Engineering & its Applications, 223-237, Nanterre, Frankreich : EC2 (1990)

Th. Janning/M. Lefering: *The Transformation of Requirements Definitions - A First Step for Integrating Two Important Lifecycle Phases,* Proc. of the ESF-Seminar 1990, Berlin: ESF

M. Nagl: *Characterization of the IPSEN Project,* in N. Madhavji/W. Schäfer/H. Weber (Eds.): Proc. 1st Int. Conf. on System Development Environments & Factories, 141-150, London: Pitman (1990)

M. Nagl: *Das Forschungsprojekt IPSEN,* Informatik-Forschung und Entwicklung, 5, 103-105 (1990)

M. Nagl: *Schlagwörter Ada und Smalltalk,* in P. Mertens u.a. (Hrsg.): Lexikon der Wirtschaftsinformatik, 2. Aufl., 8-9, 386-387 (1990)

M. Nagl/A. Schürr: *A Graph Grammar Specification Environment,* Aachener Informatik-Bericht Nr. 90-15, RWTH Aachan, 1990

M. Nagl/A. Schürr: *A Specification Environment for Graph Grammars,* in: Ehrig H., Kreowski H.-J., Rozenberg G. (Eds.):Proc. 4th Int. Workshop on Graph-Grammars and Their Application to Computer Science, erscheint in: LNCS, Springer Verlag (1991)

M. Nagl: *Software Architectures: Importance, Notions, Experiences,* ersch. Tagungsband Online 91 Konf., auch Aachener Informatik-Bericht Nr. 90-21, RWTH Aachen, 1990

M. Nagl: *Werkzeuge für die Softwareentwicklung,* Beitrag für das Sonderheft der RWTH-Themen zur Informatik (1990)

A. Schürr: *PROGRESS: A VHL-Language Based on Graph Grammars,* in: Ehrig H., Kreowski H.-J., Rozenberg G. (Eds.): Proc. 4th Int. Workshop on Graph-Grammars and Their Application to Computer Science, erscheint in: LNCS, Springer Verlag (1991)

6.3.1 Bücher

M. Nagl: *Ada: Eine Einführung in die Softwaretechnik-Programmiersprache Ada,* 3. Auflage, 345 S., Wiesbaden: Vieweg-Verlag (1991)

M. Nagl (Ed.): *Proc. WG'89 15th Int. Workshop on „Graphtheoretic Concepts in Computer Science",* Lecture Notes in Computer Science 411, 372 S., Berlin: Springer-Verlag (1990)

M. Nagl: *Softwaretechnik: Methodisches Programmieren im Großen* 387 S., Berlin: Springer-Verlag (1990)

Lehr- und Forschungsgebiet Informatik
am Lehrstuhl für Informatik III

1 Personal

- Leiter:

  Doz. Dr. L. Staiger
  (Vertretung des Lehr- und Forschungsgebietes)

Das Lehr- und Forschungsgebiet Informatik am Lehrstuhl III (vormals Prof. Dr. B. Walter) wird seit Beginn des Jahres durch Doz. Dr. L. Staiger vertreten. Das Berufungsverfahren wurde im Jahre 1990 abgeschlossen, so daß die Hoffnung besteht, daß die Professur im Sommersemester '91 besetzt sein wird.

2 Forschungsvorhaben

Projekt ω-Sprachen

ω-Sprachen wurden als Beschreibungen konstruktiver Klassen von Mengen unendlicher Wörter ausgenutzt, um Beziehungen zwischen der Kolmogorovschen Programmkomplexität sogenannter ω-Wörter und gewissen Informationsmaßen (wie z.B. Hausdorff-Dimension oder Entropie) sie enthaltender ω-Sprachen herzuleiten. Es ergibt sich dabei, daß die Hausdorff-Dimension einer ω-Sprache untere Abschätzungen für die Komplexität eines maximal komplexen ω-Wortes dieser Sprache liefert, während sich mit der Entropie unter gewissen Konstruktivitätsforderungen an die ω-Sprache obere Schranken ergeben. Auf diesem Wege lassen sich auch die Sätze von P. Martin-Löf über die Komplexität von Zufallsfolgen auf den Fall maximal komplexer ω-Wörter in regulären ω-Sprachen verallgemeinern.

Daneben wurde über die Erzeugung von ω-Sprachen aus formalen Sprachen mit Hilfe der ω-Potenz und des δ-Limes und die Schwierigkeiten, die sich bei der Beschreibung der ω-Potenz einer Sprache ergeben, im Januar 1991 zum Seminar "Automatentheorie und Anwendungen in Logik und Komplexitätstheorie", in Schloß Dagstuhl vorgetragen. Ferner wurde eine weitere Beziehung zwischen verschiedenen syntaktischen Kongruenzen für ω-Sprachen hergeleitet.

Tagungsbesuche:

10.-15.06.90 *1990 Information Theory Workshop, Veldhoven, Niederlande.*
Kurzvortrag: Orthogonal Spaces of Convolutional Codes

22.10.90 Diskussionstagung "*Algorithmische und statistische Informationstheorie*", TH Darmstadt.
Vortrag: Kolmogorov Komplexität und Hausdorff Dimension

Gastvorträge:

06.06.90 *am Institut für Informatik der Universität Würzburg*
Bei: J. Albert und K. Culik II
Vortrag: Zur Berechnung der Hausdorff Dimension von Bildern

22.08.90 *am Fachbereich Mathematik und Informatik der Fernuniversität Hagen*
Bei: K. Weihrauch
Vortrag: Kolmogorov Komplexität und Hausdorff Dimension

Fertiggestellte Artikel:

Staiger, L.: *Kolmogorov complexity and Hausdorff dimension, Inform. and Comput.*, to appear

Staiger, L. : *Syntactic congruences of ω-languages*, Submitted to: Theoretical Computer Science

Lehrstuhl für Informatik IV

Personal

- Lehrstuhlinhaber:

  Prof. Dr. Otto Spaniol

- Sekretariat:

  G. Hüning
  W. Makowski
  P. Zeidler

- Wissenschaftliche Mitarbeiter:

  T. Anwar
  P. Davids
  C. Engel
  D. Hübner
  We. Kremer
  Wo. Kremer
  D. Kreuer
  T. Meuser
  U. Quernheim
  C. Popien
  F. Reichert
  M. Rupprecht
  M. Schümmer
  P. Szczurko

- Technische Angestellte:

  H. Bolke
  K. Jakobs

Forschungsprojekte:

Projekt:

Echtzeitkommunikation in der Fabrikumgebung und innerhalb von Fahrzeugen

Beteiligt: **M. Schümmer, O. Spaniol**

Feldbussysteme und Intra-Vehikel-Kommunikationssysteme stellen heute zum Teil konkurrierende Ansätze für die realzeitkritische Übertragung von kurzen Prozeßdaten dar. Die Designphilosophie richtet sich dabei jeweils nach den Hauptanforderungen der Anwendung. Das Datenaufkommen erfordert in der Regel jedoch sich widersprechende Eigenschaften von einem Kommunikationssystem. Somit ist hier eine sorgfältige Abwägung der Anforderungen und der eingesetzten Kommunikationsmechanismen notwendig. Hierbei ist insbesondere den Echtzeitanforderungen der Prozeßsteuerungsumgebungen Rechnung zu tragen.

Die Standardisierung der Feldbussysteme kann heute noch nicht als beendet angesehen werden. Insbesondere der Streit über zentral oder dezentral organisierte Medienzugangsverfahren ist immer noch nicht beigelegt. Somit waren eine Reihe verschiedener Konzepte zu untersuchen. Die Ergebnisse, der größtenteils simulativ durchgeführten Analysen, sind inzwischen in einer Reihe von Diplom- und Studienarbeiten und einer Dissertation nachzulesen. Besondere Beachtung fanden wegen der internationalen und nationalen Bedeutung der Systeme: der deutsche PDV-Bus, der deutsche Standard PROFIBUS, MiniMAP als Ableger des amerikanischen MAP-Konzepts, der französische FIP und der japanischer Industriestandard IINET.

Die Untersuchungen wurden nach der ausführlichen Analyse der Medienzugriffsverfahren, die bei der reduzierten Kommunikationsarchitektur der Feldbussysteme von besonderer Bedeutung sind, auf die Anwendungsebene ausgedehnt. Wegen seiner internationalen Bedeutung wurde hierbei besonders auf den MMS-Standard und dem gegenüber MMS leicht modifizierten FMS eingegangen. Für MAP und PROFIBUS liegen auch hier die Ergebnisse bereits in Form von zwei Diplomarbeiten vor.

Als durchaus attraktive Alternative zu den oben genannten Feldbussystemen bieten sich die Intra-Vehikel-Kommunikationssysteme an. Der CAN (Controller Area Network) basiert auf einem nachrichten-orientierten Buszugangsverfahren, das zuverlässig immer nur der Nachricht mit der aktuell höchsten Priorität den Buszugriff gestattet. Um Kollisionen zu vermeiden, ist es nötig, daß jede Nachricht nach ihrer Funktionalität und Bedeutung eine eigene Priorität erhält. Dabei ergeben sich insbesondere Probleme der Zuordnung von Prioritäten, da es nicht sofort einsichtig erscheint, daß Alarmnachrichten der Bremsfunktionen denen der Lenkkontrolle vorgezogen werden sollen. Entscheidungshilfen für die a-priori-Prioritätszuordnung sind in der zum Projektthema entstandenen Dissertation nachzulesen.

In der Dissertation wurden auch die Überlegungen und Diskussionspunkte zu dem Thema der gesellschaftlichen Auswirkungen der fortschreitenden Prozeßautomatisierung festgehalten. Hierbei wurde ein besonderer Schwerpunkt auf die sozialen und ökonomischen Auswirkungen und die Risikoeinschätzung der automatischen Prozeßkontrolle gelegt.

Obwohl es auf diesem Bereich noch eine Reihe von untersuchenswerten Fragestellungen gibt (Updates der Standards, Zusammenhang Realzeitbetriebssysteme mit der Kommunikation, Realzeitfähigkeit durch Ressourcenkontrolle, ...), ist dieses Projekt am Lehrstuhl für Informatik IV mit dem Abschluß der verschiedenen Arbeiten beendet.

Projekt:

Satellitennetze und ISDN

Beteiligt: **M. Aghadavoodi Jolfaei, St. Jacobs, D. Kreuer, S. Martin, U. Quernheim, O. Spaniol, R. Vermöhlen**

Die Satellitentechnik erfährt im Augenblick einen starken Innovationsschub durch die Verbesserung der Empfangsanlagen und die Erschließung neuer Trägerfrequenzen, die bei gleichbleibender Sendeleistung den Einsatz wesentlich kleinerer Erdstationen ermöglichen. Dieser Trend führt dazu, daß auch kleinere Unternehmen die Vorteile der Satellitenkommunikation nutzen können. Dies ermöglicht die Einführung völlig neuer Dienste und Anwendungen. Beispiele hierfür sind europaweite Videokonferenzen im Unternehmensbereich, Übertragung von Bankaufträgen und Kreditkartenüberprüfungen, Mobilfunk im Speditionsbetrieb, d.h. ständige Erreichbarkeit der Lastkraftfahrzeuge durch die Spedition, usw.

Darüberhinaus werden zunehmend postalische Dienste auch im Inlandverkehr über Satelliten vermittelt (Bsp.: schneller Aufbau des Telefonnetzes in den neuen Bundesländern), die eine Einbindung von Satellitenstrecken in das gerade im Aufbau befindliche diensteintegrierende digitale Netz (ISDN) wünschenswert erscheinen lassen. Das ISDN wurde ursprünglich nicht für den Einsatz auf Satellitenstrecken vorgesehen, so daß die lange Signallaufzeit von rund 270 ms modifizierte Protokolle erfordert, um unnötige Verzögerungen zu vermeiden.

Schließlich erfordert die Verwendung des neuen Ka-Bandes (20/30 GHz), welches stärker von wetterbedingter Dämpfung betroffen ist als das bisher vorwiegend benutzte Ku-Band (10/12 GHz), den Einsatz adaptiver Schwundgegenmaßnahmen. Dabei wird z.B. versucht, durch Bandbreitenentzug (und damit einer leichten Verschlechterung der Signalqualität) von Stationen mit wetterbedingt gutem Empfang mehr Bandbreite für die von starken Dämpfungen betroffenen Stationen zur Verfügung zu stellen. Dadurch können die Link-Margins (d.h. diejenigen Sende- und Empfangsleistungen, die mindestens nötig sind, um den Betrieb der Strecke für einen vorgegebenen Anteil des Jahres sicherzustellen) weniger restriktiv gewählt werden, was einen erheblich höheren Durchsatz bei guten Empfangsbedingungen, d.h. über den weitaus größten Teil des Jahres, erlaubt.

Im Rahmen des Projekts werden Zugangsverfahren für VSAT-Netze (Very Small Aperture Terminals) bewertet und weiterentwickelt. Erfolgversprechend ist insbesondere das Dynamische PODA-Verfahren, welches wie PODA die Reservierung von Nutzdatenslots mittels wahlfreiem Zugang auf Reservierungsslots ermöglicht, wobei einige Teilnehmer höhere Prioritäten erhalten können. Zusätzlich erlaubt DYPO eine dynamische Anpassung des Verhältnisses Reservierungsslots/Nutzdatenslots an das augenblickliche Verkehrsaufkommen, so daß Reservierungen bei geringer Belastung schneller erfolgen können. Neben den Zugriffsverfahren soll in Zukunft die Fehlererkennung und -korrektur in VSAT-Netzen einen Schwerpunkt des Projekts bilden.

Bei der Bewertung von Protokollen für die Integration von Satellitenstrecken in das ISDN wurde das Zeichengabesystem Nr. 7 simulativ untersucht. Es zeigte sich, daß bedingt durch die hohe Signallaufzeit die Fenstergröße dieses Verfahrens unzureichend ist, womit es zu starken Durchsatzeinbußen kommt. Eine einfache Vergrößerung des Fensters erhöht zwar den Durchsatz, führt aber zu erheblichen Wartezeiten bei der Fehlerbehebung. Daher wurde ein neues Fehlerkorrekturverfahren entwickelt, das trotz vergrößertem Fenster sehr kurze Fehlerkorrekturzeiten ermöglicht; diese liegen durch präventive Wiederholungen im Mittel sogar unterhalb der Signallaufzeit der Satellitenstrecke. Das Verfahren ist dabei gegenüber Lastspitzen sehr stabil, da in diesem Fall der Durchsatz auf Kosten der Verzögerung adaptiv vergrößert wird.

In Zusammenarbeit mit dem Fernmeldetechnischen Zentralamt, der Schweizer PTT und der DLR Oberpfaffenhofen sollen schließlich adaptive Fehlersicherungsverfahren für das Ka-Band entwickelt und im Rahmen eines Experiments am Olympus-Forschungssatelliten auf einem marktgängigen TDMA-Controller zur Anwendung gebracht werden. Im Verlauf der Kooperation sollen nachfolgend weitere ARQ-Verfahren ihre Tauglichkeit auf einer realen Satellitenstrecke unter Beweis stellen.

Projekt:

PROMETHEUS PRO-COM

Beteiligt: **M. Achten, R. Drennhaus, M. Fichtner, S. Hoff, D. Hübner, J. Kaltwasser, J. Kassubek, We. Kremer, F. Reichert, A. Scunio, O. Spaniol**

PROMETHEUS (PROgraM for a European Traffic with Highest Efficiency and Unprecedented Safety) ist ein EUREKA-Projekt, das in Zusammenarbeit von Forschungsinstituten und Automobilfirmen durchgeführt wird.

Das Hauptziel von PROMETHEUS ist eine erhöhte Sicherheit im Straßenverkehr, die sich in einer Verminderung von Verkehrsunfällen niederschlagen wird.

Die „elektronischen Kopiloten" berechnen das Umfeld eines Fahrzeuges und warnen den Fahrer vor kritischen Manövern. Damit der „elektronische Kopilot" Manöver mit anderen abstimmen kann, muß er Daten auf funktechnischem Wege austauschen. Aufgabe des Lehrstuhles für Informatik IV im Rahmen dieses Projektes ist, geeignete Methoden für diese Fahrzeug-Fahrzeug-Kommunikation zu erarbeiten.

Aufgaben hierbei sind:

- Dezentrale Verwaltung der Kanalkapazität zur zuverlässigen Übertragung von Nachrichten zwischen Fahrzeugen. („Media-Access"); hierzu konnten bereits leistungsfähige Verfahren entwickelt werden; weitere Forschungen sind besonders auf dem Bereich der dezentralen Paketsynchronisation und entsprechendem Kanalzugriff erforderlich

- Entwicklung von Mechanismen zur Ausnutzung der broadcast-Eigenschaft des Funkkanals und zur Reduzierung des Quittungsaustauschs („Logical-Link-Control")

- Übergabe von Nachrichten von Fahrzeugen an zentrale Einrichtungen um z.B. Hilfe zu holen („Routing" ; „Internetting"), u.U. mittels Überbrückung von Distanzen zu Nachrichtenzielen, die sich außerhalb des Senderadius' eines Fahrzeuges befinden („Multi-Hop"-Verfahren); hierzu wurde ein spezialisiertes Multi-Hop-Verfahren entwickelt

- Optimierung der Verfahren mittels einer Systemsimulation, die die Bewegung der Fahrzeuge, die Funkkanaleigenschaften, die Nachrichtenlast und deren Auswirkungen auf das Leistungsvermögen des Gesamtsystems untersucht. (neue Version des Simulationssystems (ereignisgesteuert); Arbeitsname: MONET-3)

- Viele Einzelaspekte können nur in Zusammenarbeit mit anderen Forschungseinrichtungen geklärt werden. So werden u.a. funktechnische Aspekte und Verkehrsgesichtspunkte auf internationaler Ebene gemeinsam erarbeitet.

Die Laufzeit der zweiten Phase des PROMETHEUS-Projektes wird 1992 enden. Bis dahin sollen mehrere Demonstrationsversuche die technische Lösbarkeit beweisen und das Sicherheitspotential aufzeigen.

Projekt:

PICOM

Beteiligt: **Christian Engel, Bernd Heinrichs, Raschid Karabek, Spyros Komninos, Peter Langen, Michael Rupprecht**

PICOM (Parallelism In COMmunication protocols) ist ein DFG-Projekt, das in Kooperation mit der TU Dresden seit Oktober '90 durchgeführt wird.

Die Hauptziele sind die Entwicklung neuer Implementierungsstrategien und das Design von Kommunikationscontrollern zur parallelen Verarbeitung von Protokollen.

Hochgeschwindigkeitstechnologie auf Netzebene mit zunehmenden Übertragungsraten bei geringeren Fehlerraten bzw. höhere Anforderungen aufgrund heutiger und zukünftiger Applikationen (Multimedia-Anwendungen) zeigen, daß derzeitige Protokoll-Implementierungen den gehobenen Ansprüchen nicht genügen.

Aufgabenbereiche sind:

- Untersuchung der Funktionalität von existierenden OSI-konformen und neueren Protokollansätzen (u.a. lightweigth protocols wie XTP oder VMTP)
- Implementierungsnahe Leistungsbewertung existierender Protokolle oberhalb des Medienzugangs
- Weiterentwicklung, Definition und Einsatz einer speziellen Implementierungssprache für Protokolle. Höhere Petrinetzmodelle bilden aufgrund ihrer Möglichkeit, parallel ausführbare Aufgaben zu beschreiben, die funktionale Basis für diese Sprache (PENCIL ...**Pe**tri **N**et based **C**ommunication-Protocol **I**mplementation **L**anguage). Dadurch wird erreicht, daß Methoden, die für die kinematische und stochastische Analyse von Petrinetzen existieren, auch auf Kommunikationssoftware anwendbar sind. Damit ist die Korrektheit und Konformität der Software beweisbar.
- Abbildung der Kommunikationssoftware auf ein geeignetes Hardwaremodell (Mehrprozessorsystem); Bewertung verschiedener Ansätze
- Entwicklung eines Hardware-Emulators
- Entwicklung eines Compilers zur automatischen Umsetzung von PENCIL-Programmen in Zielcode (für ein spezielles Multiprozessorsystem)
- Entwicklung eines Simulationstools für PENCIL-Beschreibungen

Projekt:

DRIVE „Inter-Vehicle Communication"

Beteiligt: **A. Fasbender, R. Hager, We. Kremer, Wo. Kremer, O. Spaniol**

DRIVE (Dedicated Road Infrastructure for Vehicle Safety in Europe) ist ein Projekt der Europäischen Gemeinschaft, an dem alle großen Elektronikfirmen, Automobilfirmen, Verkehrsbehörden und eine Vielzahl von Hochschulinstituten aus ganz Europa beteiligt sind.

Ziel dieses Projektes ist die Verbesserung der Straßenverkehrssicherheit mit Hilfe der Fahrzeug-Fahrzeug Kommunikation. Dadurch soll den Fahrzeugen ermöglicht werden deren Position, Geschw., Fahrtrichtung, Gewicht, ... und die beabsichtigten Fahrmanöver den umliegenden Fahrzeugen mitzuteilen. Die so miteinander verbundenen On-Board Computer ermöglichen ein „kooperatives" Fahren.

Spezielle Probleme ergeben sich bei diesem System gegenüber herkömmlichen Kommunikationssystemen unter anderem durch folgende weitreichende Merkmale:

- Die Teilanwendungen für das „kooperative" Fahren haben ganz unterschiedliche Sicherheits- und Zeitanforderungen, wodurch ein Prioritäten-Management erforderlich wird.
- Durch die zeitkritischen Anwendungen muß der Datenaustausch in vordefinierten Zeitschranken erfolgen (Realzeitfähigkeit).
- Die Netzwerktopologie ist hochgradig dynamisch und dezentral zu verwalten (d.h.: dynamische Zugriffsverfahren, Adressierungen, Sendereichweiten-Anpassung, ...).

Der Lehrstuhl für Informatik IV behandelt in diesem Projekt folgende Arbeitsschwerpunkte:

- Spezifizierung der Teilapplikationen für das kooperative Fahren
- Definition der Kommunikationsanforderungen basierend auf den spezifizierten Anwendungen.
- Entwicklung eines Lasmodells unter zu Hilfenahme bekannter Verkehrsstatistiken und -simulationen.
- Entwicklung einer geeigneten Kommunikationsarchitektur unter Berücksichtigung des ISO-Referenzmodells.
- Entwicklung von Warteschlangen Abarbeitungsstrategieen zur Optimierung des Datenflusses bezüglich den ausgearbeiteten Zeit- und Prioritätsanforderungen.

- Ausarbeitung von Intergationskonzepten für den Zugang zu einem Infrastruktur unterstütztes Verkehrsinformations- und Leitsystem.
- Mitarbeit an der Entwicklung von Prototypen der Automobilindustrie zu Testzwecken.

Andere Aktivitäten

Gremientätigkeit (O. Spaniol)

- Deutscher Delegierter in IFIP-TC 6 (Data Communication)
- Sprecher GI/NTG-Fachgruppe 'Rechnernetze' (August 81 - März 85)
- Stellv. Sprecher Fachgruppe 'Rechnernetze' (seit März 85)
- Mitherausgeber 'Informatik-Spektrum'
- Mitherausgeber 'Informatik: Forschung und Entwicklung'
- Fachausschußvorsitz Informatik der Deutschen Forschungsgemeinschaft (seit April 1984)
- Referent ESPRIT-Projekte 'Office Systems' (seit 1984)
- Mitarbeit bzw. Vorsitz im Programmausschuß von nationalen und internationalen Tagungen
- Gutachter für 'IEEE Transactions on Communications' und zahlreiche andere Zeitschriften

Promotionen

A. Mann: *Funktion und Leistungsaspekte von Mobilfunknetzen*

W. Michael: *Eine Untersuchung über Multiprozessor-Busnetzwerke*

M. Schümmer: *Kommunikationssysteme in Echtzeitumgebungen - Modellierung und Bewertung von Fertigungs - und Intra - Vehikel - Kommunikationssystemen*

Diplomarbeiten

H. Bartz: *Entwurf und Implementierung eines Choirometers auf der Basis des Ultraschall-Scannings*

H.-J. Eisenbach: *Leistungsmessungen und Bewertung des Anwendungsprotokolls Manufacturing Message Specification (MMS) auf einer MAP-Implementierung innerhalb einer simulierten Fertigungsumgebung*

R. Goldau: *Entwurf einer Einführungsmaßnahme für ein rechnergestütztes Betriebsführungssystem*

R. Hager: *Entwicklung und Leistungsbewertung eines prioritätsorientierten Muliplexverfahrens für Mobilfunknetze*

B. Heinrichs: *Telefax in Lokalen Netzen*

S. Jacobs: *Modellierung und Leistungsbewertung von Zugangsprotokollen in VSAT-Satellitennetzen unter Berücksichtigung inhomogener Lasten*

J. Kaltwasser: *Einfluß von Raum-Multiplex-Interferenzen realer Kanäle im Mikrowellenbereich auf die Leistung von dezentralen TDMA-Verfahren für mobile Kommunikation*

J. Kassubek: *Entwicklung und Bewertung eines Netzwerkprotokolls zur Multihop-Übertragung von Datenpaketen mobiler Stationen an ausgezeichnete, ortsfeste Kommunikationseinrichtungen*

D. Kreuer: *Untersuchung des CCITT Signalisiersystems Nr. 7 im Hinblick auf eine Anwendung in Satellitennetzen; Simulation der Zeichengabestrecke*

Th. Meuser: *Leistungsbewertung getakteter Ringe als Hochgeschwindigkeitsnetze zur Integration heterogener Lastprofile*

R. Ridder: *Leistungsvewertung von Mini-MAP als Kommunikationssystem der Fertigungsumgebung*

Chr. Rose: *Entwicklung einer integrierten Datenbasis für ein mittelständisches Unternehmen der Eisen- und Metallindustrie*

P. Schmitz: *Entwurf und Implementierung eines Verfahrens zur Identikation und Zuordnung von zwei zweiwertigen Signalen mit starken stochastischen Schwankungen*
I. Sichelschmidt: *Datenkommunikation in einem Fernwirksystem: Realisierung eines Übertragungsprotokolls zur Meßdatenerfassung*

E. Weineck: *Entwicklung einer Konzeption zur statistischen Analyse umweltmedizinischer Daten*

Veröffentlichungen

Anwar, T., Jones, C. : *Two Routing Strategy Ideas for land-based mobile Packet Radio Networks,* Proc. of the IFIP TC6 Third International Conference on Information Network and Data Communication, Lillehammer, Norway, 26–29 March 1990

Anwar, T., Jones, C., Schreib, D., Strauch, G.: *Mobile Network Simulation in a City,* Proc. of the 1990 European Simulation Multiconference, Nuernberg, West Germany, June 10–13,1990

Anwar, T., Jones, C., Schreib, D., Strauch, G.: *Mobile PR-Network in a Densely Populated Urban Environment,* Proc. of the IEEE Global Telecommunications Conference 1990 (GLOBECOM '90), San Diego, California, U.S.A, December 2-5, 1990

Anwar, T., Jones, C. : *A Simulation of Mobile Network for Road Traffics in Overcrowded Regions,* Proc. of 1990 International Symposium on Information Theory and its Application (ISITA '90), Hawaii, 27-30 November 1990

Anwar, T., Jones, C. : *A Routing Strategy Idea for Land-based Mobile PR-nets Using a Self-supporting, Modular Navigational System for Automobiles,* Proc. of 1990 International Symposium on Information Theory and its Application (ISITA '90), Hawaii, 27-30 November 1990

Davids, P., Martini, P.: *Performance Analysis of DQDB,* IEEE Proc. of the ICX international Phoenix Conference on Computers and Communication (IPCC 90), Phoenix (Arizona), March 1990

Kauffels, F.-J., Kurbel, K. (Hrsg.), Spaniol, O.: *Architektur von Datenkommunikationssystemen,* Handbuch Wirtschaftsinformatik, Stuttgart: Poeschel, 1990, S. 893-927

Form, P., Grotheer, U., Hoff, S., Hübner, D., Reichert, F.: *Transponder Technique - An Approach to Integrate Relative Position Finding and Intervehicle Communication into a Single System,* Proc. of the 4th Prometheus Workshop, Compiègne 1990

Hager, R., Kremer, Wo.: *Studie eines Paketierten Sprach- und Datenmultiplexers mit adaptiver Prioritätensteuerung in LANs,* Intern.Workshop "Architecture of LAN for the reach of Campus and CIM", Dresden, Okt. 1990

Jakobs, K., Reichert, F.: *Directory Services for Mobile Communications,* Aachener Informatik Berichte Nr. 90-6, ISSN 0935-3232, Aachen 1990

Jakobs, K., Reichert, F.: *Directory Services in Mobile Communication,* Proc. of INDC-90, Information Network and Data Communication, Lillehammer 1990

Jakobs, K., Reichert, F.: *Using Directory Services in Mobile Communication,* Proc. of COMMNET'90, Budapest 1990

Jakobs, K., Reichert, F.,Spaniol, O.: *Der Computer fährt mit,* Aachener Informatik Berichte Nr. 90-12, ISSN 0935-3232, Aachen 1990

Kaltwasser, J., Hübner, D.,Reichert, F.: *Highway Intersection Aachen,* Proc. of the 3rd Prometheus Workshop, Turino 1990

Kaltwasser, J., Reichert, F., Schümmer, M.: *Performance Results of Prometheus Protocols,* Proc. of the 3rd Prometheus Workshop, Turino 1990

Kassubek, J., Jakobs, J., Reichert, F.: *MARION - Multiple Adaptive Routing in Mobile Networks,* Proc. of the 3rd Prometheus Workshop, Turino 1990

Reichert, F.: *PRO-COM Tasks or How to Achieve PROMETHEUS Functions,* Prometheus Report, Aachen 1990

Kauffels, F.-J., Kurbel, K. (Hrsg.), Spaniol, O.: *Architektur von Datenkommunikationssystemen,* Handbuch Wirtschaftsinformatik, Stuttgart: Poeschel, 1990, S. 893–927

Kremer, We.: *Analyse eines LAN Netzverkehrs im Hinblick auf Modellierbarkeit von Protokollabläufen,* 11.ITG/GI -Fachtagung Architektur von Rechensystemen, München 1990

Kremer, We.: *Explizite Quittungsmechanismen für Punkt-zu-Mehrpunkt Verbindungen auf lokalen Broadcastmedien,* Workshop Architektur von LAN für den Campus- und CIM-Bereich,Wissenschaftliche Beiträge zur Informatik, Dresden 1990

Kremer, We.: *Explicit Acknowledgment Mechanisms for Safe Data Exchange,* Proc. of the 3rd Prometheus Workshop, Turino 1990

Kremer, We.: *IMAP, Intelligent Multicast Acknowledgment Protocol: A New Approach Towards Increasing Reliability* , Proc. of the 4thPrometheus Workshop, Compiègne 1990

Kremer, We., Kremer, Wo.: *What Protocols are Already Candidates for VIC,* 8th DRIVE-Concertation Meeting in Brussels, May 1990

Kremer, We., Kremer, Wo.: *New Traffic Scenarios Due to Vehicle Inter-Communication Services,* 8th DRIVE-Concertation Meeting in Brussels, May 1990

Kremer, We., Kremer, Wo. (ed.): *Communication Functionalities and Requirements for Cooperative Control,* Prometheus Task Force "Cooperative Control", Aachen, Oktober 1990

Kremer, We., Kremer, Wo., et al.: *Cooperative Control Specifications,* Prometheus Task Force "Cooperative Control", Oktober 1990

Kremer, Wo.: *Common Specification Scheme,* Final DRIVE-Report, VIC/TH/009/c/F, Sept. 1990

Kremer, Wo.: *Communication Global Requirements - Network Aspects,* Final DRIVE-Report, VIC/TH/013/d/F, Nov. 1990

Kremer, Wo.: *Communication Global Requirements - Communication Characteristics,* Final DRIVE-Report, VIC/TH/015/d/F, Nov. 1990

Martini, P.; Meuser, T.: *Service Integration in FDDI,* Proceedings 15th Conference on Local Computer Networks, Minneapolis, IEEE Computer Society Press, Oktober 1990

Meuser, T. : *Leistungsbewertung getakteter Ringe als Hochgeschwindigkeitsnetze zur Integration heterogener Lastprofile,* Diplomarbeit an der RWTH Aachen 1990

Heinrichs, B.: *Telefax in Lokalen Netzen,* Diplomarbeit an der RWTH Aachen, April 1990

Heinrichs, B., Martini, P.: *Rechnergestützte Telefax-Kommunikation in Lokalen Netzen,* Workshop "Architektur von Lokalen Netzen für den Campus und CIM-Bereich", Dresden, 29.10.–02.11.1990

Heinrichs, B., Martini, P.: *Telefax in Local Area Networks,* Proceedings of the 15th Conference on Local Computer Networks, Minneapolis, Minnesota, 30.09.–03.10.1990

Schümmer, M.: *'Divide and Conquer within the Intra-Car Communication',* Proc. of the 3rd PROMETHEUS Workshop, pp.560–70, April 1990

Schümmer, M.: *'Multicasting in Realzeitumgebungen',* Tagungsband ECHTZEIT'90, Sindelfingen, S. 55–64, Juni 1990

Schümmer, M.: *'Kommunikationssysteme in Echtzeitumgebungen - Modellierung und Bewertung von Fertigungs- und Intra-Vehikel-Kommunikation',* Dissertation an der RWTH Aachen, 1990

Reinhardt, W., Schümmer, M.: *'Leistungsbewertung des Feldbusstandards PROFIBUS',* Tagungsband ECHTZEIT'90, Sindelfingen, S. 85–95, Juni 1990

Jacobs, St., Quernheim, U.: *ARQ Error Control for Multipoint Satellite Communication,* Proc. of the COMNET 90 Conference, Budapest

Kreuer, D., Quernheim, U.: *A Modified Error Correction Protocol for CCITT Signalling System No. 7 on Satellite Links,* erscheint als Conf. Proc. zur ECSC-2 91, Lüttich, Belgien

Quernheim, U.: *Performance Comparison of VSAT Access Protocols,* erscheint als Conf. Proc. zur ECSC-2 91, Lüttich, Belgien

Quernheim, U., Jacobs, St.: *Multicopy ARQ Error Control Techniques for Multipoint Satellite Links,* Proc. of the INDC-90 Conference, Lillehammer, Norwegen

Quernheim, U., Jacobs, St.: *Go-Back-N and Selective Repeat Error Control Techniques for Point-to-Multipoint Satellite Communications,* Elsevier Space Communications, Nr. 7, 1990, Elsevier Science Publishers B.V.

Quernheim, U., Jacobs, St., Kreuer, D.: *Adaptive ARQ-Fehlersicherungsverfahren für die Point-to-Multipoint-Satellitenkommunikation,* Aachener Informatik-Bericht Nr. 90-10 , Fachgruppe Informatik, RWTH-Aachen

Quernheim, U., Kreuer, D.: *Das CCITT-Signalisierungssystem Nr. 7 auf Satellitenstrecken; Simulation der Zeichengabestrecke,* Aachener Informatik-Bericht Nr. 90-9, Fachgruppe Informatik, RWTH-Aachen.

Rupprecht, M. : *Parallele Implementierung von Kommunikationsprotokollen,* Workshop Architektur von LANs für den Campus- und CIM-Bereich, Schloß Gaußig, 29.10. – 02.11. 1990

Lehrstuhl für angewandte Mathematik, insbesondere Informatik

1 Personal des Lehrstuhls

- Leiter:

  Prof. Dr. Walter Oberschelp

- Sekretariat:

  Marianne Kuckertz

- Mathematisch-Technischer Assistent:

  Rolf Eschmann (ab 1. 6. 90)

- Wissenschaftliche Mitarbeiter:

  Akad.ORat Dr. Volker Penner (zeitweilig beurlaubt)
  Dipl.-Math. Norbert Beinlich
  Dr. Susanne Kemmerich
  Dipl.-Inform. Toni Klingler
  Dr. Jörg Noack
  Dipl.-Inform. Michael Portz
  Dipl.-Math Andreas Schikarski
  Dipl.-Math. Dipl.-Inform. Robert Schneiders
  Priv.-Doz. Dr. Gottfried Vossen
  (beurlaubt zur Vertretung einer Professur in Düsseldorf bzw. Koblenz)

2 Spezielle Forschungsvorhaben und Interessenrichtungen

2.1 Große Diskrete Systeme

2.1.1 Algorithmen

Beteiligt: **W. Oberschelp und Mitarbeiter des Lehrstuhls**

Es werden Algorithmen für verschiedene Optimierungs- Fragestellungen entwickelt und analysiert.

Betreute Diplomarbeiten u.a.:

M. Buro: *Ein Beitrag zur Bestimmung von Rados $\Sigma(5)$ oder: Wie fängt man fleißige Biber*

Vgl. auch die im Anhang aufgeführten Diplomarbeiten von A. Pfaller, A. Podolecki, D. Schuckließ , G. Schulz, L. Vu.

2.1.2 Anwendung diskreter Verfahren in der Finite-Elemente-Methode (FEM)

Beteiligt: **N. Beinlich, R. Schneiders**

Der Lehrstuhl beschäftigt sich im Rahmen einer Kooperation mit dem IBF (Institut für Bildsame Formgebung, Prof. Kopp) mit der Lösung kombinatorischer (Optimierungs-)probleme, die bei der Rechnersimulation von Umformvorgängen mit der FEM auftreten. Schwerpunkte sind:

- Pre- und Postprocessing: Generierung von Vierecks- und Hexaedernetzen, Netzverfeinerung, Remeshing)
- Parallelisierung der FEM-Methode

Vorträge:

R. Schneiders: *Automatisches Remeshing bei der FEM-Simulation von Umformvorgängen.* 5. Symposium über Anwendung und rechentechnische Realisierung der Methode der finiten Elemente, Chemnitz, 18. – 20.9.1990

Veröffentlichung:

R. Schneiders, W. Oberschelp, R. Kopp, M. Becker: *New and Effective Remeshing Scheme for the Simulation of Metal Forming Processes;* submitted to Engineering with Computers

Diplomarbeiten:

R. Schneiders: *Automatische Generierung von zulässigen Vierecksnetzen für die FEM-Methode*

Vgl. auch die im Anhang aufgeführten Arbeiten von H. Dettmar, M. Kramer und B. Steindor.

2.1.3 Mehrstufige Verbindungsnetzwerke

Beteiligt: **A. Schikarski, H. Gawlik (Diplomand)**

In vielen parallelen Rechnersystemen kommunizieren die Prozessorelemente über ein mehrstufiges Verbindungsnetzwerk mit den Speicherzellen des Systems. Prominente Vertreter solcher Verbindungsnetzwerke wie das ω - Netzwerk und das Banyan-Netzwerk erlauben durch lokales Routen eine Vielzahl simultaner Zugriffe von $N(=2^n)$ Prozessoren auf N Speicherzellen, die als Permutationen von N Elementen modelliert werden können.

Es wurde eine Äquivalenzklasse von Verbindungsnetzwerken definiert, die viele bekannte Netzwerke enthält, und ein Algorithmus erstellt, der die Mitgliedschaft eines beliebig vorgegebenen Netzwerkes in dieser Klasse effizient entscheidet. Für die Mitglieder dieser Äquivalenzklasse können die realisierbaren Permutationsmengen leicht spezifiziert werden. Darüber hinaus gelang mit gruppentheoretischen Methoden eine tiefere Einsicht in die Struktur der jeweils realisierbaren Permutationsmengen.

Vortrag:

A. Schikarski: *Analyse einer topologischen Äquivalenzklasse mehrstufiger Verbindungsnetzwerke;* 10. Kolloquium über Kombinatorik, Bielefeld, 19.-23.11.90

Veröffentlichungen:

A. Schikarski, D. Wagner: *Parallel Matrix Inversion on Interconnection Networks.* Bericht Nr. 142, Schriften zur Informatik und Angewandten Mathematik

W. Oberschelp, B. Ruf, A. Schikarski: *Schaltkreistheorie.* Bericht Nr. 143, Schriften zur Informatik und Angewandten Mathematik

2.2 Kryptologie, Datensicherung

Beteiligt: **M. Portz**

Forschungsschwerpunkte der Arbeitsgemeinschaft Kryptologie waren 1990 die Erforschung einer neuen Klasse von Kryptosystemen auf der Basis von Verbindungsnetzwerken. In diesem Zusammenhang wurden verschiedene Arten der Kryptoanalyse untersucht, insbesondere die neue Methode der "Differential Cryptoanalysis". Daneben wurden die im Vorjahr begonnenen Projekte "Parallele Generierungen von

Primzahlen" und "Erzeugung statistisch hochwertiger Pseudozufallszahlen mittels kryptologischer Methode" weitergeführt und abgeschlossen.

Vorträge:

M. Portz: *Parallele Generierung von Primzahlen;* Seminar in Zusammenarbeit mit der TU Dresden und der TU Chemnitz in Aachen, 13.6.–15.6

M. Portz: *Erzeugung von binären Pseudo-Zufallszahlen mit Methoden der Kryptologie;* Seminar Datenfernverarbeitung, RWTH Aachen, 22.6

Veröffentlichungen:

M. Portz: *Parallelizing a Prime-Generating Algorithm,* Manuskript, Aachen 6/90

M. Portz, W. Oberschelp: *Mathematische Methoden der Datensicherheit;* erscheint im RWTH-Themenheft

2.3 Datenbanken

2.3.1 Datenmodellierung der Transaktionsverarbeitung für neue Anwendungsbereiche

Beteiligt: **G. Vossen**

Tätigkeit:

April 1989 – März 1990: Vertretung einer C3-Professur für Informatik an der Heinrich-Heine-Universität Düsseldorf

April 1990 – September 1990: Vertretung einer C4-Professur (Wirtschaftsinformatik) an der Erziehungswissenschaftlichen Hochschule Rheinland-Pfalz, Abteilung Koblenz

Oktober 1990 – März 1991: Vertretung einer C3-Professur an der Universität Koblenz-Landau

Arbeitsgebiete:

- Transaktionsverarbeitung in zentralisierten und verteilten Datenbank-Systemen
- Daten- und Transaktions-Modellierung bei objekt-orientierten Datenbanksystemen
- Datenbank-Entwurf
- Rechnerarchitektur

Laufende Forschungsprojekte:

1. Forschungskooperation mit der Sektion Mathematik der TU Dresden (Prof. Bachmann) sowie dem Fachbereich Informatik der U Rostock (Prof. Thalheim) zum Thema "Rechnergestützter Entwurf von Informationssystemen"

2. Forschungskooperation mit dem Instytut Informatyki des Politechnika Poznanska, Poznan, Polen (Profs. Cellary, Morzy) zum Thema "CAD-Datenbanksysteme"

Lehrveranstaltungen:

1. Wintersemester 1989/90:
   Vorlesung *Schnelle Algorithmen*,
   Seminar *Datenbanksysteme*,
   Universität Düsseldorf
2. Sommersemester 1990:
   Vorlesung *Transaktionsverarbeitung in Datenbanksystemen* (Datenbanken II),
   Vorlesung *Spezielle Kapitel der Datenbank-Forschung*,
   EWH Koblenz
3. Wintersemester 1990/91:
   Vorlesung *Objekt-orientierte Datenbank-Systeme*,
   Projektübung zur Programmierung (für Studenten im 3. Fachsemester Informatik),
   Universität Koblenz-Landau

Betreute Diplomarbeiten:

1. *Mehrversionen-Scheduling in Design-Datenbanken* (R. Schwarz, U Kiel, 4/90)
2. *Evaluierung eines Serialisierbarkeits-Schedulers für Update-Transaktionen auf konzeptioneller Ebene relationaler Datenbanken* (U. Fliegen, RWTH Aachen, 7/90)
3. *Semantik-basiertes Recovery relationaler Update-Transaktionen* (C. Lück, U Kiel, 12/90)
4. *Erkennung und Auflösung von Deadlocks (Verklemmungen) in verteilten Datenbanksystemen* (O. Wustrow, U Kiel, 1/91)
5. *Vergleich eines Serialisierbarkeits-Schedulers und eines ziel-orientierten Schedulers* (O. Fischer, RWTH Aachen, 1/91)
6. *Konzeption und Realisierung eines graphischen Werkzeugs zum Entwurf objekt-orientierter Datenbank-Schemata* (U. Masermann, U Koblenz, i.V.)
7. *Entwurf einer Erklärungs-Komponente zur Benutzerführung beim Datenbank-Entwurf* (H. Berning, RWTH Aachen, i.V.)
8. *Update-Transaktionen in verteilten relationalen Datenbanken* (H. Niemann, U Kiel, i.V.)

9. *Verwaltung von Replikation in verteilten Datenbanken* (H. Petersen, U Kiel, i.V.)

Veröffentlichungen:

G. Vossen, K.-U. Witt: *Das DB2-Handbuch*; 1. Auflage, Bonn: Addison-Wesley Verlag Deutschland GmbH, 1990, ISBN 3-89319-175-5 (457 Seiten)

W. Oberschelp, G. Vossen: *Rechneraufbau und Rechnerstrukturen*; 4. Auflage, München: R. Oldenbourg Verlag, 1990, ISBN 3-486-21743-7 (517 Seiten)

G. Vossen: *Transaktionsverarbeitung in Datenbanksystemen unter Ausnutzung semantischer Information*; 1. Auflage, Heidelberg: Hüthig Buch Verlag, 1990, Reihe Hochschultexte Informatik (P. Horster, Hrsg.) Band 22, ISBN 3-7785-1997-2 (171 Seiten)

G. Vossen: *Data Models, Database Languages and Database Management Systems*; 1. Auflage, Wokingham: Addison-Wesley Publishing Company, 1991, ISBN 0-201-41604-2 (590 Seiten)

G. Vossen, K.U. Witt: *Zur Klassifikation von strukturellen Erweiterungen des relationalen Datenmodells*; In: Informationstechnik **it** 32 (2) 1990, 97–106

G. Vossen, R. Schwarz: *Dynamic Scheduling of Transactions in Design Databases based on Version Consistency* (Extended Abstract); In: Proc. International Conference on Data Base and Expert System Applications (DEXA), Wien, Springer-Verlag 1990, 196–202

G. Vossen: *Modellbildung für Datenbank-Transaktionen*; wird erscheinen in: G. Vossen, K.-U. Witt (Hrsg.), *Entwicklungstendenzen bei Datenbank-Systemen*, Oldenbourg-Verlag, München; verfügbar als Informatik-Bericht 7/90, EWH Koblenz, August 1990

G. Vossen: *Towards Dynamic Schedule Correctness for Relational Update Transactions*; erschienen als Informatik-Bericht 4/90, EWH Koblenz, Mai 1990

G. Vossen: *Bibliography on Object-Oriented Database Management*; wird erscheinen in: ACM SIGMOD Record 20 (1) 1991; verfügbar als Informatik-Bericht 10/90, Universität Koblenz-Landau, Oktober 1990

Vorträge:

23.01.: *"Mein Auto ist objekt-orientiert" — Perspektiven bei Datenbanksystemen*; Vortrag in der Augustinus-Buchhandlung, Aachen (zusammen mit K.-U. Witt)

04.04.: *Transaktionsverarbeitung in Datenbanksystemen unter Ausnutzung semantischer Information*; eingeladener Vortrag (Prof. Dr. J.W. Schmidt), Institut für Informatik, Universität Hamburg

25.04.: (Titel wie 04.04.); eingeladener Vortrag (Prof. Dr. H. Kleine Büning), Fachbereich 11 (Mathematik), Universität-Gesamthochschule Duisburg

20.06.: *Objekt-orientierte Datenbank-Systeme*; Probevortrag i. Z. mit den mündlichen Habilitationsleistungen, RWTH Aachen

02.07.: *Transaktionsverarbeitung in Entwurfs-Datenbanken*; eingeladener Vortrag (Prof. Dr. K. Waldschmidt), Fachbereich Informatik, Universität Frankfurt

17.07.: (Titel wie 04.04.); eingeladener Vortrag (Prof. Dr. H. Ganzinger), Fachbereich Informatik, Universität Dortmund

30.08.: *Dynamic Scheduling of Transactions in Design Databases based on Version Consistency*; Vortrag auf der 1st International Conference on Database and Expert Systems Applications (DEXA), Wien, Österreich

08.10.: *Dynamische Korrektheit von Schedules relationaler Transaktionen*; eingeladener Vortrag (Prof. Dr. A. Schönhage), Institut für Informatik, Universität Bonn

20.10.: (Titel wie 20.06.); eingeladener Vortrag (Prof. Dr. N. Schmitz), Fachbereich Mathematik, Universität Münster

02.11.: *Transaction Processing in Database Systems using Semantic Information*; eingeladener Vortrag (Prof. Dr. W. Cellary), Instytut Informatyki, Politechnika Poznanska, Poznan, Polen

08.11.: (Titel wie 20.06.); eingeladener Vortrag (Prof. Dr. A. Beutelspacher), Fachbereich Mathematik, Universität Gießen

13.11.: *Einführung in das objekt-orientierte Datenbanksystem O_2*; Vortrag in der CASE-Arbeitsgemeinschaft des Fachbereichs Informatik der Universität Koblenz-Landau

Lehrerfortbildung:

1. Unterregionalisierte Fortbildung in den Regierungsbezirken Düsseldorf und Köln:

   Thema: *Datenbanksysteme — Eine Einführung*
   06.02. Planungstagung für Lehrerfortbildung in Burscheid
   12.02. Erich-Kästner-Gymnasium, Köln
   12.03. Gymnasium Schwertstraße, Solingen
   19.03. Couven-Gymnasium, Aachen

20.03. Gymnasium Siegesstraße, Wuppertal
23.04. Städtisches Gymnasium, Bedburg
22.08. Kopernikus-Gymnasium, Ratingen-Lintorf
17.10. Ernst-Moritz-Arndt-Gymnasium, Bonn
22.10. Werner-Heisenberg-Schule, Leverkusen
09.11. Integrierte Gesamtschule, Bonn-Beuel
23.11. Kopernikus-Gymnasium, Duisburg

2. Präsenzveranstaltungen für Fernstudienkurse am Landesinstitut für Schule und Weiterbildung in Soest:

   - Thema: *Grundlagen von Datenbanksystemen*
     07. – 08.02., 16. – 17.08., 25. – 26.10.
   - Thema: *Grundlagen von Betriebssystemen*
     10. – 11.12.

2.3.2 Datenbanken und natürliche Sprache

Beteiligt: **J. Noack**

Kommunikation mit (relationalen) Datenbanken in natürlicher deutscher Sprache, Unifikationsgrammatiken, Datenbanken, Datenstrukturen, Datenbank-Schnittstellen. Erweiterung der Dialogmöglichkeiten des natürlichsprachlichen Systems NATHAN, insbesondere Behandlung von Updates auf der Basis von Sichtabhängigkeitsgraphen.

Lehrveranstaltungen:

WS 89/90 Seminar “Unifikationsbasierte Grammatikformalismen” (in Zusammenarbeit mit dem Institut für medizinische Statistik und Dokumentation)
SS 90 Seminar “Datenstrukturen für Nonstandard-Datenbanken”
WS 90/91 Seminar “Seminar deduktive und objektorientierte Datenbanken”

Betreute Diplomarbeiten:

M. Reyers: *Verwendung morphologischer Informationen zur Verfeinerung der syntaktischen und semantischen Analyse eines natürlichsprachlichen Datenbankzugangssystems*

H. Wings: *Übersetzung natürlichsprachlicher Datenbank-Updates durch Sichtabhängigkeitsgraphen*

Ferner die im Anhang aufgeführten Diplomarbeiten von R. Kuron, R. Pütz und W. Zimmermann.

Veröffentlichungen:

J. Noack: *Die Behandlung von semantisch unvollständigen Anfragen in einer transportablen Datenbank–Schnittstelle,* in H. Marburger(Hrsg.): GWAI–90, IFB 251, Springer Verlag, 1990, 221–230

J. Noack: *Natürlichsprachliche Interaktion mit Datenbanken,* Aachener Informatik–Bericht, Nr. 90–24

Vorträge:

1. 2.90: *NATHAN: Natürlichsprachiger Zugang zu relationalen Datenbanken;* Informatik–Kolloquium, Universität Hildesheim

30.11.90: (W. Oberschelp, J. Noack) *Semantische Analyse für eine natürlichsprachliche Datenbank-Schnittstelle,* Tag der Informatik, RWTH Aachen

19.12.90: *Natürlichsprachliche Interaktion mit Datenbanken;* Informatik–Kolloquium, Universität Koblenz–Landau

2.4 Paralleles PROLOG (DFG-Projekt)

Beteiligt: **Penner, Klingler**

Die globale Zielsetzung des Forschungsvorhabens ist die Erarbeitung von Lösungsansätzen zur parallelen Implementierung logischer Programmiersprachen, die eine verbesserte Nutzung paralleler Architekturen gewährleisten sollen. Im Rahmen des Projekts werden, stellvertretend für die Klasse der lose gekoppelten Parallelrechner, Techniken zur Implementierung der Sprache Prolog auf rekonfigurierbaren Transputer-Netzen entwickelt und erprobt.

Ein zentraler Aspekt ist dabei die Einbeziehung von Möglichkeiten zur Laufzeitrekonfigurierung von Transputersystemen im Hinblick auf die gleichmäßige Auslastung der Prozessoren des Netzes.

Das Hauptergebnis der bisherigen Arbeit besteht in einem Gesamtkonzept zur verteilten Implementation von Prolog für lose gekoppelte Multiprozessorsysteme. Aufgrund der Erfahrungen mit dem im Rahmen des Projekts entwickelten Parallelen Prolog-Systems PPS empfiehlt sich eine Implementation unter Verwendung eines virtuellen Assoziativspeichers und von Multicasttechniken.

Der verfolgte Ansatz beruht auf einem Analyseverfahren, mit dessen Hilfe skeletale Ableitungsbäume erzeugt werden können, die in Form von Vorgriffen die zu erwartenden Aufrufe beschreiben. Zugleich geben die Bäume Aufschluß über parallele Teilziele und unabhängige Datenzugriffe, die parallel ausgeführt werden können. Die diesbezügliche Beschreibung im skeletalen Baum kann zur Verteilung der Daten im virtuellen Speicher verwendet werden mit dem Ziel, dieses Parallelpotential weitgehend nutzen zu können. Zur Implementation werden Hashverfahren verwendet, und weiter ist geplant, Multicasttechniken zur Realisierung von Routing- und Backtracking-Verfahren zu nutzen. Erste Versuche zeigen, daß das Verbindungsnetzwerk des Projektrechners über Multicast-Eigenschaften verfügt.

Vorträge:

T. Klingler: *Verteilte Implementierung von Prolog;* Seminar in Zusammenarbeit mit der TU Dresden und der TU Chemnitz in Aachen, 13.6. – 15.6.90

V. Penner: *Lastverteilung im Parallelen Prolog-System;* (s.o.)

Diplomarbeiten:

Carl Friedrich Steffens:*Implementation eines Verfahrens zur Ableitung von 'modes' für Prolog-Programme*

Ulrich Hermanns:*Routing in rekonfigurierbaren Transputer-Netzwerken*

2.5 Logik für AI

Beteiligt: **S. Kemmerich**

Automatische Beweisverfahren, Logikprogrammierung, deduktive, bzw. wissensbasierte Datenbanken, nicht-klassische Logiken, insb. epistemische und nicht-monotone Logiken.

2.6 Bildverarbeitung

Beteiligt: **W. Oberschelp und verschiedene Kollegen und Mitarbeiter von Instituten der Medizinischen Fakultät (insbesondere M. Drobnitzki, R. Grebe, H. Handels)**

Entwicklung effizienter Algorithmen für Analyse-Aufgaben der 2D- und 3D-Bildverarbeitung unter Einbeziehung paralleler Rechnerstrukturen. Anwendungen insbesondere in der medizinischen Diagnostik (Zusammenarbeit hier mit den Professoren Repges, Günter, Schmid-Schönbein, Rau, Böcking).

Diplomarbeiten:

gemeinsam mit den o.g. Lehrstühlen.
Vgl. die im Anhang genannten Arbeiten von H. Auer, A. Hiestermann, B. Höges, J. Honsbrok, A. Kiep, E. Müllender, M. Schell, J. Schulz, J. Theis, G. Viehöver.

Veröffentlichungen:

W. Oberschelp: *Grundlagen der parallelen Datenverarbeitung,* Informatik-Fachberichte 237 (1990), p. 1 - 8.

3 Weitere Aktivitäten

Der Lehrstuhl ist auch tätig im Bereich der Mathematik unter folgender Charakterisierung:

Arbeitsgebiete:

Diskrete Mathematik, Daten- und Rechnerstrukturen, Anwendungen in der Informatik

Forschungsschwerpunkte:

Kombinatorische Abzählungstheorie und Optimierung, Kryptologie, Logik und Datenbanken, Parallele Algorithmen, Computergraphik und Bildverarbeitung, Anwendungen in Medizin und Maschinenbau

Ausbildung:

Es werden auch Bewerber für eine Erweiterungsprüfung Informatik im Gymnasial-Lehramt betreut.

Vorträge von W. Oberschelp (soweit nicht bereits genannt):

31. 1.90: Ringvorlesung RWTH Aachen: Kurt Gödel, ein Wegbereiter der Informatik

27. 9.90: MNU - Fachtagung Hannover: Informatik und die Schulfächer des naturwissenschaftlichen Aufgabenfeldes

1.10.90: Bundes-Fachleitertagung Mathematik, Burg Gemen: Rechenverfahren und Formelalgorithmen als Unterrichtsgegenstände

27.11.90: Ringvorlesung RWTH Aachen: Gedanken zum Grundrecht auf informationelle Selbstbestimmung

Außerdem wurden zwei Kontaktseminare mit Wissenschaftlern der Technischen Hochschulen Dresden und Chemnitz durchgeführt:

13. – 15. 6.90: RWTH Aachen

16. – 18.11.90: Gästehaus DJH Oberwesel

Lehrgebiet Theoretische Informatik

1 Personal

- Leiterin:

  Prof. Dr. Britta Schinzel

- Sekretariat:

  Annegret Freytag

- Wissenschaftliche Mitarbeiter:

  Dipl.-Inform. Friedrich Strauß
  Dipl.-Inform. Roman Majewski
  Dipl.-Math. Frank Piron

2 Forschungsvorhaben

Die Forschungsaktivitäten des Lehrgebiets liegen vorwiegend auf dem Gebiet der Künstlichen Intelligenz, befassen sich also mit dem Versuch der Nachbildung von kognitiven und intelligenten Fähigkeiten des Menschen, hier insbesondere mit

- Lernvorgängen
- Konnektionistischer Mustererkennung
- Natürlichsprachlichen Systemen
- Wissensbasierten Systemen

In der Arbeitsgemeinschaft Künstliche Intelligenz werden Ansätze und Ergebnisse diskutiert und neuere Literatur besprochen. Es bestehen ferner interdisziplinäre Kontakte zum Germanistischen Institut, zum Lehrstuhl für Anatomie I, zum Institut für Biologie II (Zoologie), sowie zum Institut für Soziologie und schließlich zum Institut für Kunststoffverarbeitung.

2.1 Projekt: Funktioneller Hirnatlas

Beteiligt: **O. Brendel, D. Graf von Keyserlingk, F. Klomp, F. Piron, B. Schinzel**

In diesem Projekt, das den Aufbau eines hybriden Informationssystems für die Neurowissenschaften zum Gegenstand hat, wurden folgende Arbeiten durchgeführt:

Zunächst wurden die Arbeiten an der Syntax einer Anfragesprache abgeschlossen, die natürlichen Sprachen nähersteht als SQL- Anweisungsfolgen. Hierbei handelt es sich um eine Mustersprache, die auf Schlüsselwörtern oder Schlüsselwortkombinationen basiert. Nach der Fertigstellung eines Entity-Relationship-Modells für neuromedizinisches Literaturwissen einschließlich der Übersetzung in ein relationales Datenbankschema, konnte für diese Systemkomponente auch die Semantik der Anfragesprache definiert und implementiert werden. Dabei war als wesentliches Problem die Festlegung geeigneter heuristischer Joins in Abhängigkeit von bestimmten Schlüsselwörtern zu lösen. Schließlich wurde ein Übersetzungsalgorithmus für den Übergang von Ausdrücken der Anfragesprache zu SQL-Anweisungsfolgen implementiert.

Im Bereich des symbolischen Hirnatlanten wurde zunächst die Theorie der neuroanatomischen Topographie in der Prädikatenlogik erster Stufe axiomatisiert. Dazu gehören die morphologische Heterarchie, verschiedene Verbindungswege (Blut-, Nerven- u.a.) und räumliche Nachbarschaften. Die grundlegenden Objekte sind Zentren oder Areale. Es wurde zunächst ein mengentheoretisches Standardmodell in jeder endlichen Mächtigkeit angegeben, einschließlich Modellnachweis. Schließlich wurde dieses Modell als symbolischer Hirnatlas in C auf einer SUN4 implementiert.

Veröffentlichungen:

O. Brendel: *„Funktioneller Hirnatlas - Axiomatisierung und Repräsentation der Topographie"*, Diplomarbeit an der RWTH-Aachen, 1990

F. Klomp: *„Funktioneller Hirnatlas - Die Datenbankkomponente"*, Diplomarbeit an der RWTH-Aachen, 1990

2.2 Lernen durch Analogie

Beteiligt: **R. Majewski, A. Dollen, H. Schimpe**

Ein System, das in der Lage ist, Analogien zu bilden, könnte dies nutzen, um schneller auf vorhandenes Wissen zuzugreifen, Inkonsistenzen zu verarbeiten oder neues Wissen zu inferieren. Es ist daher durchaus wünschenswert, wissensbasierte Systeme mit der Fähigkeit auszustatten, Analogien zwischen Domänen erkennen zu können. Dies setzt voraus, daß das System vor allem sehr gut auf bereits vorhandenes Wissen zugreifen kann. Das betrifft sowohl das explizite Domänenwissen, als auch Wissen über bereits durchgeführte Inferenzen, implizit vorhandenes Wissen, dieses wird normalerweise nicht genutzt, und schon gebildeten Analogien. Insbesondere sollte der analoge Inferenzprozeß nicht isoliert durchgeführt werden, sondern vielmehr unter Berücksichtigung der bereits im System vorhandenen Inferenzarten.

Grundlegende Bedeutung hat in einem Analogien bildenden System vor allem der Wissensrepräsentationsmechanismus. Er muß das Erkennen von Ähnlichkeiten unterstützen, den Zugriff auf Ergebnisse durchgeführter Inferenzen, analog oder deduktiv, ermöglichen, und zusätzliche, explizite Charakterisierungen des vorhandenen Wissens erlauben.

Es wurde ein Repräsentationsmechanismus entwickelt, der die oben aufgeführten Forderungen berücksichtigt. Dabei sind die beiden folgenden Darstellungsebenen zu unterscheiden: die Benutzerebene und die Systemebene. Der Benutzer formuliert sein Wissen auf der Benutzerebene in einem Formalismus, der auf der Prädikatenlogik beruht. Dieses Wissen wird dann vom System in den Repräsentationsmechanismus der Systemebene übersetzt und in die Datenbasis eingetragen. Dies ist notwendig, da die Systemebenendarstellung sehr ausführlich ist und daher einem Benutzer nicht zuzumuten ist, sein Wissen in dieser Weise einzugeben. Sämtliche Inferenzen werden dann auf Systemebene durchgeführt.

Neben der Darstellung des Domänenwissens enthält die Systemebene auch Informationen über bereits duchgeführte Inferenzen. Dabei werden neue Inferenzen so weit wie möglich auf vorhandene zurückgeführt, wodurch Ähnlichkeiten von Domänen auch in einem größeren Zusammenhang deutlich werden. Die Darstellung von Inferenzen ist dabei so realisiert, daß auch auf deren Teile effizient zurückgegriffen werden kann.

Vorträge:

Achim Dollen: Expansion eines Deduktionssystems durch Integration implizit gegebener Determinationen.

Roman Majewski: Über Zusammenhänge höherer Ordnung und ihre Rolle bei der Analogiebildung.

Die Vorträge wurden im Rahmen des Arbeitstreffens der Fachgruppe Maschinelles Lernen der GI, Bonn, Juli/August 1990, gehalten.

Veröffentlichungen:

Achim Dollen: *„Expansion eines Deduktionssystems durch Integration implizit gegebener Determinationen"*, Diplomarbeit an der RWTH- Aachen, 1990

2.3 Konnektionistische Mustererkennung

Beteiligt: **F. Beuting, F. Piron, K. Rausch, B. Schinzel**

Im Berichtzeitraum wurden die Arbeiten an einem konnektionistischen Modell für die Erregung visueller Aufmerksamkeit abgeschlossen.

Es wurde ein Modell untersucht, das auf einer verteilten, iterativen Schwerpunktsberechnung für auf einem Eingaberaster gegebene Figuren basiert. In den Simulationen ergab sich zunächst für einige Fälle das gewünschte Verhalten. So entstand z.B. bei „eiförmigen" Figuren in den ersten Updateschritten zunächst ein Aktivitätsmaximum im „augenfälligen" Schwerpunkt, das dann im weiteren Verlauf in den mathematischen Schwerpunkt abwanderte und sich dort stabilisierte. Bei „Hanteln" bildeten sich anfänglich zwei Aktivitätsmaxima in den Schwerpunkten der optischen Fast-Zusammenhangskomponenten aus, die dann schließlich miteinander verschmolzen.

Die Simulationen wurden jeweils für orthogonale und hexagonale Eingaberaster durchgeführt und es stellte sich heraus, daß für dieses Modell das hexagonale Raster einen optimalen Kompromiß zwischen Interkonnektivitätskosten und Geschwindigkeit bei der Ausbildung optischer Schwerpunkte darstellt.

Es konnte weiter für eingeschränkte Figurenklassen bewiesen werden, daß sich asymptotisch der mathematische Schwerpunkt als Aktivitätsmaximum stabilisiert, obwohl die Aktivitätsmuster einzelner Units (Pixel) keine fallende Monotonie mit dem euklidischen Abstand aufweisen. Gegenbeispiele hierzu entstehen im orthogonalen Modell nach 5 - im hexagonalen Modell nach 12 Iterationsschritten.

Schließlich wurde das Modell noch durch die Hinzunahme variabler Kantengewichte verfeinert, was eine Relaxation und somit eine Dämpfung der Interferenz mehrerer Figuren im Eingaberaster ermöglichte. Auf dieses Modell wurde eine Lokalisierungskaskade in Anlehnung an das Neocognitron von K. Fukushima aufgesetzt, so daß nun eine konkrete Anwendung in der Robotersteuerung auf konnektionistischer Basis denkbar ist.

Alle Algorithmen sind in Turbo-Pascal auf einem IBM-AT implementiert.

Veröffentlichungen:

F. Beuting, K. Rausch: *„Ein neuronales Modell für die Erregung visueller Aufmerksamkeit"*, I und II, Diplomarbeiten an der RWTH- Aachen, 1990

F. Beuting, F. Piron, K. Rausch: *„Ein neuronales Modell zur Feststellung von für die visuelle Aufmerksamkeit relevanten Figureneigenschaften"*, Aachener Informatik Bericht 91-2 (wird z.Zt. gedruckt), Aachen, 1991

2.4 Natürlichsprachliche Systeme

Beteiligt: **F. Piron, R.Widdig**

Hier wurden zunächst die Arbeiten zur Akquisition und Repräsentation grammatischen Wissens für die Linguistische Datenverarbeitung (Grammmatisches Telefon) abgeschlossen. Dafür wurden nun zusätzlich ein Lexikon für Valenzmuster und eine Kontextheterarchie implementiert. Die Kontextheterarchie ist durch den Systemverwalter graphisch editierbar (hierbei wurden Smalltalk-Methoden verwendet) und die Kontexte können als Restriktionen in den Valenzmustern der eingetragenen Verben aufgeführt werden. Ferner wurden die semantischen Kategorisierungen um disjunktive Kombinationen erweitert.

Weiter wurde eine syntaktische Metasprache als Grundlage einer formal-deskriptiven Grammatik entwickelt. Ausgangspunkt hierfür war die monadisch-zweistufige Logik des Büchi-Kalküls, die wir um sogenannte Zählerrelationen erweitert haben. Die Äquivalenz zu einem geeigneten Automatenmodell (endliche Automaten mit Zählerrelationen) konnte gezeigt werden. Der Beweis ist konstruktiv, so daß ein Compiler für die Übersetzung von erweiterten Büchi-Formeln in die entsprechenden Automaten vorliegt. Die beschreibbare, formale Sprachklasse K liegt innerhalb von CH-1 , oberhalb von CH-3 und folgende Mengen sind nichtleer:

-K ohne CH-2

-CH-1 ohne K

-CH-2 ohne K

-Der Durchschnitt der echt-kontextfreien Sprachen mit K

Die logische Beschreibungssprache eignet sich besonders zur Spezifikation von Valenzmustern und ermöglicht einen modularen, verteilten Aufbau von Grammatiken alternativ zu einem Phrasenstrukturansatz.

Veröffentlichungen:

R. Widdig: *„Akquisition und Repräsentation natürlichsprachlicher Grammatiken"*, Diplomarbeit an der RWTH-Aachen, 1990

F. Piron, C. Stetter, B. Schinzel: *„Maschinelle Verarbeitung natürlicher Schriftsprache"*, Artikel für die RWTH-Themen, erscheint bis 5/91

2.5 Frauen und Informatik

Beteiligt: **Ch. Funken (Soziologie), K. Hammerich (Soziologie), H. Westram (Gymnasium Monschau), M. Schümmer (Informatik IV), B. Schinzel**

Die interdisziplinäre Forschungsgruppe führte eine empirische Untersuchung zur geschlechtsspezifischen Differenzierung im Informatikunericht durch. Zu diesem Zweck wurden 1132 Schülerinnen und Schüler der Jahrgangsstufe 11.1 an 18 Schulen der Aachener Region befragt.

Das Interesse galt besonders den Vorstellungen der SchülerInnen zu

-dem Unterrichtsfach Informatik

-der Arbeitsweise am Computer

-Technikakzeptanz

-Berufsvorstellungen über InformatikerInnen

etc.

Die Ergebnisse stellen einen wichtigen Beitrag zur Koedukationsdebatte dar, ein Thema, das gerade in Bezug auf die Informatik stark diskutiert wird. Unter anderem konnte nachgewiesen werden, daß Mädchen nicht etwa weniger Zugang zur Computertechnik haben, sondern daß sich erst bei zunehmendem Jungenanteil im Informatikunterricht Selbstvertrauen, Motivation und Technikakzeptanz reduzieren.

Ein weiterer Abbruchgrund für Mädchen sind die außerschulischen Ressourcen, d.h. der fehlende Computer zu Hause. Im Gegensatz hierzu haben über 70 Prozent der Jungen persönlich einen PC, den sie oft auch fanatisch in ihrer Freizeit nutzen. Diese unterschiedlichen „Trainings" -zeiten wirken sich maßgeblich im Unterricht aus, z.B. bei den Jungen durch Wissensvorsprung und Dominanz.

Als nächstes ist eine Studie zur Situation von Informatikstudentinnen und -studenten geplant. Hierbei interessieren u.a. Gründe und Ausmaß von Studienabbuch und Studienwechsel.

Es wurde ein Gutachten für das BMBW „Zur Lage des wissenschaftlichen weiblichen Nachwuchses in der Informatik" erstellt. Hierbei handelt es sich um die 1. Enquete zum Thema.

2.6 Abhängigkeit der geistigen Leistungsfähigkeit vom Informationsangebot durch den Computer

Beteiligt: **B. Schinzel, H. Stieve (Zoologie), F. Strauß**

Bei der Arbeit mit dem Rechner ist die Form des Informationsangebots von Wichtigkeit für das Problemlöseverhalten und den Streß des Menschen, der damit arbeitet, wobei die Größen Problemlöseverhalten und Streß sich gegenseitig beeinflussen.

Je nach Funktionalität der Programme und Ausprägung der Schnittstelle Mensch-Maschine ist der Computereinsatz unterschiedlich belastend für den Menschen.

Man kann annehmen, daß die Variation verschiedener Arten des Informationsangebots unterschiedliche Auswirkungen auf das menschliche Verhalten hat. Deshalb ist es zunächst notwendig, Informationsarten zu unterscheiden, sowie verschiedene Aspekte von Informationen zu betrachten, um anhand einer solchen Typologie gewisse Eigenschaften menschlicher Informationsverarbeitung zu erfassen. Eine Veränderung der aus der Typologie gewonnenen verschiedenen Aspekte des Informationsangebotes kann sich nun positiv oder negativ auf die Interaktion auswirken, indem einerseits die Qualität des Problemlöseverhaltens beim Menschen verändert wird und andererseits die Streßbelastung erhöht oder erniedrigt wird. Geht man von Texten als Informationsträgern aus, so läßt sich vor dem Hintergrund einer rationalistisch-analytischen Sprachphilosophie eine Einteilung in Syntax, Semantik und Pragmatik vornehmen. Die Ansätze der Informationstheorie berücksichtigen kaum semantische und pragmatische Aspekte, die aber gerade hier zur Untersuchung kommen sollen. Es ist wohl nicht möglich einen universell gültigen Informationsbegriff, der alle drei Aspekte, oder auch nur semantische und pragmatische Aspekte berücksichtigt, unabhängig vom Gegenstandsbereich und vom Gebrauchszusammenhang zu spezifizieren.

Daher beschränken wir uns hier, orientiert am oben ausgeführten Erkenntnisinteresse, auf eine Implementierung des Spiels Master Mind als Gegenstandsbereich, und die Computerinteraktion zwischen Programm als erstem Spieler und Benutzer als zweitem Spieler als Gebrauchszusammenhang ein. Der semantische und pragmatische Informationsgehalt einer Nachricht vom ersten Spieler soll ein Maß auf einer am Wissensbegriff der Künstlichen Intelligenz orientierten Formalisierung des Textinhalts sein.

Für die Wissensdarstellung dieses Spieles verwenden wir die Prädikatenlogik erster Stufe. Jede Spielsituation besteht aus zwei Kombinationen von k Farben an n Steckplätzen, die einerseits die Problemstellung P und andererseits die Hypothesenstellung H darstellt. Dabei soll die Hypothesenstellung H des Spielers 2 an möglichst vielen der mit der Problemstellung korrespondierenden n Steckplätzen in der Farbe übereinstimmen. Die möglichen Spielsituationen können leicht in einer Theorie T axiomatisiert werden, so daß jedes Modell für T eine Spielsituation darstellt. Ein Spiel also ist eine endliche Folge von Spielsituationen mit gleichbleibendem P und Formeln F, so daß die letzte Spielsituation eine Lösung darstellt, d.h. Hypothesenstellung und Problemstellung stimmen bezüglich der Farbfolgen auf den Steckplätzen überein. Die Formeln F sind nun Informationen an Spieler 2. Für sie kann die logische Sprache um weitere, z.B. abstraktere Prädikate als Farbe und Platz angereichert werden. Auch kann der Informationsgehalt je nach der natürlichsprachlichen Umsetzung der Formeln weiter unterschieden werden. Ausgehend von der angegebenen Formalisierung des Spiels können drei Informationsmaße beschrieben werden, die auf unterschiedlichen Ebenen Informationsmengen beschreiben:

a) auf den Zeichen des Textes, die Spieler 2 als Informationen erhält, ist das Shan-

nonsche Maß der Informationstheorie definiert, das anhand der Häufigkeit für das Auftreten der einzelnen Zeichen und deren Wahrscheinlichkeitsverteilung berechnet wird. Dadurch ist ein Inforamtionsmaß gegeben, das sich rein auf die physiologische Ebene bezieht

b) Betrachtet man die Informationen als Formeln unserer Logik, so kann man ausgehend von der Struktur der Formeln ein weiteres Informationsmaß beschreiben. Die Formeln werden aus atomaren Formeln mit bestimmten, festzulegenden Informationsgrundwerten durch Konnektive aufgebaut, und der Informationswert zusammengesetzter Formeln wird rekursiv über diesen Aufbau definiert. Dabei bewirken die Konnektive unterschiedliche Maßveränderungen. Das Vorwissen des Spielers 2 in einer Spielsituation i größer als 1 geht durch die Konjunktion alter Formelangebote auf alten Spielsituationen mit der neuen Formel auf der aktuellen Spielsituation ein. Der Informationsgehalt der neuen Formel verringert sich z.B. in Abhängigkeit von bereits bekanntem Wissen, erhöht sich in Abhängigkeit von Kombinationsmöglichkeiten mit bekanntem Wissen zu neuem Wissen

c) Zudem ist durch die beschriebene Modellierung implizit ein semantisches Informationsmaß gegeben, nämlich die Anzahl der Farbkombinationen, die durch eine Formel in einer Spielsituation ausgeschlossen werden.

Vorteile dieses formalen Ansatzes sind:
1. Problem und Hypothese sind in einem Modell unterzubringen;
2. Der Problemlösevorgang eines Menschen wird sich bis auf die Reihenfolge des Durchsuchens des Suchraumes nicht von einer automatisierten Problemlösung unterscheiden, so daß die Güte des Problemlösevorgangs nicht nur durch den empirischen Versuch, sondern auch durch den Rechner objektiviert werden kann;
3. Die beschriebene logische Sprache ist einfach, daher bleibt auch der Suchraum einfach;
4. Es ist möglich, die verschiedenen, oben angedeuteten Informationsbegriffe unterzubringen und zu verwenden, so daß sowohl die Modellierung als auch die empirische Untersuchung einen Vergleich der drei Informationsbegriffe erlaubt.

In einer empirischen Untersuchung sollen die Auswirkungen des Informationsgehalts der Hilfestellungen und der Problemkomplexität auf Problemlöseleistung und Streß erhoben werden. Vorteil der Wahl des Spiels Master Mind: Es erlaubt den Einsatz computerunerfahrener Probanden, die den Lernvorgang während des Versuchs absolvieren, d.h. Vorwissen außerhalb des Spiels oder unterschiedliche Lernfähigkeit von Probanden stören den Versuch wenig. Die Untersuchung gibt Hinweise für eine dem menschlichen Informationsverarbeitungsprozeß adäquate Informationsdarbietung, und damit Gestaltungshilfen für sozialverträgliche Technikgestaltung.

Bericht der Rechnerbetriebsgruppe

1 Personal

Die folgenden Personen sind mit den Aufgaben der Rechnerbetriebsgruppe betraut:

- Leiter:

  Prof. Dr. J. Merkwitz

- Wissenschaftliche Mitarbeiter:

  Dipl.–Inform. M. Bender
  Dipl.–Inform. G. Bunsen

- Mathematisch-Technische Assistentinnen:

  M. Brandt-Röhrig
  S. Höckelmann

- Mathematisch-Technische Assistenten zur Ausbildung:

  U. Kämmerling
  G. Pohl

2 Aktivitäten des RBI

In erster Linie ist der Rechnerbetrieb Informatik für die Verfügbarkeit und Funktionsfähigkeit der vorhandenen Ausstattung zuständig. Dazu gehört z.B. die Administration der Benutzerkennungen, Beschaffung und Installation von Software und Hardware, regelmäßige Durchführung von Datensicherungen und auch die Beratung der Benutzer in Fragen der Rechneranwendung und Bedienung. Um einen Eindruck vom Umfang dieser Tätigkeiten zu vermitteln, sei hier nur erwähnt, daß zur Zeit mehr als 600 Benutzerkennungen vom Rechnerbetrieb Informatik verwaltet werden, und daß im Berichtszeitraum wöchentlich mehr als 11 Gigabyte auf 16 Rechnern an fünf Lehrstühlen und Lehrgebieten gesichert wurden.

Zur Betreuung der Studenten an den Terminals und Rechnern besteht ein Beratungsdienst. Zu den Beratungszeiten (38 Stunden pro Woche) ist eine studentische Hilfskraft anwesend, die hilfesuchenden Benutzern zur Beantwortung von Fragen zur Verfügung steht.

Soweit es die bereits erwähnten Arbeiten zuließen arbeiteten sich die Mitarbeiter des Rechnerbetrieb Informatiks in neue Themenbereiche ein. So wurde zum Beispiel probehalber eine Novellbridge installiert, um die Novellserver in anderen Bereichen des FDDI-Netzwerkes erreichen zu können. Weiterhin fand die vorläufige Inbetriebnahme eines X.500 Directoryservices statt. Darüber hinaus konnten erste Erfahrungen mit dem Einsatz des SNMP[2] Standards zur Netzwerküberwachung gesammelt werden.

Aufgrund des zunehmenden Bekanntheitsgrades des Rechnerbetrieb Informatiks in der Hochschule wenden sich immer mehr Hilfesuchende an die Mitarbeiter des Rechnerbetrieb Informatiks. Die Fragen betreffen überwiegend Vernetzung, Vernetzungskonzepte und Installation von Workstations. Dabei ist ein Einsatz der RBI-Mitglieder außerhalb des Hauses durchaus keine Seltenheit, obwohl die Mehrzahl aller Fragen entweder telefonisch oder in einem persönlichen Beratungsgespräch geklärt werden können.

Herr Bender hat im Frühjahr für Angestellte an der RWTH Aachen einen einwöchigen Einführungskurs in das Textverarbeitungssystem MS-Word gehalten.

Frau Höckelmann hat das Rektorat bei der Erstellung eines Programms zum Drucken von Doktorurkunden maßgeblich unterstützt.

3 Ausstattung

3.1 Informatik Forschungsrechner

Seit November 1986 betreibt der Rechnerbetrieb Informatik 3 Rechner vom Typ Celerity 1260D. Die genaue Art der Nutzung dieser Rechner ist den Berichten der einzelnen Lehreinheiten zu entnehmen.

Die Celerities sind wie folgt konfiguriert:

- Plattenlaufwerke mit 1,8 Gigabyte
- 3 Magnetbandlaufwerke
- 28 Megabyte Hauptspeicher
- Ethernet-Controller
- V24-Controller mit insgesamt 40 Schnittstellen
- 1 hochauflösendes Graphikterminal mit einer Auflösung von $1280 * 1024$ Pixeln und 6 Bit Farbinformation pro Pixel.

[2] Simple Network Management Protocol

Die folgende Software steht zur Verfügung:

- C
- C++
- Pascal
- Fortran 77
- Interface Prolog
- FranzLisp
- Datenbanksystem INFORMIX

3.2 CIP-Pool

Der vom Rechnerbetrieb Informatik betreute CIP-Pool wurde im November 1990 erweitert. Er besteht jetzt aus 16 Sun-Workstations und 11 IBM-AT Arbeitsplätzen. Weitere 12 IBM-AT Arbeitsplätze sind bestellt und stehen kurz vor der Inbetriebnahme. Diese Arbeitsplätze wurden im Rahmen des CIP-Programms speziell für die Ausbildung beschafft. Die folgenden Softwareprodukte sind auf dem CIP-Pool installiert:

- C
- C++
- Pascal
- Modula
- Prolog
- Ada
- Fortran
- Miranda

3.3 Lokales Netzwerk

Der Rechnerbetrieb Informatik betreibt das in dem Gebäudekomplex Ahornstraße installierte Ethernet. Es besteht aus einem den gesamten Gebäudekomplex durchlaufenden Backbonesegment und 15 davon abzweigenden Nebensegmenten bzw. Subnetzen. Diese Nebensegmente sind über MAC-Level-Bridges bzw. Gateways an das Backbonesegment angekoppelt. Insgesamt sind deutlich mehr als 200 Endgeräte an das Netz angeschlossen. Seit Ende September hat das LAN den Zugang zum FDDI Hochgeschwindigkeitsnetz der RWTH Aachen.

Der Betrieb eines netzweit zugänglichen Archivservers erlaubt es Benutzern, auf eine Sammlung von etwa 500 Megabyte an frei verteilbarer Software und Daten zuzugreifen. Dazu gehören zum Beispiel Dokumentationen, Compiler, Netzwerksoftware, Textverarbeitungssysteme, Spiele, Grafiksoftware und die *X*-Window Sourcen.

Bericht der Informatik-Bibliothek

Öffnungszeiten:	montags bis donnerstags:	9.00 - 15.45 Uhr
	freitags:	9.00 - 13.00 Uhr
Auskunft:	Ursula Wachten Susanne Neunkirch Christian Krausch	
Telefon:	0241/80-21025	

Personalsituation der Bibliothek

Ab 01.01.91 wird der Bibliothek eine zweite Halbtagsstelle zur Verfügung stehen, die mit der Diplom-Bibliothekarin Renate Eschenbach besetzt werden wird. Christine Bücher-Vogler befindet sich noch bis zum 15.04.91 in Erziehungsurlaub. Die Bibliotheksverwaltung hat für den gesamten Zeitraum Ursula Wachten übernommen. Als HiWi's unterstützten sie Susanne Neunkirch und seit dem 01.06.90 Christian Krausch.

Änderung der Öffnungszeiten

Zu Beginn des WS 90/91 konnten die Öffnungszeiten auf zwei Nachmittage erweitert werden. Ab 1991 wird es durch die zusätzliche Halbtagsstelle möglich sein, an 4 Nachmittagen pro Woche zu öffnen.

Sonderetat für Bücherbeschaffungen

Anfang November erhielt die Bibliothek eine Sonderzuweisung von ca. 30.000 DM, die folgendermaßen verwendet wurden:

- Neuanschaffung der Zeitschriften
  - ACM. Transactions on Modelling and Computer Simulation
  - ACM. Transactions on software engineering and methodology
  - IEEE. Transactions on parallel and distributed systems
  - Space communication
  - International Journal of satellite communications
    (die beiden letzten nur vorübergehend)
- Nachbestellung der Bücher, die Professor Walter aus Drittmitteln finanziert und bei seinem Stellenwechsel mitgenommen hat
- Bestellung von Büchern auf Anregung des neuen Mitglieds des Professorenkollegiums, Professor B. Steffen
- Bestellung von Büchern, die in den einzelnen Lehrstühlen gewünscht wurden

Besondere Veranstaltungen

Workshop on the Integration of Functional and Logic Programming, Granada / Spanien

Vom 24. bis 28. September 1990 veranstaltete der Lehrstuhl für Informatik II der RWTH Aachen unter der Leitung von Prof. Dr. K. Indermark und die Forschungsgruppe der Universidad Complutense de Madrid unter der Leitung von Prof. M. Rodríguez Artalejo im Rahmen der "Acción Integrada" in Granada einen Workshop, an dem außer den beiden Arbeitsgruppen noch weitere Gäste aus Spanien und Deutschland teilnahmen.

Um einen Überblick über diesen Workshop zu vermitteln, sind neben dem Programm auch noch die Kurzfassungen der Vorträge in diesem Abschnitt enthalten.

Programm

- M. Alpuente Fresnedo (Valencia, Spain):
  The Logical + Algebraic EUROPA Environment and its Application to the Rapid Prototyping of Database Applications.
- W. Damm, T. Peikenkamp (Oldenburg, Germany):
  La Flor: Functional and Logic Language on a RISC.
- H. Fassbender, H. Vogler (Aachen, Germany):
  A Narrowing Machine for the Functional Logic Language BABEL, Restricted to Primitive Recursion.
- M. I. Fernández Camacho, D. de Frutos Escrig (Madrid, Spain):
  On Narrowing Strategies for Partial Non-Strict Functions.
- A. Goerdt (Duisburg, Germany):
  Unrestricted Resolution versus N-Resolution.
- J. C. González Moreno, M. T. Hortalá González, M. Rodríguez Artalejo (Madrid, Spain):
  Semantic Foundations of the Higher Order Functional Logic Language BABEL.

- G. Hogen (Aachen, Germany):

  Extension of the Parallel Abstract Machine (PAM) with Evaluators (diploma thesis).

- H. Kuchen, W. Hans (Aachen, Germany):

  An AND-Parallel Implementation of the Functional Logic Language BABEL.

- F. Liu (Oldenburg, Germany):

  Narrowing Mechanism in Functional Logic Models.

- R. Loogen (Aachen, Germany):

  From Reduction Machines to Narrowing Machines.

- F. López Fraguas (Madrid, Spain):

  Building a BABEL Compiler with Logic Programming Techniques.

- J. J. Moreno Navarro (Madrid, Spain):

  State of an Eager Implementation of BABEL.

- J. J. Moreno Navarro (Madrid, Spain), H. Kuchen, R. Loogen (Aachen, Germany), M. Rodríguez Artalejo (Madrid, Spain):

  A Lazy Graph Narrowing Machine for BABEL.

- M. Rodríguez Artalejo (Madrid, Spain):

  The Functional Logic Language BABEL.

- T. Peikenkamp (Oldenburg, Germany):

  A Denotational Semantics for K-LEAF.

- M. Sassin (Aachen, Germany):

  Design of an Abstract Shared Memory Machine for AND-Parallel BABEL with Dependency Information (diploma thesis).

- B. Steffen (Aachen, Germany):

  Data Flow Analysis as Model Checking.

- W. Thomas (Kiel, Germany), M. Rodríguez Artalejo (Madrid, Spain):

  Using Regular Tree Languages for Typing Logic Programs.

- St. Winkler (Aachen, Germany):

  Implementation of a BABEL System in Miranda (diploma thesis).

Kurzfassung der Vorträge

The logical + algebraic EUROPA environment and its application to the rapid prototyping of database applications

M. Alpuente Fresnedo (Universidad Politécnia de Valencia, Spain)

The aim of this paper is twofold. First, we try to present logic+equational specification techniques as an effective tool for the rapid prototyping of database applications. Second, we give a sketch, using the logic+equational[1] experimental language EUROPA, of an integrated environment for executable specifications, able to operate with running specifications both in parsing and in generating mode and whose development is in progress by our working group. The language takes a discipline of constructors that makes feasible the use of Flat-SLD-resolution as the basis for the development of an efficient interpreter. It differs from other similar languages because it is able to cope with some suitable kind of equations between constructors, fitting for database applications. The integrated environment is built on top of a standard Prolog.

[1]Some sources call 'functional languages' to the class of equational or term-rewriting languages that distinguish between functions and data constructors. Although this terminology is widespread, we will not do it explicit in this paper.

La Flor: Functional and Logic Language on a RISC

W. Damm, T. Peikenkamp (Universität Oldenburg, Germany)

We present the design of a RISC architecture for a logic + functional language K-LEAF. This language as well as an abstract machine K-WAM supporting lazy narrowing were developped within ESPRIT project 415 D at the University of Pisa under Giorgio Levi and at CSELT in Torino by Pier Giorgio Bosco, Carlo Cecchi, and Corrado Moiso. This project resulted in an OR parallel implementation on a 64 node transputer system using a custom designed delta-network.

While this project did not address dedicated hardware support for the lanaguage K-Leaf, the aim of this paper is to investigate the degree of improvement achievable in execution-time by including a few dedicated extensions in a standard RISC type processor. Based on an analysis of a set of benchmarks on an instrumented K-WAM simulator, the following features were selected as relevant add-ons:

- a top-of choice point cache: already a one-entry cache realizes (using copy-back and a demand driven strategy in reloading an invalidated top choice-point) a hit ratio well above 80%
- a two-level tagging scheme supported by fast multiway branching;
- a built-in stack to support unification and evalutation of built-in predicates and functions.

The design of an instruction-set incorporating theses features was carried out in parallel with the development of a compiler expanding K-WAM instructions into machine code of the evolving instruction set. It was validated and tuned in several design cycles using a simulator at instruction-set-level, execution the translation of the benchmark suit.

The RT-design is centered around a dual port register-bank with two sets of 64 registers, allowing to implement reinstalling registers out of the choice-point and loading the choice point by a simple switch of the current register set.

In addition it contains an eight entry deep built-in stack used during unification and evaluation of built-in predicates. The data-path is split into three pipeline stages. As of today, only a preliminary design has been completed. First evaluations suggest major changes in the implementation of basic routines for dereferencing and forcing evaluation of suspended functions in order to obtain a better exploitation of the pipeline. Further work will also consider extensions needed to support a parallel execution of K-LEAF.

A Narrowing Machine for the Functional Logic Language BABEL, Restricted to Primitive Recursion

H. Fassbender, H. Vogler (RWTH Aachen, Germany)

We investigate the implementation of a particular subset of the functional logic programming language BABEL; we call this language *syntax directed BABEL* (for short: sd-BABEL). Programs written in sd-BABEL allow the definition of functions over some freely generated datatype by means of primitive recursion schemes. Computations of sd-BABEL are narrowing sequences that solve goals of the form $equal(\xi, t)$, where equal is the usual equality predicate, ξ is an expression containing functions, logical variables, and constructors, and t is an expression containing constructors and logical variables only.

On an informal level we discuss a machine that performs narrowing sequences for sd-BABEL programs. This narrowing machine is based on a machine called checking-tree nested-stack transducer (for short: ct-ns transducer), where the latter machine can reduce expressions ξ containing functions and constructors only, to normal form. In other words, ct-ns transducers can solve goals of the form $equal(\xi, z)$, where z is a logical variable. Even, this machine characterizes the narrowing sequences of such particular goals and, due to this 'minimally', the presented machine might give a contribution to the efficient implementation of sd-BABEL.

On Narrowing Strategies for Partial Non-Strict Functions

M. I. Fernández Camacho, D. de Frutos Escrig (Universidad Complutense Madrid, Spain)

We study completeness of narrowing strategies for a class of programs defining (possible partial and non-strict) functions by means of equations, with a lazy semantics, so that infinite values are also admissible. We consider a syntactical restriction introduced by Echahed, under which he proved that any narrowing strategy is complete for specifications defining total functions with finite values. Unfortunately things are not so pretty for the larger class of programs that we consider. So we see that laziness of strategies is necessary in order to cope with non strictness, fairness if we want to compute infinite values, and syntactially complete specifications (those including rules covering all possible patterns for each function) if we are also interested in the computation of partial values.

Unrestricted Resolution versus N-Resolution

A. Goerdt (Universität -GH- Duisburg, Germany)

An N-resolution proof is a resolution proof in which the resolution rule is restricted: One of the clauses to which it is applied must only consist of negative literals. N-resolution is one of several resolution restrictions. It is complete.

We present an infinite family of unsatisfiable propositional formulas and show: These formulas have unrestricted resolution proofs whose length is polynomially bounded in their size. The N-resolution proofs of these formulas are shown to be not polynomially bounded. *N-resolution does not p-simulate unrestricted resolution.*

Semantic Foundations of the Higher Order Functional Logic Language BABEL

J. C. González Moreno, (Universidad Politécnica de Madrid, Spain), M. T. Hortalá González, M. Rodríguez Artalejo (Universidad Complutense de Madrid, Spain)

We describe an extension of the first order functional logic language BABEL which supports higher order programming, including higher order logic variables. The operational semantics is based on lazy narrowing and subsumes reduction and SLD-resolution. General higher order unification is avoided and decidable unification is used instead. By means of a syntactical translation, we relate the operational semantics to the declarative semantics of the first order fragment of the language. This provides soundness and completeness theorems which give an exact characterization of the expected solutions on the higher order level. The class of potential solutions is program dependent. For some programs, it is as powerful as the class of λ-expressions.

This paper is part of a research project which includes the design of abstract machines for implementing BABEL.

Extension of the Parallel Abstract Machine (PAM) with Evaluators

G. Hogen (RWTH Aachen, Germany)

The PAM is a parallel graph reduction machine for functional programs. It has been implemented in OCCAM on a transputer system. A functional program has to be transformed into a parallel functional program, in which all subexpressions which can be evaluated in parallel, are marked by a special syntactic construct. This parallel program is then compiled into code for the parallel machine.

Using lazy evaluation we are able to treat infinite data structures. Lazy evaluation also avoids unnecessary computations. To evaluate subexpressions, we have to ensure, that we do not initiate a computation, unless its non-termination would imply that the semantics of the overall expression was bottom. To transform a lazy functional program into a parallel version we need strictness information about functions, because it is possible to evaluate all arguments in strict parameter positions of a function application in parallel.

Traditionally strictness analysis is done by abstract interpretation of the program over the abstract domain $\underline{2}$:={0,1}, where 0 represents the undefined value and 1 represents any value. G.Burn [Burn 87] considers a more detailed abstract domain

for algebraic data structures and takes into account contextual information. So we get more detailed strictness information, which can be used to parallelize a program, but also can be used to optimize the program execution.

The aim of this paper is to give a short introduction into the evaluator technique of [Burn 87], the incoperation of this technique in the PAM and the extension of a compiler backend and the implementation of the PAM by evaluators.

An AND-Parallel Implementation of the Functional Logic Language BABEL

H. Kuchen, W. Hans (RWTH Aachen, Germany)

We desribe an AND-parallel implementation of the functional logic programming language BABEL. This implementation is based on the abstract machine ABAM which serves as a model for the BABEL run-time system on a multicomputer, i. e. a system, where all processors have their own local memory and communicate by exchanging messages. The ABAM can also be used for AND parallel implementations of other functional logic or purely logic languages such as (pure) PROLOG. More parallelism can be exploited than with the well-known restricted AND-parallelism.

Narrowing Mechanism in Functional Logic Models

F. Liu (Universität Oldenburg, Germany)

As the operational mechanism of functional logic models, narrowing possesses the capability of reducing and solving programs. In this paper, we first examine the natural relationship and uniformity between reduction and narrowing and their operational behaviour in functional logic models, and then demonstrate the importance of taking reduction as an independent evaluator since narrowing, in general, is considered less efficient than reduction. Thereafter, we suggest a general evaluation scheme for functional logic models in which reduction is used whenever possible, and narrowing is demanded when it is absolutely necessary, in particular, when it is required to solve. Accordingly, we propose a lazy narrowing strategy called Weak Head Normal Formal Normal Order Narrowing strategy which has many advantages over other narrowing strategies.

From Reduction Machines To Narrowing Machines

R. Loogen (RWTH Aachen, Germany)

Narrowing, the evaluation mechanism of functional logic languages, can be seen as a generalization of *reduction*, the evaluation mechanism of purely functional languages. The unidirectional pattern matching, which is used for parameter passing in functional languages, is simply replaced by the bidirectional *unification* known from logic programming languages. We show in this paper, how to extend a reduction machine, that has been designed for the evaluation of purely functional programs to a machine that performs narrowing. The necessary extensions concern the realization of unification and backtracking. The latter has to be incorporated to handle non-deterministic computations. It turns out that the resulting narrowing machine can also be seen as an extension of Warren's Prolog engine [Warren 83]. This extension enables a space efficient handling of nested expressions and embodies an optimized treatment of deterministic computations. As in Warren's machine the central component of the machine is a stack that contains environments, i.e. activation records of function calls, and choice points to keep track of possible alternative computations. It is ensured that choice points always contain the minimal amount of information that is necessary to restore a previous state on backtracking. A complete specification of the machine and of the translation of a sample language into abstract machine code is given. To test the feasibility of the new implementation technique a preliminary implementation has been developed in Miranda.

Building a BABEL Compiler with Logic Programming Techniques

F. López Fraguas (Universidad Politécnica de Madrid, Spain)

We describe the development of a real compiler written in Prolog for the functional-logic language BABEL. We have based our work on the usual approach to compiler writing in Prolog that can be found in the related literature. The most outstanding characteristic of this approach is the use of a uniform tool, – Definite Clause Grammars (DCG's) –, for all the phases of a compiler: lexical analysis, parsing and code generating. But a 'real' compiler in Prolog presents some problems that are frequently underestimated, mainly: memory overflow and error recovery. We examine how any solution to these affects to the declarative style of Prolog programming , and describe how we have enfaced them in our compiler. Special emphasis is made in the enhancement of DCG's we have designed and implemented for the error handling problem.

A Denotational Semantics for K-LEAF

T. Peikenkamp (Universität Oldenburg, Germany)

Implementations of logic languages usually lack completeness (w.r.t. the declarative semantics) due to efficiency reasons. This is also true for the language K-LEAF and its implementation. Its original operational semantics does not explain why the implementation in general yields some, but not all of the "logic" solutions. To overcome this disadvantage we present a denotational semantics for the language K-LEAF, which correctly describes the behaviour of the implementation we are thinking of. The last restriction is important since we do not(!) describe the original K-LEAF implementation developed in the ESPRIT project 415 D, but an implementation, we plan to implement. The difference stems from the fact that the original implementation tries to perform all possible syntactic unification steps before using user-defined equations. This deviation from the original operational behaviour makes the force list, a data structure used to store suspended computations, superfluous and therefore allows for a simpler implementation.

Design of an Abstract Shared Memory Machine for AND-Parallel BABEL with Dependency Information

M. Sassin (RWTH Aachen, Germany)

This paper gives the reader a short impression of an AND-parallel abstract machine for the functional logic language BABEL. The aim is to enable an abstract BABEL machine to evaluate as much subexpressions as possible in parallel.
In contrast to expressions of pure functional languages expressions of logic or functional logic languages contain free variables that can be bound to terms during the execution of expressions. Suppose X is a free variable of the expression E and E_1 and E_2 are two subexpressions of E that contain the variable X. If the variable X is not fully instantiated, parallel execution of the subexpressions E_1 and E_2 can cause two different bindings for the variable X. The same problem can occur if E_1 and E_2 contain shared variables.
To avoid communication and synchronisation overhead it is sensible to forbid to evaluate such expressions in parallel.
An abstract PROLOG machine checks during runtime if parallel executions can be done without conflicts. The parallel execution of subexpressions is admitted only in the cases in which a runtime-test yields that

- all common variables of the subexpressions are fully instantiated and

- all uninstantiated variables of a subexpression are not shared with uninstantiated variables of other subexpressions.

The advantage of this method is the high degree of parallelism that can be reached during execution. The disadvantage is the great overhead that is caused by runtime-tests.
The overhead during runtime can only be minimalized if during compile time it can be decided if a variable is fully instantiated or if variables are shared. We can get this information by *abstract interpretation* of the program. [Hwang] describes a compile time approach for exploiting and-parallelism in a PROLOG program with the help of an abstract interpretation.
We took up the idea of abstract interpretation to detect if parallel execution of BABEL expressions can be done without conflicts.

Data Flow Analysis as Model Checking

B. Steffen (RWTH Aachen, Germany)

The paper develops a framework that is based upon the idea that model logic provides an appropriate framework for the specification of data flow analysis (DFA) algorithms as soon as programs are represented as models of the logic. This can be exploited to construct a DFA-*generator* that generates efficient implementations of DFA-algorithms from modal specifications by partially evaluating a specific model checker wrt specifying modal formula. Moreover, the use of a modal logic as specification language for DFA-algorithms supports the compositional development of specifications and structured proofs of properties of DFA-algorithms. The framework is illustrated by means of a real life example: the problem of determing optimal computation points within the flow graphs.

Using Regular Tree Languages for Typing Logic Programs

W. Thomas (Universität Kiel, Germany), M. Rodríguez Artalejo
(Universidad Complutense de Madrid, Spain)

We adopt here the view that typing a logic program P means to derive from P a superset of its semantics. The emptiness of this superset would indicate inconsistencies or mistakes in the program. We present a procedure which extracts from a pure logic program such types (for all the arguments of the program predicates) in the form of

regular sets of terms (or trees). For this purpose, a given logic program is transformed into a "regular logic grammar", basically by dropping the dependencies between different arguments in the program predicates. A regular logic grammar is monadic and can be considered as an ***alternating tree automaton***, equipped with existential and universal states. We supply a reduction of these tree automata to nondeterministic ones in order to solve the emptiness problem. The (single exponential) complexity of this emptiness test is shown to be optimal in the general case. Also a refined inference method is developed which yields a better approximation to the program semantics by regular types, using a more "hierarchical" approach to detect program bodies that fail.

Implementation of a BABEL System in Miranda

St.Winkler (RWTH Aachen, Germany)

This paper is an extended abstract of my diploma thesis, which contains a description of an implementation of a stack-based narrowing machine and of a prototype system, which enables Babel programming with higher-order functions, Prolog-like clauses and where boolean and numerical datatypes and some further optimization are supported.

The implementation of the stack-based narrowing machine for Babel has been realized in Miranda on a Sun Workstation. Furthermore a convenient environment for Babel programming has been implemented, which allows to edit, to compile and to run Babel programs.

In this extended abstract we concentrate on the description of the Babel programming environment.

RWTH Aachen
Fachgruppe Informatik

TAG DER INFORMATIK

Freitag, den 30. November 1990
Ahornstr. 55 (ehem. PH), Hörsaal AH V

Programm

9.15 – 9.25 **Eröffnung**

Automatische Analyse
Leitung: K. Indermark

9.30 – 10.00 *B. Steffen, Lehr- und Forschungsgebiet Informatik:*
Reflexion über die automatische Analyse von Systemen.

10.00 – 10.30 *J. Noack, W. Oberschelp*
Lehrstuhl für Angewandte Math. insb. Informatik:
Semantische Analyse für eine natürlichsprachliche Datenbankschnittstelle.

Verteilte Systeme
Leitung: J. Merkwitz

11.00 – 11.30 *D. Meyer-Ebrecht, Lehrstuhl für Meßtechnik:*
Verteilte Bildverarbeitungssysteme für den Routineeinsatz: Architektur von Verbindungsnetzwerken.

11.30 – 12.00 *C. Weise, Lehrstuhl für Informatik I:*
Verifikation periodischer Prozesse in statischen, verteilten Echtzeitsystemen.

12.00 – 12.30 *F. Brand, Lehrstuhl für Betriebssysteme:*
Telefax-Kommunikation über Datennetze.

14.00 – 14.30 *K. Garbe, Technische Universität Dresden:*
Informatik an der TU Dresden: Stand und Prespektiven.

Wissensrepräsentation und Informationssysteme.
Leitung: W. Oberschelp

14.30 – 15.00 *R. Majewski,*
Lehr- und Forschungsgebiet Theoretische Informatik:
Repräsentation von Wissen für analoge Interferenzen.

15.00 – 15.30 *B. Thull, Helmholtz-Institut:*
Ansatz eines Dialogmanagers für ein intensivmedizinisches Informationssystem.

Steuerungssysteme.
Leitung: O. Spaniol

16.00 – 16.30 *W. Seiche, D. Abel, H. Rake,*
Institut für Regelungstechnik:
Steuerungsentwurf auf der Grundlage einer graphentheoretischen Petri-Netz-Analyse.

16.30 – 17.00 *K. Hohenauer, Institut für Kunststoffverarbeitung:*
Konzepte einer objektorientierten Steuerung für Anlagen und Maschinen.

17.15 – 18.15 **Hauptvortrag**
H. Maurer, Technische Universität Graz:
Prognosen über die Zukunft der Informationstechnik sind unmöglich, aber notwendig.

18.15 – 18.45 Verleihung des SUN-Preises
(Übergabe, Kurzreferat, Demonstration)
Stiftung eines *Algorithmik–Preis*es
durch die Firma GEI.
Schlußwort.

ab 18.45 *Buffet*

Kurzfassungen der Vorträge

Reflexion über die automatische Analyse von Systemen

B. Steffen (Lehr- und Forschungsgebiet Informatik)

Die Analyse physikalischer, chemischer oder sonstiger von der Außenwelt vorgegebener Systeme hat zum Ziel, Gesetzmäßigkeiten (Eigenschaften) offenzulegen, die den zukünftigen Umgang mit solchen Systemen erleichtern sollen. Gewonnene Erkenntnisse gelten dann solange, wie keine Widersprüche bei weiteren Beobachtungen auftreten. Korrektheitsbeweise, im mathematischen Sinne, können hier nicht geführt werden.

Anders als die oben genannten Systeme sind Softwaresysteme mathematische Objekte. Das ermöglicht, zumindest in gewissem Rahmen, mathematische Korrektheitsbeweise. Aber auch hier scheitern automatische Analysemethoden in der Praxis (üblicherweise aus Komplexitätsgründen).

Softwaresysteme werden von Menschen konstruiert. Das erlaubt, den Konstruktionsprozeß in Hinblick auf eine spätere Analyse auszurichten. Zentral ist hier die Forderung nach Kompositionalität: Softwaresysteme sollten so strukturiert sein, daß sich die Eigenschaften komplexer Systeme aus den Eigenschaften ihrer unmittelbaren Teilsysteme ableiten lassen. Auf dieser Weise lassen sich sich dann auch praxisrelevante Systeme automatisch analysieren.

Kompositionalität hat ihren Preis. Der saubere baukastenartige und transparente Aufbau eines Systems führt üblicherweise zu Ineffizienzen, sowohl im Laufzeitverhalten, als auch in der Codelänge. Das ist sicher auch ein wesentlicher Grund dafür, daß sich ein solcher Aufbau in der Praxis noch nicht durchgesetzt hat.

Softwaresysteme können also “analysefreundlich” konstruiert werden. Der Preis für eine solche Konstruktion zahlt sich langfristig, besonders bei sicherheitskritischen Systemen, leicht aus. Nur eine saubere, wohlstrukturierte Konstruktion führt zu zuverlässigen, automatisch analysierbaren und gut erweiterbaren Softwaresystemen.

Semantische Analyse für eine natürlichsprachliche Datenbank-Schnittstelle

J. Noack, W. Oberschelp (Lehrstuhl für angewandte Mathematik insbesondere Informatik)

In vielen Arbeiten über natürlichsprachliche Schnittstellen zu Datenbanken (Pereira 82, Grosz et al. 87, Heinz & Matiasek 89) werden Fragen zunächst in eine prädikatenlogische Form überführt, die unabhängig vom zugrundeliegenden Datenbankschema ist. Danach erst erfolgt eine Abbildung in Datenbank-Prädikate. Obwohl richtig erkannt wurde (Heinz & Matiasek 89), daß solche Abbildungen nicht isomorph und damit nicht trivial sind, hat dieses Problem im Bereich der natürlichsprachlichen Anfragesysteme bisher wenig Beachtung gefunden. Wir skizzieren zuerst die übliche Vorgehensweise und zeigen dann anhand einiger Beispiele verschiedene Problempunkte auf.

Einen theoretischen Hintergrund für die Verwendung des kompositionellen Ansatzes bei Datenbankfragen liefern Arbeiten über Generalisierte Quantoren (Mostowski 57, Barwise & Cooper 81, Keenan & Stavi 86), deren grundlegende Idee darin besteht, daß der eigentliche Quantor nicht allein aus dem Determinator, sondern aus der vollständigen Nominalphrase (NP) gebildet wird. Mit Hilfe der Typenlogik (Montague 74) lassen sich Sätze der natürlichen Sprache in die Formelsprache der Generalisierten Quantoren übertragen.

Durch eine graphentheoretisch motivierte Diskursmodellierung gelingt es, das durch die natürlichsprachlichen Anfragen induzierte Abbildungsproblem als NP-vollständig nachzuweisen. Mit einem heuristischen Algorithmus, der das Gerüst der formalen Anfrage festlegt, und einem regelbasierten Verfahren, das den Skopus der quantifizierenden Informationen bestimmt, können für viele relevante Situationen adäquate Übersetzungen gefunden werden.

Verteilte Bildverarbeitungssysteme für den Routineeinsatz: Architektur von Verbindungsnetzwerken

D. Meyer-Ebrecht (Lehrstuhl für Meßtechnik)

PACS ist das Kürzel für ***Picture Archiving and Communication Systems*** für den Einsatz in der medizinischen Diagnostik. Hinter diesem Kürzel steht der Plan, die Filmfolie als Träger diagnostischer Bildinformation durch elektronische Systeme abzulösen, in denen Bilder durchgängig digital transportiert, gespeichert und verarbeitet werden. Die Konzeption von PACS-Systemen muß von Leistungseckwerten ausgehen, die um Größenordnungen über denen herkömmlicher verteilter Datenverarbeitungssysteme liegen: Bilder ergeben Datensätze von der Größenordnung 10 MByte. Die gesamte Bildproduktion einer einzigen Klinik – um 300.000 Röntgenbilder und 300.000 primär digitale Bilder (CT, MR, DSA etc.) pro Jahr – führt zu Datenvolumina in der Größenordnung TeraByte.

Lokale Netze, die Bildquellen mit Bildarbeitsplatzsystemen, Bildarchiven etc. vernetzen sollen, müssen einen Nettodatendurchsatz von mehreren hundert Mbit/s aufbringen, damit das für die Arbeitsroutine benötigte Bildmaterial arbeitsgerecht zügig gehandhabt werden kann. Neue konzeptionelle Ansätze sind nötig, um die erforderliche Leistung mit der verfügbaren Technologie zu erreichen.

ImNet*/2**, ein neuer Typ eines lokalen Netzwerks, verwendet als Übertragungsmedium Lichtleiter. Eine Mehrfach-Stern-Topologie verbunden mit dem ***line-switch-Prinzip ermöglicht eine optimale Anpassung an die Übertragungsanforderungen in einer gegebenen PACS-Umgebung. Ein inhärent kollisionsfreies Vermittlungskonzept kann konkurrierende Übertragungswünsche konfliktlos befriedigen. Eine wortweise asynchrone Übertragung ermöglicht eine transparente Verbindung zwischen den vermittelten Partnern, so daß die Bilddatenübertragung nicht mehr durch weitere Protokollmaßnahmen belastet wird. Bei einer Übertragungsfrequenz von 175 Mbit/s (physikalische Rate) wird deshalb eine Transportrate von 140 MByte/s gleichzeitig in allen nicht mehrfach belegten Zweigen des Netzwerkes erreicht. Die Protokollebenen 5 bis 7 wurden nach dem Standardisierungsvorschlag *ACR/NEMA* implementiert.

An der RWTH wird bereits seit längerem ein Pilotnetzwerk des Vorläufers ***ImNet*/1** (physikalische Rate 16 Mbit/s) mit einer Ausdehnung von über 2 km betrieben. Es verbindet mehrere bilderzeugende Modalitäten im Klinikum Aachen (u.a Kernspin-Tomographie, digitale Röntgensysteme und Laser-Augenhintergrund-Camera) mit Bildverarbeitungs- und Bildspeichersystemen im Gebäude der Elektrotechnik. Seine Umstellung auf 175 Mbit/s ist in Arbeit.

Verifikation periodischer Prozesse in verteilten, statischen Echtzeitsystemen

Carsten Weise (Informatik I)

Von einer Echtzeitanwendung spricht man, wenn durch die Umgebung eines Rechnernetzwerkes bestimmte zeitliche Anforderungen an die Prozesse des Systems gestellt werden. Sind dabei alle Prozesse a priori bekannt, spricht man von einem statischen System.

In vielen Anwendungen müssen die vorgegebenen Zeitschranken unter allen Umständen eingehalten werden. Bei solchen Anwendungen kann man sich nicht allein auf die Intuition der Programmierer verlassen. Eine Verifikation der Implementierung ist unumgänglich. Da dies eine schwierige und fehleranfällige Aufgabe ist, gilt der automatischen Verifikation solcher Systeme ein besonderes Interesse. Dazu benötigt man eine geeignete Modellierung des Systems.

Obwohl man im allgemeinen bei Echtzeitsystemen zwischen periodischen und aperiodischen Prozessen unterscheidet, werden im vorgestellten Ansatz alle Prozesse als periodische Prozesse modelliert. Dies ist dadurch motiviert, daß auch für einen aperiodischen Prozeß immer genügend Rechenkapazität reserviert werden muß, um ihn im Bedarfsfalle auch ausführen zu können. Deshalb kann man einen solchen aperiodischen Prozeß auch als einen periodischen beschreiben, der sich bei jeder Aktivierung entscheidet, ob er seine üblichen Aktionen durchführen muß oder nicht.

Eine synchrone Kommunikation führt dazu, daß auf Nachrichten wartende Prozesse theoretisch beliebig lange verzögert werden können. In diesem Fall ist keine Aussage über Reaktionszeiten der Prozesse möglich. Deshalb wird die Kommunikation im vorgestellten Modell asynchron über Puffer durchgeführt.

Diese Überlegungen führen dazu, Prozesse als eine Reihe von Alternativen zu beschreiben. Ein Prozeß wird innerhalb seiner Periode mindestens einmal geweckt, und entscheidet sich dann aufgrund der in seinem Puffer bereits angekommenen Nachrichten für eine seiner Alternativen. Im Extremfall kann eine solche Alternative leer sein, d.h. der Prozeß tut nichts. Während der Ausführung einer Alternative kann ein Prozeß nicht mehr auf weitere Nachrichten warten. Nur das Senden von Nachrichten ist ihm erlaubt.

Das Verhalten des Gesamtsystems kann dann durch die Menge der „zulässigen Traces" beschrieben werden. Ist die Kapazität der Nachrichtenpuffer endlich, so kann diese Menge der zulässigen Traces durch die Menge der von einem endlichen Automaten erkannten, unendlichen Wörter spezifiziert werden. Der endliche Automat kann direkt aus den Prozeßbeschreibungen konstruiert werden. Ein Schritt dieses Automaten entspricht dabei einer Zeiteinheit des Systems.

Die zeitlichen Anforderungen an das System können z.B. in der Echzeitlogik RTTL (Real Time Temporal Logic) von Alur und Henzinger spezifiziert werden. Die Erfüllbarkeit von Formeln dieser Logik kann mit Hilfe des endlichen Automaten, der die zulässigen Traces des Systems beschreibt, entschieden werden. Damit ist insgesamt eine automatische Verifikation möglich.

Telefax-Kommunikation über Datennetze

F. Brand (Lehrstuhl für Betriebssysteme)

Aufgaben:

A. Fax im Wissenschaftsnetz

Für die Übertragung von Daten innerhalb geschlossener Benutzergruppen existieren oft weiträumige Datenverbindungen. Sie finden vielfach im Datex-P-Dienst der TELEKOM statt. Da viele dieser Dienste (z.B. das die Hochschulen und Großforschungseinrichtungen verbindende Wissenschaftsnetz des DFN-Vereins) pauschal und nicht nach übertragenem Datenvolumen abgerechnet werden, ist es interessant, dieses Netz auch zur Dokumentübertragung zu nutzen. Hierzu müssen Übergänge vom Telefax-Dienst (analog) auf Datennetze (digital) geschaffen werden. Diese Datennetze können, wie das Wissenschaftsnetz, X.25-Netze sein, oder beliebige lokale Datennetze, die ihrerseits an das WIN angeschlossen sind.

B. Inhouse-Fax

Zur Beschleunigung innerbetrieblicher Kommunikationsvorgänge werden heutzutage Telefon-Nebenstellenanlagen und, in zunehmendem Maße, lokale Datennetze verwendet. Da jedoch viele der zu übermittelnden Dokumente in Papierform vorliegen und eine direkte Digitalisierung und Übertragung über die Datennetze an Inkompatibilitäten der verwendeten Rechnersysteme scheitert, ist das Telefax eine interessante, universelle Alternative. Wegen der begrenzten Übertragungskapazität der Telefon-Nebenstellenanlagen (durch abgehende oder ankommende Telefaxe wäre ein Telefonanschluß blockiert) auf der einen und freien Kapazitäten von lokalen Datennetzen auf der anderen Seite erscheint es lohnend, die Möglichkeiten der Telefaxübertragung über lokale Datennetze für den Inhouse-Einsatz zu untersuchen. Hierbei würden entweder Telefaxgeräte über PC's mit Fax- Einsteckkarten an das Datennetz angeschlossen oder PC's mit Scanner und Drucker verwendet, die bereits in bestehende Netze integriert sind. Wegen der weiten Verbreitung von PC's am Arbeitsplatz sollen im folgenden nur PC-Fax-Lösungen betrachtet werden. Ebenfalls verfügbare Fax-Lösungen der UNIX-Welt sind im Preis erheblich höher angesiedelt.

C. Fax-Dienst der TELEKOM

Oft besteht die Notwendigkeit, mit Adressaten im Telefonnetz zu kommunizieren. Zu diesem Zweck ist ein Übergang zum Telefaxdienst der TELEKOM nötig. Hierzu müßte eine für den Nutzer transparente Verbindung des Datennetzes zum Telefaxnetz geschaffen werden.

Aufbau:

- Vom Telefonnetz ist über die Anwahl eines PC's mit PC-Fax- Karte (mit Auswertung der Nachwahl über Vorschaltgerät) der Übergang in ein lokales Datennetz oder in ein X.25-Netz sowie in umgekehrter Richtung möglich.
- Die PC-Fax-Geräte können untereinander über das lokale Datennetz oder über Weitverkehrs-Datennetze Dokumente austauschen.
- Aus technischer Sicht wäre es auch denkbar, direkt im Anschluß an eine Datennetz-Übertragung über weite Strecken den Übergang in ein Telefonnetz bereitzustellen.

Offene Probleme:

Wie können Teilnehmer adressiert werden?

Hierbei soll die Adressierung allgemein zugänglich und im Aufbau verständlich sein. Außerdem sollte sich die Stellenzahl auf etwa 10-30 Stellen (numerisch) beschränken lassen.

Wie wird diese Adressinformation übertragen und dem Teilnehmer zugeordnet?

Erfolgversprechende Ansätze sind die Übertragung der Adresse als Wahlinformation (von jedem Telefaxgerät aus möglich) und die Zuordnung von WIN und Telefon-Adressen verteilt auf mehrere Gateways. Dies wirft jedoch Probleme mit der Verfügbarkeit von Verzeichnisinformationen auf. Außerdem muß nach einem Gerät gesucht werden, das die Wahlinformation von der Telefonleitung aus auf einen PC umsetzen kann.

Wie hoch wird die Belastung von Server-PC's und die eventuelle Wartezeit auf Server-Dienste sein?

Die als Übergänge zwischen Telefonnetz und Datennetz eingesetzten PC's werden von vielen Teilnehmern genutzt werden. Da die Telefax-Karten kein Multitasking gestatten und es auch nicht möglich sein dürfte, einen PC mit mehreren Karten auszustatten, muß für die Umsetzung Telefonnetz - Datennetz ein Mehrfachanschluß mit mehreren gleichartigen Server-PC's vorgesehen werden, um häufige "Besetzt"-Zeiten zu vermeiden. Die Anzahl der notwendigen Server kann nur experimentell ermittelt werden, da die Akzeptanz des Dienstes und die Länge der übertragenen Dokumente schlecht geschätzt werden kann. Dies gilt, in abgeschwächter Form, auch für die Gegenrichtung (nur Überlauf der Warteschlange möglich).

Repräsentation von Wissen für analoge Inferenzen

R. Majewski (Lehr- und Forschungsgebiet Theoretische Informatik)

Ein Teilgebiet des Maschinellen Lernens, der Bereich der analogen Inferenz, befaßt sich mit dem Problem, wie die Ähnlichkeit zweier Wissensbereiche genutzt werden kann, um Wissen von einem zum anderen zu übertragen. Da es sich bei der Bildung einer Analogie im allgemeinen um einen sehr komplexen und wissensintensiven Prozeß handelt, sollte bereits der Wissensrepräsentationsmechanismus die Ähnlichkeit zweier Domänen in syntaktischer Hinsicht berücksichtigen. Dadurch wird es möglich, einen Teil der Analogiebildung auf den Zeitpunkt der Wisseneingabe vorzuverlegen. Außerdem können sonst implizit vorhandene Übereinstimmungen zwischen Domänen zusammen mit dem Wissen explizit in die Wissensbasis eingetragen werden. Dies erleichtert z.B. die Suche nach zwei analogen Domänen erheblich.

Ein möglicher Lösungsansatz für eine Wissensrepräsentation, welche die Bildung von Analogien im Hinblick auf analoge Inferenzen unterstützt, besteht in der Modularisierung des Wissens bzw. dessen Darstellung. Übereinstimmende Teilaspekte von Domänen werden zusammengefaßt und nur einmal in die Datenbasis eingetragen. Dadurch kann der Umfang der Datenbasis reduziert werden. Insbesondere können die zusätzlichen expliziten Angaben sich auf solche Module beziehen, wodurch sie nicht immer wieder neu erzeugt werden müssen.

Der Vortrag stellt einen auf diesem Lösungsansatz basierenden Darstellungsmechanismus vor. Das Wissen wird mittels eines auf der Prädikatenlogik basierenden Formalismus eingegeben, um dann in eine Darstellung auf Systemebene, auf dieser Ebene findet dann auch die Analogiebildung statt, übersetzt zu werden. Die Grundlage der Systemebenendarstellung bildet eine Menge ausgezeichneter Prädikate, die jeweils bestimmte Aspekte der Ausgangsformel charakterisieren. So wird beispielsweise die Menge der Literale der Ausgangsformel explizit in die beiden Mengen der ein- und mehrstelligen Literale eingeteilt.

Ansatz eines Dialogmanagers für ein intensivmedizinisches Informationssystem

B. Thull (Helmholtz-Institut)

Die Gestaltung der Benutzerschnittstelle für ein medizinisches Informationssystem stellt insbesondere an die Auslegung der Dialog- und Interaktionssequenzen hohe Anforderungen. Es wurde ein Ansatz für einen Dialogmanager vorgestellt, der die Definition komplexer Dialoge unterstützt, Mechanismen für eine Benutzerführung auf der Basis einer Farbkodierung anbietet und schließlich eine Analyse von Dialogen an einer Benutzerschnittstelle mit Hilfe graphentheoretischer Algorithmen ermöglicht. Am Beispiel einer Kontrolleinheit für eine Infusionspumpe wurde der beschriebene Ansatz illustriert. Nach den bisherigen Erfahrungen beim Einsatz des Dialogmanagers haben sich vor allem die Möglichkeit zur einfachen Definition komplexer Dialoge und die automatisierte Benutzerführung bewährt. Die Analyse von Dialogen mit Hilfe graphentheoretischer Algorithmen muß für einen praktischen Einsatz bei der Gestaltung von Benutzerschnittstellen noch methodisch weiterentwickelt werden.

Steuerungsentwurf auf der Grundlage einer graphentheoretischen Petri-Netz-Analyse

W. Seiche, D. Abel, H. Rake (Institut für Regelungstechnik)

Systeme, in denen nicht zeitkontinuierliche Abläufe, sondern deren gegenseitige diskrete Kopplungen interessieren – sogenannte ereignisdiskrete Systeme – lassen sich formal durch Petri-Netze beschreiben. Die mathematische Analyse eines Netzes mittels algebraischer und graphentheoretischer Verfahren des modellierten Systems. Offen bleibt jedoch, ob und wie aus den Ergebnissen der Analyse entsprechende Netzmodifikationen abgeleitet werden können, falls die Netzanalyse ein unerwünschtes Systemverhalten aufzeigt.

In dem Beitrag wird diese Problematik aufgegriffen und ein Verfahren vorgestellt, das auf der graphentheoretischen Analyse eines Petri-Netzes aufsetzt. Es kann notwendige Netzmodifikationen aufzeigen, die eine Korrektur des erkannten Fehlverhaltens auf Netzebene ermöglichen. Da die Modifikationen so beschaffen sind, daß sie eine Erweiterung des bestehenden Netzmodells darstellen, können sie als Abbild einer Steuerung angesehen werden, die das System gemäß den Zielvorgaben beeinflußt und ihm so das gewünschte dynamische Verhalten aufprägt.

Konzepte einer objektorientierten Steuerung für Anlagen und Maschinen

K. Hohenhauer (Institut für Kunststoffverarbeitung)

Wirtschaftliche und qualitative Aspekte in der Produktion führen zum Einsatz von Prozeßrechnern zur Kontrolle des technischen Fertigungsprozesses. Mit dem steigenden Automatisierungsgrad wächst damit der der Bedarf an leistungsfähiger und zuverlässiger Software. Die Bearbeitung nebenläufiger Prozesse, wie sie beispielsweise in Fertigungszellen vorkommen, zwingt zum Einsatz von Multitasking-Systemen. Die dadurch zunehmende Komplexität der Software ist mit herkömmlichen Programmiermethoden immer schwerer beherrschbar.

Die Analyse der Steuerungsaufgaben (Ablaufsteuerungen, Regelung, Meßwerterfassung, Prozeßvisualisierung) zeigt, daß die objektorientierte Sicht sehr gut zur Beschreibung der Zusammenhänge innerhalb eines Steuerungsprogramms geeignet ist. Die Objektklassen bilden die Bausteine eines konfigurierbaren Steuerungssystems, die den applikationsspezifischen Anteil neuer Programme reduzieren.

Die externen Signale des Fertigungsprozesses werden mit geeigneten Objektklassen im Rechner abgebildet, so daß eine Steuerung des Prozesses durch eine Zustandsänderung der Objekte dieser Klassen im sogenannten internen Prozeßabbild erfolgt. Damit übernehmen die Objekte eine aktive Funktion im Programm, die es ihnen einerseits ermöglicht, eine Änderung ihrer internen Daten selbst bekanntzugeben und andererseits gestattet, den parallelen Zugriff auf Daten oder Prozessignale, wie er in einem Multitasking-System zwangläufig vorkommt, zu koordinieren.

Die Änderung eines Objektzustands im Prozeßabbild, gleichbedeutend mit einer Änderung des externen Prozesses, kann als Ereignis aufgefaßt werden. Ereignisse können aber auch Anweisungen sein, die beispielsweise Prozeßabläufe beeinflussen und damit den externen Prozeß verändern. Hierdurch lassen sich die einzelnen Vorgänge und Abläufe eines Fertigungsprozesses mit Hilfe von Ereignissen einheitlich beschreiben. Die eigentlichen Steuerungsanweisungen werden letztendlich in Form von endlichen Automaten oder Petri Netzen programmiert, die die Regeln beinhalten, nach denen aus eintrffenden Ereignissen neue Ereignisse erzeugt werden.

Prognosen über die Zukunft der Informationstechnik sind unmöglich, aber notwendig

H. Maurer (Technische Universität Graz)

In diesem Vortrag wird zuerst erklärt, warum es so wichtig ist, die Entwicklung der Informationstechnik für die nächsten 20-30 Jahre abschätzen zu können. Dann wird allerdings nachgewiesen, daß man das Ausmaß und die Auswirkungen technologischer Entwicklungen über einen solchen Zeitraum nicht einmal annähernd vorhersagen kann. Ein spezieller Ansatz erlaubt schließlich dennoch interessante und fundierte Aussagen; und ein anderes Argument zeigt, daß selbst weiterführende Spekulationen gerechtfertigt sind.

Habilitationsschriften (Kurzfassungen)

Funktionale Programmierung mit primitiver Rekursion – formale Modelle zur Reduktionssemantik

Heiko Vogler

Eine Analyse der in Anwendungen auftretenden funktionalen Programme zeigt, daß das Programmierhilfsmittel Rekursion häufig nur in der eingeschränkten Form der primitiven Rekursion über baumförmigen Datenstrukturen auftritt. Dieses Konzept nennen wir funktionale Programmierung mit primitiver Rekursion. In der vorliegenden Arbeit gehen wir von der funktionalen Programmiersprache SRDL* aus, welche gerade den Kern dieses Konzeptes realisiert und ausschließlich die Datenstruktur Baum benutzt. Insbesondere gilt aber, daß SRDL*-Programme Funktionen berechnen, bei denen Argumente und Resultate Bäume sind.

In dieser Arbeit entwickeln wir zwei Berechnungsmodelle für die Reduktionssemantik von SRDL*-Programmen; die Modelle beschreiben die Top-down rekursive Reduktionssemantik, bzw. die Bottom-up iterative Reduktionssemantik. Dabei werden die Ergebnisbäume der in SRDL* programmierten Baumfunktionen von der Wurzel zu den Blättern bzw. von den Blättern zur Wurzel aufgebaut. Unter Berücksichtigung spezieller Teilklassen wird die Effizienz der Berechnung beider Semantiken miteinander verglichen.

Referenten: Prof. Dr. K. Indermark
Prof. Dr. W. Thomas
Weiterer Gutachter: Dr. J. Engelfriet

Tag der Habilitation: 24. Oktober 1990

Transaktionsverarbeitung in Datenbanksystemen unter Ausnutzung semantischer Information

Gottfried Vossen

Datenbanksysteme bedienen sich des ***Transaktions-Konzepts*** zur Realisierung von Parallelität und Fehlertoleranz. Im Rahmen der Beschreibung eines Datenbanksystems über eine Schichtenarchitektur lassen sich die hieraus resultierenden Aufgaben der ***Synchronisation*** von Transaktionen (Concurrency Control) und des ***Wiederanlaufs*** nach Fehlern (Recovery) auf unterschiedlichen Abstraktionsebenen lösen. Dabei besteht ein Ziel darin, durch Berücksichtigung von auf höheren Ebenen zur Verfügung stehender semantischer Information den erzielbaren Grad an Parallelität zu optimieren.

Es wird ein konkreter Ansatz beschrieben, in welchem für relationale Datenbanken die ***konzeptionelle*** Abstraktionsebene gewählt wird. Auf dieser Ebene kann speziell die Semantik von ***Update-Operationen*** ausgenutzt werden. Es wird gezeigt, welche neuen Erkenntnisse insbesondere über die Korrektheit von Schedules von Update-***Transaktionen*** dadurch ermöglicht werden; speziell werden verschiedene neue ***Serialisierbarkeits-Begriffe*** identifiziert und einem Begriff der ***Ziel-Korrektheit*** von Schedules gegenübergestellt. Unter Berücksichtigung von Recovery-Aspekten wird schließlich gezeigt, wie aus diesen Ergebnissen Rückschlüsse auf den Entwurf praktikabler (Schichten-) Scheduler für die konzeptionelle Ebene gewonnen werden können.

Referenten: Prof. Dr. W. Oberschelp
Prof. Dr. J. Paredaens
Weiterer Gutachter: Prof. Dr. B. Walter

Tag der Habilitation: 20. Juni 1990

Dissertationen (Kurzfassungen)

Generation of Picture Languages by Chomsky-Grammars – Questions of Decidability and Complexity

Friedhelm Hinz

This thesis continues the research on chain code picture languages, a concept which allows to study two-dimensional objects in formal language theory. Over an alphabet of four letters u, d, r, l encoding four directions up, down, right, left, a word can be understood as a program for a plotter by interpreting each letter as an elementary instructionto move (and draw) in a certain direction. A picture language is a set of pictures given by the execution of all programs contained in a string language. A picture language is called context-free, linear, regular, etc., resp., if it is described by a context-free, linear, regular, etc., resp. string language. Maurer, Rozenberg, and Welzl established a Chomsky hierarchy of picture languages and considered questions of decidability and complexity (1982 and 1983). One of the questions they raised is the question of descriptional complexity, which is the question of the length of a word in a string language relative to the size of the picture described. Since then a number of papers on this topic have been published.

A disadvantage of this alphabet is that only connected pictures can be described. So we extend the concept introducing colours 0, ...,n, where the colour 0 is the "invisible" colour ("drawn" by moves of a lifted pen). We show that there are significant differences between picture languages over the alphabet with colour 0, resp. without colour 0, while the difference between uncoloured (n=1) and uncoloured (n>1) pictures turns out to be not significant. For the unrestricted alphabet there are two questions of descriptional complexity: The question of word length requirement, which was mentioned above, and the question of space requirement, concerned with the space needed for the lines drawn with colour 0.

In this thesis we consider hierarchy problems (II and III), decision problems and their complexity (IV and V), and questions of descriptional complexity (VI - VIII). We introduce iteration theorems similar to the iteration theorems for string languages and use them to show that the following inclusions are proper for picture languages: "regular" included in "linear" included in "context-free" included in "context-sensitive" (II). We establish normal forms for picture languages and show that several notions coincide, namely "deterministic context-sensitive" and "recursively enumerable", "context-free" and "simple deterministic context-free", "regular" and "star-free regular" (III). While for regular languages the situation is similar in the general case and in the restricted concept (without colour 0), for linear picture languages the membership problem is undecidable in the general case and decidable in the restricted (IV). For context-free languages the membership problem is NP-complete in the restricted concept and the subpicture problem is NP-complete in both concepts; we also develop efficient algorithms for several special cases of the membership problem, including a cubic time algorithm for the superpicture problem (V). We give presice upper bounds for the descriptional complexity of regular picture languages and show that no upper bounds by recursive functions exist for linear or context-free languages (VI). For the restricted concept Sudborough and Welzl have established upper bounds; we show that these bounds are essentially optimal for linear picture languages (VII) and for context- free picture languages (VIII). The final remarks include a new undecidability result in the classical theory of regular string languages as well as some remarks viewing the theory of picture languages as a special case of the theory of inverse monoids.

Parts of the results have been published in LNCS 349, LNCS 389, and in EATCS-Bulletin 38.

Referent: Prof. Dr. W. Thomas
Korreferent: Prof. Dr. E. Welzl, Prof. Dr. W. Oberschelp

Tag der mündlichen Prüfung: 25. Juni 1990

Funktion und Leistungsaspekte von Mobilfunknetzen

Andreas Mann

Auf Funkverbindungen basierende Kommunikationssysteme finden heute eine immer weitere Verbreitung, weil zum einen Leitungen entfallen und zum anderen mobile Stationen integriert werden können. Man spricht daher auch von Mobilfunknetzen. In dieser Arbeit werden Mobilfunknetze untersucht und insbesondere der Einfluß der Stationsmobilität auf ihre Kommunikationsprotokolle analysiert. Die Modifikation von Protokollen leitungsgebundener Kommunikationssysteme genügt den Anforderungen dieser Netze in der Regel nicht.

Zwei Teillösungen zur Realisierung leistungsfähiger, dezentral organisierter, multihop Mobilfunknetze werden vorgestellt. Zum einen wird ein neues Verfahren zur Sendereichweitensteuerung vorgeschlagen, zum anderen wird ein Kanalzugriffsprotokoll entwickelt und seine Leistung unter Berücksichtigung der Stationsmobilität bewertet. Offene Probleme, die die Verbreitung dieser speziellen Kommunikationssysteme verzögern, werden diskutiert.

Die Sendereichweite von Stationen beeinflußt die Leistungsfähigkeit dezentral organisierter, multihop Mobilfunknetze entscheidend. Die Entfernung, die eine Nachricht bis zu ihrem Ziel noch zurücklegen muß, wird als neuer Parameter eingeführt. Er fließt in die Berechnung der optimalen Sendereichweite ein. Aus den Ergebnissen der mathematischen Analyse folgt ein Protokoll zur Sendereichweitensteuerung. Stationen erhöhen ihre Sendereichweite für eine Nachricht, die ihrem Ziel nahe ist, um die Wahrscheinlichkeit einer zielgerichteten Übertragung zu erhöhen. Stationen verringern ihre Sendereichweite für Nachrichten, die noch weit von ihrem Ziel entfernt sind.

Das Kanalzugriffsprotokoll CSAP wurde für dezentral organisierte, multihop Mobilfunknetze entwickelt. Stationen reservieren implizit Kanäle und tauschen über diese Nutzdaten aus. Zusätzlich werden spezielle, für die Kanalbelegung benötigte Informationen an alle Nachbarstationen verteilt. Analytische und simulative Untersuchungen zeigen, daß CSAP über weite Bereiche der Stationsmobilität leistungsfähig ist. Stationen belegen nach Interferenzstörungen, die durch die Stationsmobilität verursacht werden, schnell freie Kanäle. Die Zusatzinformationen reduzieren zwar den Durchsatz, ermöglichen dafür aber eine wesentlich effizientere Kanalbelegung.

Referent: Prof. Dr. O. Spaniol
Korreferent: Prof. Dr. B. Walter

Tag der mündlichen Prüfung: 28. Juni 1990

Eine Untersuchung über Multiprozessor-Busnetzwerke

Wisam Michael

In der vorliegenden Arbeit wird ein neues Multiprozessor-Buszugriffsverfahren vorgestellt. Nach Beschreibung des Buszugriffsverfahrens am Anfang der Arbeit wird auf seine hardwaremäßige Realisierung eingegangen. Die Busleistung wird anschließend durch analytische und simulative Verfahren untersucht.

Bei Multiprozessor-Systemen, die eine große Anzahl von Prozessoren haben, wirkt das Busnetzwerk meistens als System-Bottleneck. Um Systeme mit hoher Leistung zu bauen, wurden Busnetzwerke mit hoher Bandbreite entwickelt. Im Gegensatz zum 'Single-Bus', der die niedrigste Bandbreite aufweist, bietet der 'Crossbar-Bus' die höchste Bandbreite. Andere Netzwerke wie 'Multiple-Bus' und 'Multi Stage-Netzwerke' liegen mit ihren Bandbreiten zwischen den obengenannten Busnetzwerken.

Betrachtet man einen Speicherzugriffszyklus bei einem der oben aufgeführten Multiprozessor-Busnetzwerke, so stellt man fest, daß die Zeit minimal ist, in der der Bus effektiv verwendet wird. Der Adressbus wird nach Übertragung der Adressinformation am Anfang des Buszyklus nicht mehr verwendet. Der Datenbus wird erst bei der Datenübertragung am Ende des Buszyklus verwendet. Während des Speicherzugriffes, der die meiste Zeit eines Buszyklus einnimmt, müssen die beiden Busse auf den Speicher warten. Durch dieses Zugriffsverfahren wird die Busleistung stark vermindert.

Bei dem in dieser Arbeit vorgestellten Multiprozessor-Bus werden durch Anwendung eines neuen Zugriffsverfahrens diese Nachteile vermieden. Im Gegensatz zu einem normalen Multiprozessor-Bus, wo ein Speicherzugriff in einem Buszyklus stattfindet, geschieht hier der Speicherzugriff in drei Zyklen. Im ersten Zyklus überträgt der Prozessor seine Adressinformation durch den Adressbus zu dem Speicher-Modul. Die Verarbeitung der Daten durch den Speicher geschieht im zweiten Zyklus. Im dritten Zyklus werden die Daten zwischen dem Prozessor und dem Speicher-Modul durch den Datenbus übertragen. Durch dieses Verfahren wird vermieden, daß der Bus während der Speicherzugriffszeit auf den Speicher wartet und dadurch wird die Busleistung stark erhöht.

Referent: Prof. Dr. O. Spaniol
Korreferent: Prof. Dr.-Ing. F. Schreiber

Tag der mündlichen Prüfung: 29. Oktober 1990

Kommunikationssysteme in Echtzeitumgebungen – Modellierung und Bewertung von Fertigungs- und Intra-Vehikel-Kommunikation

Martine Schümmer

Seit einigen Jahren werden digitale Kommunikationsmedien für die Feldebene der Fertigungsumgebung (Feldbussysteme) und in der Fahrzeugkontrolle (Intra-Vehikel-Kommunikationssysteme) angeboten. Die kommerziellen Systeme sind jedoch herstellerspezifisch und bieten häufig keine offenen Kommunikationslösungen. Als nationale und internationale anerkannte Feldbusentwicklungen werden die Feld-Version von MAP (MiniMAP), der deutsche PROFIBUS, der PDV-Bus und das japanische IINET-Ringsystem (Industrienorm) den Kommunikationssystemen der Fahrzeugkontrolle CAN und ABUS gegenübergestellt.

Die Leistungsbewertung der innovativen Kommunikationssysteme kann nur zum Teil mit Hilfe der für Kommunikationssysteme konventionellen Analysemitteln vorgenommen werden. Insbesondere wegen der hohen Realzeit-Ansprüche dieser Umgebungen sind die klassischen Ansätze der Warteschlangentheorie mit ihren Mittelwertberechnungen nur beschränkt einsetzbar. Neue Metriken sowie adäquate Leistungsanalysen müssen für diese speziellen Systeme entwickelt und angewendet werden. So wird hier die Pollingverzögerung in Multimastersystemen als ein Leistungsmaß für den in Meß- und Steuerungsumgebungen anfallenden synchronen Datenaustausch (z.B. Sensordaten) definiert. Der Mittelwert der Pollingverzögerung kann mit der entwickelten Formel nach unten abgeschätzt werden. Dies gilt unabhängig von dem jeweiligen Medienzugangsverfahren.

Die Hauptforderungen an ein Echtzeitsystem sind: Rechtzeitigkeit und Gleichzeitigkeit der Datenerfassung und Ausführung von Prozessen. Für Kommunikationssysteme mit Echtzeitanspruch bedeutet das, daß nicht nur die Schnelligkeit der Übertragung eine Rolle spielt, sondern die Rechtzeitigkeit der Übertragung oder Reaktionen (Quittungen) innerhalb vorgegebenen und vorhersehbaren Zeitschranken. Die zweite Forderung bezieht sich auf die Möglichkeit zur Parallelarbeit und Synchronisation von Anwendungsprozessen. Um dies leisten zu können, muß das Medienzugangsverfahren eines Kommunikationssystems in einer verteilten Umgebung adäquate Mechanismen zur Verfügung stellen. Die wichtigsten sind in diesem Zusammenhang die Zuordnung und Abarbeitungsstrategien von Prioritäten und die Möglichkeit zu zuverlässigen Multicast-Diensten.

Nur das MiniMAP-System kann mit einer adäquaten Einstellung der Prioritätstimer das Kriterium der strengen Rechtzeitigkeit erfüllen. Alle anderen hier betrachteten Feldbussysteme (PROFIBUS und PDV) erfüllen nur abgeschwächte Rechtzeitigkeitseigenschaften. Der PROFIBUS bietet allerdings adäquate Mechanismen für die Unterstützung des synchronen Datenverkehrs. Auch im Hochlastfall hat hier das Polling Vorrang vor asynchronen, niederprioren Datenübertragungen. Kommunikationssysteme mit nachrichtenorientierter Prioritätenvergabe (IINET und CAN) können nicht einmal die abgeschwächte Rechtzeitigkeitseigenschaften erfüllen. Die Ursache dafür ist in der Vielzahl der verwendeten Prioritäten zu sehen, für die keine der untersuchten Vergabestrategien als allgemeingültig angesehen werden kann. Allerdings weisen Intra-Vehikel-Kommunikationssysteme zuverlässige Broadcast-Mechanismen auf, die von den meisten Feldbussystemen nur durch aufwendige zusätzliche Quittungsmechanismen erreicht werden können.

Insgesamt ergibt sich die Notwendigkeit Kommunikationsstrukturen nach dem Prinzip "Divide and Conquer" zu entwerfen, da die Komplexität gerade in technischen Kontrollprozessen schnell Dimensionen erreicht, die nicht mehr zu beherrschen sind. Allerding müssen bei den Aufteilungen die Kommunikationsstrukturen der einzelnen Teilprozesse und ihre Abhängigkeiten berücksichtigt werden. Die in dieser Arbeit ermittelten Ergebnisse können dabei als Richtlinie für die Leistungskapazität der verschiedenen Kommunikationssysteme verwendet werden.

Auf die Möglichkeiten der Bewertung der Kommunikationssysteme hinsichtlich des Faktors Mensch, Arbeitsgestaltung und Sicherheit wird in Kapitel 9 eingegangen.

Referent: Prof. Dr. O. Spaniol
Korreferent: Prof. Dr. T. Pfeifer

Tag der mündlichen Prüfung: 22. Oktober 1990

Liste der Diplomarbeiten

(Berücksichtigt sind die Diplomarbeiten derjenigen 106 Studenten, die im Jahre 1990 ihre Diplomprüfung abgeschlossen haben. In Klammern ist jeweils der betreuende Hochschullehrer, gegebenenfalls mit Korreferent, angegeben)

Abel, Jürgen-Karl: *Entwicklung eines syntaxgesteuerten, graphischen Petrinetzeditors* (Ameling)

Aghadavoodi, Masoud Jolfaei: *Untersuchung verschiedener Implementierungsvarianten für verteilte Simulation mit dem Simulationssystem SIC* (Schreiber)

Auer, Hans: *Strukturierung und Erstellung eines Programms mit Benutzeroberfläche zur Texturanalyse und DNA-Messung von Zellkernen* (Böcking / Oberschelp)

Bartmann, Michael: *Entwurf einer Beschreibungssprache für Robotergeometrien und Implementierung des zugehörigen Compilers* (Ameling)

Biswas, Atish: *Wissensrepräsentation und Wissensakquisition bei einem Expertensystem zur Bestimmung der Medienbeständigkeit von Kunststoffen* (Michaeli / Schinzel)

Bock, Susanna: *EFTA-Query Processing: Kostenmodell und nichtalgebraische Optimierung* (Walter)

Boll, Silke: *Ein Inferenzmechanismus für Wissen in hybrider Darstellung* (Peeken / Schinzel)

Brendel, Oliver: *Funktioneller Hirnatlas – Die Axiomatisierung und Repräsentation der Topographie* (Schinzel)

Broekmanns, Michael: *Änderungsverwaltung in einem Nichtstandard-Datenbanksystem* (Nagl)

Burg, Heribert: *Implementierung und Untersuchung paralleler Algorithmen zur schnellen Fourier-Transformation auf CRAY-Mehrprozessorrechnern* (Hoßfeld)

Buro, Michael: *Die Sigma-Funktion von Rado und das initiale Halteproblem für Turingmaschinen mit fünf binären Zuständen* (Oberschelp)

Coors, Stefan: *Entwurf einer integrierten Requirements-Engineering-Umgebung* (Nagl)

Davids, Peter: *Leistungsbewertung des DQDB-Hochgeschwindigkeitsnetzwerkes* (Spaniol)

Delonge, Franz-René: *Eine Beschreibungssprache für Digitaloberflächen* (Indermark)

Denker, Harald: *Zwischencodeoptimierung bei der parallelen Implementierung einer funktionalen Sprache* (Indermark)

Dettmar, Hartmut: *Effiziente Algorithmen und Datenstrukturen zur Optimierung des Zeitaufwandes von Erstarrungssimulationsprogrammen mittels der Finite Elemente Methode* (Sahm / Oberschelp)

Dieke, Detlev: *Ein Präprozessor zur Beobachtung der Kommunikation über die logischen Kanäle der Programmiersprache OCCAM* (Merkwitz)

Docquier, Hildegard: *ISADT - Eine überarbeitetestrukturierte Analysemethode* (Nagl)

Dorn, Adrian: *Mehrband- und Mehrkopfautomaten über Wörtern* (Thomas)

Faßbender, Heinrich: *Implementierung der call-by-need Auswertungsstrategie für macro tree-to-string transducer auf nested stack Maschinen mit geschachtelten Ausgabebändern* (Indermark)

Fliegen, Udo: *Evaluierung eines Serialisierbarkeits-Schedulers für Update-Transaktionen auf konzeptioneller Ebene relationaler Datenbanken* (Oberschelp)

Franz, Marquart: *Implementierung eines „Bounded Disorder" Zugriffsverfahrens* (Walter)

Fröhlings, Ulrich: *Entwicklung eines parallelen Ausführungsmodells für C++* (Merkwitz)

Geiler, Günther Michael: *Implementierung verteilter applikativer Arrays* (Indermark)

Genender, Georg: *Entwicklung eines lernenden Greifplanungsverfahrens zur Berechnung optimaler und sicherer Greifstellen für Zwei-Backengreifer* (Weck / Walter)

Gerhards, Herbert: *Implementierungsorientierte Spezifikation von Kommunikationsprotokollen mittels geeigneter Petrinetze am Beispiel des Logical Link Control* (Spaniol)

Glimm, Lukas: *Implementierung von primitiv-rekursiven Schemata auf attributierten push-down Maschinen* (Indermark)

Golf, Burkhard: *Entwurf eines Assemblers für einen echtzeitgeeigneten RISC-Prozessor* (Pleßmann / Merkwitz)

Hager, Rolf: *Entwicklung und Leistungsbewertung eines prioritätsorientierten Multiplexverfahrens für Mobilfunknetze* (Spaniol)

Hans, Werner: *Entwurf einer abstrakten Maschine für eine UND-parallele Implementierung der funktionalen Logik-Programmiersprache BABEL auf einem lose gekoppelten Multicomputer-System* (Indermark)

Heimann, Peter: *Dokumentenverwaltung in einer integrierten Softwareentwicklungsumgebung* (Nagl)

Heinrichs, Bernd: *Telefax in Lokalen Netzen* (Spaniol)

Herrmanns, Ulrich: *Routing in rekonfigurierbaren Transputer-Netzwerken* (Oberschelp / Merkwitz)

Hiestermann, Arno: *Automatische Analyse und Visualisierung multidimensionaler MR-Daten zur Gewebe-Klassifikation* (Repges / Oberschelp)

Hofemann, Gerd: *Vergleich der Speicherarchitekturen der Mehrprozessor-Vektorrechner CRAY X-MP und CRAY Y-MP anhand vektorisierender Algorithmen* (Hoßfeld)

Höges, Birgitta: *Segmentierung mehrdimensionaler Bildinformation zur Diagnoseunterstützung in der Magnetresonanztomographie* (Repges / Oberschelp)

Honsbrok, Jörg: *Automatische Segmentierung und dreidimensionale Rekonstruktion von Knochenstrukturen aus Computertomogrammen* (Repges / Oberschelp)

Hormanns, Peter: *Inkrementelle, tabellengesteuerte Unparser* (Nagl)

Houben, Birgitt: *Vergleichende Untersuchungen zu alternierenden und booleschen Automaten* (Thomas)

Janssen, Dietger: *Untersuchungen zum Einsatz eines Systems zum dynamischen Binden und Laden für die objektorientierte Programmierung in C++ unter dem Betriebssystem UNIX* (Haupt)

Jacobs, Stephan: *Modellierung und Leistungsbewertung von Zugangsprotokollen in VSAT-Satellitennetzen unter Berücksichtigung inhomogener Lasten* (Spaniol)

Juffernbruch, Klaus: *Partialvolumenbestimmung des Ventrikelsystems in CCT-Schnittserien* (v. Keyserlingk / Schinzel)

Kaltwasser, Josef: *Einfluß von Raum-Multiplex-Interferenzen realer Kanäle im Mikrowellenbereich auf die Leistung von dezentralen TDMA-Verfahren für mobile Kommunikation* (Spaniol)

Karsch, Stefan: *Bestimmung maximaler Flüsse in gerichteten Graphen auf MIMD-Rechnern* (Hoßfeld)

Kassubek, Jürgen: *Entwicklung und Bewertung eines Netzwerkprotokolls zur Multihopübertragung von Datenpaketen mobiler Stationen an ausgezeichnete, ortsfeste Kommunikationseinrichtungen* (Spaniol)

Kehrbusch, Hildegard: *Umsetzung von SA-Spezifikationen in Petrinetze – Studie und Werkzeugentwicklung* (Ameling)

Kiep, Andrea: *Adaptive Restauration unterabgetasteter Herzsignale durch ARMA-Modellierung zur Verbesserung der Bildqualität retrospektiv gegateter Darstellungen des schlagenden Herzens in der Kernspintomographie* (Günther / Oberschelp)

Klomp, Frank: *Funktioneller Gehirnatlas – Die Datenbankkomponente* (Schinzel)

Komnig, Detlev: *Untersuchung über die Struktur von Viren und Antiviren auf Personal Computern* (Merkwitz)

Köther, Richard: *Generierung syntaxgesteuerter Editoren* (Nagl)

Kramer, Marcel: *Erkennen und Darstellen von Bindenähten bei der Füllsimulation* (Michaeli / Oberschelp)

Kraß , Michael: *Ein Expertensystem zur Unterstützung der Modellgenerierung von Torsionsschwingungssystemen* (Troeder / Damm)

Kreuer, Dieter: *Untersuchung des CCITT Signalisierungssystems Nr. 7 im Hinblick auf eine Anwendung in Satellitennetzen; Simulation der Zeichengabestrecke* (Spaniol)

Kuchler, Nicola: *Konzipierung und Erstellung eines Expertensystems zur Auswahl von Funktionselementen am Beispiel der Teilfunktion „Dichten"* (Michaeli / Walter)

Kühnemann, Armin: *Tranformation strukturell-rekursiver Programme in Normalform* (Indermark)

Kuron, Ralf-Dieter: *R-EIDOS - Ein Ansatz zur strukturellen Analyse und Repräsentation medizinischer Klartextdiagnosen* (Repges / Oberschelp)

Kurth, Hermann-Josef: *Entwicklung eines Programms zur Prozeßoptimierung mittels der Evolutionsstrategie* (Michaeli / Merkwitz)

Lefering, Martin: *Ein Übergang von einer integrierten Requirements-Engineering-Methode zum Programmieren-im-Großen* (Nagl)

Müllender, Edgar: *Die Benutzerschnittstelle eines Mustererkennungs-Systems für HochauflösendeThoraxelektrokardiogramme* (Rau / Oberschelp)

Menzenbach, Dirk: *Entwicklung von Datenstrukturen für die Datenbasis eines Expertensystems zur Bestimmung der Medienbeständigkeit von Kunststoffen* (Michaeli / Schinzel)

Meuser, Thomas: *Leistungsbewertung getakteter Ringe als Hochgeschwindigkeitsnetze zur Integration heterogenerLastprofile* (Spaniol)

Pauly, Jörg: *Generierung einer Arbeitsraumbeschreibung für Kollisionserkennung und Bahnplanung* (Ameling)

Perin, Sinan: *Entscheidungsunterstützung bei Mehrfachzielen – Entwicklung und Implementierung einer Methode zur Zielstrukturierung* (Eisenführ / Schinzel)

Pfaller, Andreas: *Simulation des realen Prozeßverhaltens eines mechanischen Parksystems mittels einer grafischen Oberfläche zur Verhaltensanalyse eines Prozeßsteuerungssystems* (Pleßmann / Oberschelp)

Podolecki, Adam: *Entwurf und Weiterentwicklung einer Testgenerierungsmethode zur Bearbeitung sequentieller Schaltungen* (Oberschelp)

Pohl, Alfred: *Lagebestimmung von Werkstücken durch Analyse von Graubildern aus mehreren Ansichten* (Ameling)

Pütz, Ralf: *Konzipierung einer relationalen Datenbank als ein Teilmodul eines überbetrieblichen Informationssystems* (Eversheim / Oberschelp)

Reger, Hubertus: *Ein Vergleich der Multitasking-Implementierungen auf CRAY X-MP und IBM 3090* (Hoßfeld)

Reineke, Raymond: *Untersuchung des Einflusses von Freiheitsgraden in der Leistung eines mechanischen Parksystems bei variabler Gestaltung der Zielfunktion* (Pleß mann / Merkwitz)

Reuß, Gernot: *Entwicklung und Untersuchung von Bahnplanungsverfahren für einen 6-achsigen Portalroboter in einer Umgebung mit stationären Hindernissen unter besonderer Berücksichtigung der Laufzeiteffizienz* (Weck / Walter)

Reyers, Martina: *Verwendung morphologischer Information zur Verfeinerung der syntaktischen und semantischen Analyse eines natürlichsprachlichen Datenbankzugangssystems* (Oberschelp)

Ridder, Ralph: *Leistungsbewertung von MiniMAP als Kommunikationssystem der Fertigungsumgebung* (Spaniol)

Robens, Hubert: *Abstrakte Graphreduktionsmaschinen zur Implementierung funktionaler Programmiersprachen mit Striktheitsanalyse vor Laufzeit* (Indermark)

Rose, Christian: *Entwicklung einer integrierten Datenbasis für ein mittelständisches Unternehmen der Eisen- und Metallindustrie* (Hackstein / Spaniol)

Sammartino, Guiseppe: *Statecharts: Ihre Syntax, Semantik und Anwendung auf die Beschreibung paralleler Systeme* (Thomas)

Scharrel, Holger: *Grundlegende Mechanismen zur Unterstützung geographisch verteilter Software-Entwicklung* (Nagl)

Schell, Martina: *3D Objektrekonstruktion aus lichtmikroskopischen Aufnahmen durchfokussierter Erythrozyten* (Schmid-Schönbein / Oberschelp)

Scherwinski, Reinhard: *Entwurf von Sprache, Datenstruktur und Makroprozessor für die Bearbeitung von Layoutstrukturen* (Haupt)

Schleicher, Axel: *OR - parallele Implementierung der funktional-logischen Sprache BABEL* (Indermark)

Schoeller, Annemarie: *Entwurf und Entwicklung eines Softwaresystems zum Erkennen von Robotertypen und zur Bestimmung von Konfigurationsparametern der Rückwärtstransformation des Simulationssystems GROSIM* (Weck / Schinzel)

Schuckließ , Dorothea: *Entwicklung und Implementation einer Sprache zur Formulierung von Prüfaufgaben an Stanzereiprodukten* (Pfeifer / Oberschelp)

Schulz, Günter: *Sprachentwurf und Realisierungsgrundkonzepte zur Integration von Businessgrafiken in verschiedenen Trägersprachen* (Oberschelp)

Schulz, Jürgen: *Parameterbestimmung bei Filterverfahren zur Rauschunterdrückung bei digitalen Bildern* (Günther / Oberschelp)

Schumann, Torsten: *Entwurf und Implementierung eines symbolischen Runtime-Debuggers für Modula-2* (Ameling)

Seibert, Georg: *Entwurf und Implementierung eines sprachspezifischen Editors für Modula-2* (Ameling)

Seibert, Sebastian: *Quantorenhierarchien über sprachtheoretischen Relationenklassen* (Thomas)

Siempelkamp, Bernd: *Konzepte und Realisierungen für die Unterstützung von Non-First-Normal-Form Relationenmodellen durch relationale Datenbankmanagementsysteme* (Oberschelp)

Spielmann, Ralf: *Entwicklung einer Basisschicht für Graphgrammatikinterpreter* (Nagl)

Spies, Willi: *Eingeschränkte temporale Logiken und ihre automatentheoretische Charakterisierung* (Thomas)

Springorum, Harald: *Über die praktische Verwendbarkeit des Modells der endlichen Automaten als Konfigurationsbeschreibung der Benutzerschnittstelle eines mechanischen Parksystems* (Pleßmann / Schinzel)

Steffens, Karl-Friedrich: *Implementation eines Verfahrens zur Ableitung von 'modes' für Prolog Programme* (Oberschelp)

Steindor, Brigitte: *Auslegung von Umformprozessen mit Hilfe von Expertensystemen* (Kopp / Oberschelp)

Theis, Jürgen: *Erkennung der Herzzyklen in bewegungsmodulierten Signalen bei „drahtlosem" retrospektiven Gating für die kernspintomographische Darstellung des schlagenden Herzens* (Günther / Oberschelp)

Theus, Annette: *Entwurf eines EDIF Übersetzers für elektrische Netzwerk Beschreibungen* (Engl / Indermark)

Thomas, Rainer: *Parallelisierung des Max-Flow-Algorithmus von Shiloach/Vishkin auf Cray-Rechnern* (Hoßfeld)

Viehöver, Georg: *Simulation von Erythrozytenformen mit Hilfe numerischer Optimierungsverfahren auf einem Transputersystem zur Überprüfung der 'closed fluid sheets'-Hypothese zur Erklärung ubiquitärer Membraneigenschaften* (Schmid-Schönbein / Oberschelp)

Vu, Long: *Die rechnergestützte Ermittlung von Kenngröß en zur Beschreibung der Anlaufvorgänge an Webmaschinen* (Wulfhorst / Oberschelp)

Wagener, Andreas: *Lastverteilungsstrategien und Netzwerktopologien für eine Realisierung der parallelen abstrakten Maschine PAM* (Indermark)

Weineck, Eweline: *Entwicklung einer Konzeption zur statistischen Analyse umweltmedizinischer Daten* (Eikmann / Spaniol)

Westermann, Frank: *Entwurf und Implementierung typografischer Modelle als semantische Basis für einen allgemeinen Formatierprozeß* (Haupt)

Weyand, Claudia: *Entwicklung eines Nachrichtentyps zur Übertragung von Prüfauftragsdaten* (Pfeifer / Spaniol)

Wilke, Thomas: *Eine Eilenberg-Korrespondenz für reguläre ω-Sprachen* (Thomas)

Wischnewski, Romuald: *Implementierung einer Übersetzung von MIRANDA in einen getypten Lambda-Kalkül als Frontend eines parallelisierenden Compilers* (Indermark)

Zillich, Lutz: *Entwurf und Implementierung eines Systems zur dynamischen Modifikation von Programmen unter dem Betriebssystem UNIX* (Haupt)

Zimmermann, Wolfgang: *Chart-Parsing für merkmalsbasierte Grammatiken unter Berücksichtigung 'integrierter Deduktionen'* (Repges / Oberschelp)

Zündorf, Albert: *Kontrollstrukturen für die Spezifikationssprache PROGRESS* (Nagl)

Lehrveranstaltungen

Sommersemester 1990

Vorlesungen, Übungen, Praktika

Informatik II (Datenstrukturen) (V4)	Nagl
Übungen dazu (Ü2)	Börstler, Westfechtel
Rechnerstrukturen (V3)	Spaniol
Übungen dazu (Ü2)	Davids, Schümmer
Codierungstheorie (V4)	Merkwitz
Übungen dazu (Ü2)	Engel
Informationstheoretie II (V2)	Merkwitz
Übungen dazu (Ü1)	Weise
Assemblerprogrammierung für den MC 68000 (V2)	Merkwitz
Übungen dazu (Ü2)	Weise
Logik-Programmierung (V4)	Indermark
Übungen dazu (Ü2)	Loogen
Universelle Algebra (V2)	Indermark
Praktikum zum Compilerbau (Ü4)	Kuchen
Software Praktikum : (TÜ4) Funktionales Programmieren in Miranda (für Inf. ab 4. Sem.)	Maaßen
Softwaretechnik - Projektpraktikum (Ü2)	Nagl, Börstler
Informatik f. Elektrotechniker (V2)	Nagl
Übungen dazu (Ü1)	Janning
Komplexitätstheorie (V4)	Staiger
Übungen dazu (Ü2)	
Methoden in Requirements - Engineering (V2)	Hruschka
Modellierung und Bewertung von Rechensystemen II (V3)	Spaniol
Übungen dazu (Ü1)	Rupprecht, Kremer
Logische Grundlagen der künstlichen Intelligenz (V2)	Kemmerich, Oberschelp
Transputer und ihre Programmierung (V2)	Penner, Oberschelp
Übungen dazu (Ü1)	

Informatikpraktikum (Ü4) (für Math. mit Nebenfach Informatik)	Portz
Effiziente Algorithmen (V4) Übungen dazu (Ü2)	Schinzel Piron
Entwurf und Analyse paralleler Algorithmen (V4) Übungen dazu (Ü2)	Speckenmeyer

Seminare und Arbeitsgemeinschaften

AG: Verteilte und parallele Systeme	Merkwitz u. Mitarbeiter
AG Softwaretechnik: Methoden und Werkzeuge	Nagl
AG: Kryptologie	Oberschelp, Portz
AG und Seminar: Künstliche Intelligenz	Schinzel, Majewski, Piron, Strauß
AG: Frauen und Informatik	Funken, Schinzel
Seminar: Petri-Netze und verwandte Modelle (privatissime) (Ü2)	Merkwitz, Sonnenschein
Seminar: Parallele Automaten (privatissime) (Ü2)	Merkwitz, Sonnenschein
Seminar: Compileroptimierung (Ü2)	Indermark, Vogler, Herrmann
Seminar: Neuere Kapitel aus dem Gebiet Softwaretechnik (Ü2)	Nagl, von der Beeck, Kiesel
Seminar: Funknetze (Ü2)	Spaniol, Quernheim, Hübner
Seminar: Die höheren Ebenen des ISO-Referenzmodells (Ü2)	Spaniol, Jakobs, Reichert
Seminar: Datenschutz in Netzen: Probleme und Lösungsansätze (Ü2)	Spaniol, Anwar
Seminar: Grammatikformalismen zur Verarbeitung natürlicher Sprachen (Ü2)	Repges, Oberschelp Noack, Ingenerf
Seminar: Verbindungsnetzwerke und Parallele Rechnerarchitekturen (Ü2)	Penner, Klinger, Schikarski, Oberschelp
Seminar: Optimierungsalgorithmen in der Finite-Elemente-Methode (Ü2)	Oberschelp, Beinlich, Schneiders
Seminar: Nicht-Klassische Logik	Kemmerich, Oberschelp
Seminar: Informatik und Gesellschaft (Ü2)	Funken, Schinzel, Strauß

Wintersemester 1990/91

Vorlesungen, Übungen, Praktika

Informatik I (Programmierung) (V3) Übungen dazu (Ü2)	Merkwitz Sonnenschein
Systemprogrammierung (V3) Übungen dazu (Ü2)	Spaniol Davids, Schümmers
Mathematische Grundlagen der Informatik A (V2) (für Inf. u. Math. ab 3. Sem) Übungen dazu (Ü1)	Merkwitz Sonnenschein
Compilerbau (V4) Übungen dazu (Ü2)	Indermark Kuchen
Semantik von Programmiersprachen (V3) Übungen dazu (Ü1)	Steffen
Programmieren in Prolog (Ü4)	Loogen
Einführung in die Softwaretechnik (V3) Übungen dazu (Ü2)	Nagl Schleef
Methodisches Programmieren im Großen - Modellieren auf Entwurfsebene (V3) Übungen dazu (Ü2)	Nagl Kiesel, Westfechtel
Projektpraktikum Softwaretechnik (Ü2)	Nagl, Eichstätt, Heimann
Rekursionstheorie (V4) Übungen dazu (Ü2)	Staiger
Lokale Kommunikationssysteme (V2) Übungen dazu (Ü1)	Spaniol Rupprecht, Kremer
Lokale Kommunikationssysteme (V2) Übungen dazu (Ü2)	Spaniol Rupprecht, Kremer
Mathematische Logik (V3) Übungen dazu (Ü1)	Oberschelp Kemmerich
Algorithmen für Computergrafik und Bildverarbeitung (V3) Übungen dazu (Ü1)	Oberschelp, Schneiders, Schikarski
Paralleles Programmieren (V3)	Penner, Oberschelp
Informatikpraktikum (V2) (für Math. mit Nebenfach Informatik)	Portz
Semantik natürlicher Sprachen (V2)	Piron

Seminare und Arbeitsgemeinschaften

AG: Verteilte und parallele Systeme (Ü2)	Merkwitz u. Mitarbeiter
AG: Werkzeuge zurSoftwareentwicklung (Ü2)	Nagl u. Mitarbeiter
AG: Kryptologie (Ü2)	Portz
Proseminar: Konzepte von Programmiersprachen (Ü2)	Nagl, Schürr
Seminar: Programmiersprachen für parallele Systeme (privatissime) (Ü2)	Merkwitz, Engel
Seminar: Mathematische Modellierung von Parallelität (Ü2)	Indermark, Steffen, Loogen
Seminar: Unifikation und Narrowing (Ü2)	Vogler, Faßbender
Seminar: Neuere Kapitel der Softwaretechnik (Ü2)	Nagl, v.d. Beeck, Zündorf, Janning
Seminar: Satellitennetze (Ü2)	Spaniol, Quernheim, Kreuer
Seminar: Höhere Schichten des ISO-Referenzmodells (Ü2)	Spaniol, Reichert, Jakobs, Kremer
Seminar: Internetworking (Ü2)	Spaniol, Davids
Seminar: Hochgeschwindigkeitsnetze	Spaniol, Davids, Meuser
Seminar: Deduktive und objektorientierte Datenbanken (Ü2)	Oberschelp, Noack
Seminar: Parallele Sprachen und ihre Implementierung (Ü2)	Penner, Oberschelp
Seminar: Kryptologie (Ü2)	Oberschelp, Portz
Seminar: Bildverarbeitung (Ü2)	Repges, Oberschelp Höfener, Handels
Seminar: Informatik und Gesellschaft (Ü2)	Schinzel, Strauß, Wulf, Piron

Informatik-Kolloquium

04. 01. 90	Dr. C. Walther, Universität Karlsruhe *Ein Verfahren zur Automatisierung von Terminierungsbeweisen*
18. 01. 90	Dr. W. Schilz, Philips GmbH Forschungslaboratorium Aachen *Informationstechnik und Kommunikation, Aufgabengebiet der Philips*
19. 01. 90	Dr. Christian Freksa, Institut Berkeley, Kaliforniern *Die Darstellung unscharfen Wissens – Simulation vs. Charakterisierung*
22. 01. 90	Dr. Jörg Pflügler, TH Darmstadt *Ordnung ist das halbe Leben – sozialpsychologische Aspekte des Umgangs mit Computern*
25. 01. 90	Prof. Dr. Jan Paredaens, University of Antwerp/Technical University of Eindhoven, Belgien/Niederlande *Graphical Interfaces for Object-Oriented Database Systems*
01. 02. 90	Dr. H. Vogler, RWTH Aachen *Funktionale Programmierung mit primitiver Rekursion – Formale Modelle zur Reduktionssemantik*
06. 02. 90	Doz. Dr. Ludwig Staiger, Pulheim *Komogorov Komplexität und entropieähnliche Informationsmaße*
08. 02. 90	H. Keutgen, GEI *GEI-Softwareentwicklung mit Methode*
08. 02. 90	F. D. Kübler, parytec, Aachen *Parallelverarbeitung – Einführung einer neuen Technologie in der Anwendung*
09. 02. 90	Prof. Dr. Klaus Härtig, Berlin *Eine Vereinfachung und Verallgemeinerung des Gödelschen Unvollständigkeitsbeweises*
09. 02. 90	Prof. Dr. Efim B. Kinber, Latvian State University, Riga, USSR *A Language for Synthesis of Programs ands its Application for Graph Algorithms*
05. 03. 90	Prof. Dr. D. de Frutos Escrig, Universidad Complutense de Madrid, Spanien *On Narrowing Strategies for Partial Non-Strict Functions*
06. 03. 90	Dr. Ursula Goltz, GMD St. Augustin *Refinement of Actions in Causality Based Models*

11. 03. 90 Ldo. Mat. Y. Ortega Mallén, Universidad Complutense de Madrid, Spanien
Semantische Modelle für Nebenläufigkeit (True Concurrency)

29. 03. 90 Dipl.-Inform. P. Thiemann, Universität Tübingen
Einige Optimierungen in einer funktionalen Umgebung mit call-by-value Semantik

05. 04. 90 Dr. Michael Sonnenschein, RWTH Aachen
Ein objektorientiertes und datengesteuertes Programmierkonzept für Multicomputer auf der Grundlage dynamischer Petri Netze

11. 04. 90 Dipl.-Inform. H. Seebauer, Universität Bonn
Design und Implementierung einer Rechnerarchitektur für eine einfache funktionale Sprache

02. 05. 90 Priv.-Doz. Dr. E. Speckenmeyer, Universität Dortmund
Können k Prozessoren im Mittel mehr als k-mal schneller arbeiten als einer?

02. 05. 90 Doz. Dr. Ludwig Staiger, Pulheim
Formale Sprachen und ω- Sprachen

02. 05. 90 Priv.-Doz. Dr. Wolfgang Wechler, Universität Würzburg
Zu den algebraischen Grundlagen der Spezifikation abstrakter Datentypen

03. 05. 90 Prof. Dr. Guy Pujolle, Laboratoire MASI, Univ. P. et M. Curie/CNRS, Versailles, Frankreich
A Compatible ATM High Speed LAN

10. 05. 90 Dr. Thomas Herrmann, Universität Dortmund
Dispositionsspielräume bei kooperativer Arbeit mit Hilfe vernetzter Systeme – Möglichkeiten und Grenzen

10. 05. 90 Priv.-Doz. Dr. Volker Diekert, Technische Universität München
Eine Behandlung nebenläufiger Systeme mit Hilfe der Tracetheorie von Mazurkiewicz

15. 05. 90 Dr. Peter Padawitz, Universität Passau
Induktive und reduktive Gültigkeit von Hornformeln mit Gleichheit

15. 05. 90 Dr. Bernhard Steffen, Universität Aarhus, Dänemark
Automatische Verifikation endlicher verteilter Systeme

17. 05. 90 Dr. R. Wilhelm, Technische Universität Berlin
Perspektiven des Datenschutzes

25. 05. 90 Prof. Dr. H. Löffler, Informatik-Zentrum, TU Dresden
Kommunikationsmodell des CIM-Betriebes und low-cost-LAN LOTUNET

25. 05. 90 Doz. Dr. K. Irmscher, Informatik-Zentrum, TU Dresden
Modellierung von Rechner- und Rechnerverbundsystemen mit Hilfe von Warteschlangensystemen/netzen

29. 05. 90 Prof. Dr. J. Friedrich, Universität Bremen
Anthropozentrische Technikgestaltung in der Informatik, dargestellt am Beispiel der Adaptierbarkeit und Adaptivität von Benutzerschnittstellen

07. 06. 90 Prof. Dr. David Maier, Oregon Graduate Institute, z.Zt. Altair France
A Study of three Alternative Workstation-Server Architectures for Object-Oriented Database Systems

11. 06. 90 Dr. A. Kemper, Universität Karlsruhe
Anfrageoptimierung in Objektbanken

12. 06. 90 Dr. R. Unland, FernUniversität Gesamthochschule Hagen
Ein Baukastenansatz für das Transaktionsmanagement in Objektbanken

12. 06. 90 Dr. G. Snelting, TH Darmstadt
Kontextrelationen

19. 06. 90 Dr. H. Züllighoven, GMD St. Augustin
Der Entwurf objektorientierter Systeme

19. 06. 90 Dr. U. Pletat, IBM Deutschland GmbH, Stuttgart
Wissensrepräsentation und Softwarespezifikation

21. 06. 90 Prof. Dr. Egon Börger, Universita di Pisa, Italien
Eine logisch-algebraische Semantik für PROLOG

25. 06. 90 Prof. M. Rodríguez Artalejo, Universidad Complutense de Madrid, Spanien
Semantics Foundations of a Higher Order Functional Logic Language

25. 06. 90 Dr. M. Goedicke, z.Zt. Universität Gesamthochschule Essen
Sprachunterstützung für komponenten-orientierte Beschreibung von Softwaresystemen

28. 06. 90 Dr. Joost Engelfriet, Universität Leiden, Niederlande
Context-Free Graph Languages

30. 08. 90 Dr. Zoltan Fülöp, Hungarian Academy of Sciences, Sofia, Ungarn
Congruential Tree Languages are the same as Recognizable Tree Languages

13. 09. 90 Dr. Marc Engels, Kath. Universität Leuven, Belgien
Hierarchical Interconnection Networks: the Future for Parallel Processing

30. 10. 90	Dr. Sabine Thürmel, Siemens AG, München *Paralleles Lisp für eine verteilte Maschine: ein Forschungsbericht*
15. 11. 90	Prof. Dr. Tenenbaum, TU Moskau, USSR *Ideas of Relational Database Theory and Solving some Problems in other Classes of Systems*
27. 11. 90	Frau Tiziana Magaria, Universität Turin, Italien *Proof Styles for Finite State Machines in the OTTER Resolution-Based Framework*
04. 12. 90	Dr. Hardi Hungar, Universität Oldenburg *Komplexität der Erstellung von Hoareschen Beweisen*
06. 12. 90	Dr. Ursula Goltz, GMD Bonn *Ein modularer hierarischer Kalkül für den Systementwurf*
13. 12. 90	Prof. Dr. Ernst-Rüdiger Olderog, Universität Oldenburg *Verifikation paralleler Programme mit Hilfe von Transformationen*
19. 12. 90	Dr. Gudula Rünger, Universität Saarbrücken *Modulare Semantikkonzepte für Sprachen mit Nebenläufigkeit*

Statistik

Im Wintersemester 1990/91 waren an der RWTH insgesamt 1325 Studenten im Fach Informatik eingeschrieben, davon 1251 für den Diplomstudiengang. Die Zahl der Studienanfänger lag mit 201 etwas über der des Vorjahres (bei einer Anhebung der Kapazität von 120 Plätzen für Studienanfänger im Jahr 1989 auf 169 im Jahr 1990). Lag der Anteil der Frauen bei den Studienanfängern im Studiengang Informatik in den letzten Jahren bei circa 15%, so sank er im Wintersemester 1990/91 auf 7.5%. Die Zuteilung erfolgt durch die ZVS in Dortmund nach dem Zentralen Verteilungsverfahren, welches eine gleichmäßige Überbelastung der beteiligten Hochschulen sicherstellt. Ähnlich wie in den Vorjahren hatte Aachen mit 426 Bewerbern im Verhältnis zu den 201 zugeteilten Studienplätzen eine Spitzenstellung.

Die Schaubilder der folgenden Seiten geben die Entwicklung an der RWTH seit Einführung des Studiengangs Informatik im Jahre 1972 wieder.