



JAHRESBERICHT

1991

Universität Stuttgart
Institut für Informatik

Breitwiesenstraße 20-22 7000 Stuttgart 80

Herausgeber: Prof. Dr. Jochen Ludewig
Geschäftsführender Direktor
Institut für Informatik
Breitwiesenstraße 20-22
70565 Stuttgart

Redaktion: Bernhard Ziegler Tel: (0711) 7816–365

Institutsadresse:

Breitwiesenstraße 20-22 Tel: (0711) 7816–379
7000 Stuttgart 80

Inhaltsverzeichnis

Das Jahr 1991	1
1 Selbstverwaltung und Organisation	3
1.1 Leitung des Instituts für Informatik	3
1.2 Mitarbeit und Mitgliedschaft in Gremien	4
1.2.1 Universität Stuttgart	4
1.2.2 Wissenschaftliche Institutionen	7
1.2.2.1 Mitarbeit	7
1.2.2.2 Mitgliedschaft	8
2 Forschung	11
2.1 Forschungsvorhaben und Forschungsgruppen	11
2.1.1 Abteilung Betriebsssoftware	11
2.1.2 Abteilung Computer–Systeme	15
2.1.3 Abteilung Dialogsysteme	18
2.1.4 Abteilung Formale Konzepte der Informatik	29
2.1.5 Abteilung Intelligente Systeme	32
2.1.6 Abteilung Programmiersprachen und ihre Übersetzer	38
2.1.7 Abteilung Software Engineering	39
2.1.8 Abteilung Theoretische Informatik	41
2.2 Veröffentlichungen	46
2.3 Berichte	51
2.4 Vorträge	52
2.5 Tagungen	59
2.6 Herausgabe von Zeitschriften	60
2.7 Implementierungen	61

3 Fakultätsbezogene Aufgaben	64
3.1 Dekanat der Fakultät Informatik	64
3.2 Zentrale Fakultätseinrichtungen	64
3.2.1 Bibliothek	64
3.2.2 Rechnernetz	65
3.2.3 Workstations	66
3.2.4 Elektrotechnik	66
3.2.5 Mehrbenutzersysteme	67
3.2.6 PC-Pools	67
3.3 Lehre	68
3.3.1 Aufbau des Informatikstudiums	68
3.3.1.1 Diplomstudiengang Informatik	68
3.3.1.2 Nebenfachstudium Informatik	69
3.3.2 Lehrveranstaltungen	70
3.3.2.1 Lehrangebot im Sommersemester 1991	70
3.3.2.2 Lehrangebot im Wintersemester 1991/92	75
3.3.3 Informatik-Kolloquium	80
3.3.4 Habilitationen und Examensarbeiten	82
3.3.4.1 Dissertationen	82
3.3.4.2 Diplomarbeiten	83
3.3.4.3 Studienarbeiten	92
Abkürzungen	102
Personenregister	104

Das Jahr 1991

Das Institut für Informatik (IfI) bildet zusammen mit dem Institut für parallele und verteilte Höchstleistungsrechner (IPVR) den Kern der Fakultät Informatik, der bislang jüngsten der Universität Stuttgart. Da die Fakultät Informatik keinen Jahresbericht herausgibt, sind hier auch einige Fakultätsangelegenheiten erwähnt. Sie betreffen natürlich beide Institute.

Zum IfI gehörten Ende 1991 sieben Professoren und 33 wissenschaftliche Mitarbeiterinnen und Mitarbeiter. Zum technischen Personal zählten 18 Personen, in den Sekretariaten waren insgesamt sechseinhalb Stellen besetzt. Zusätzlich wurden mehrere wissenschaftliche Mitarbeiter aus Drittmitteln finanziert.

Das Jahr 1991 brachte für die Fakultät Informatik und damit auch für das IfI den seit langem gewünschten Umzug in ein gemeinsames Gebäude. Damit sind alle Probleme gelöst, die früher durch die Verteilung auf drei Standorte in der Stuttgarter Innenstadt verursacht wurden. Auch konnte die Enge einiger Einrichtungen, so vor allem der Bibliothek und der Rechnerarbeitsplätze, überwunden werden. Heute stehen uns ausreichende Büroflächen, eigene Hörsäle und Seminarräume sowie Bibliothek, studentische Arbeitsräume und eine kleine Mensa zur Verfügung.

Allerdings entstehen durch die Lage des *Informatik-Baus* im Industriegebiet zwischen den Stadtteilen Möhringen und Vaihingen neue Probleme, rein praktische durch die vergleichsweise schlechte Erreichbarkeit mit öffentlichen Verkehrsmitteln, aber auch grundsätzliche durch den fehlenden Kontakt mit Angehörigen anderer Fakultäten. Darum wird die Planung eines neuen Gebäudes auf dem Vaihinger Campus, das zu Beginn des nächsten Jahrzehnts gemeinsam mit der Mathematik bezogen werden soll, von allen Beteiligten begrüßt.

Der Umzug selbst erzwang unvermeidliche Unterbrechungen der Lehr- und Forschungstätigkeit; für einige Wochen etwa ab Frühlingsanfang wurden Kisten gepackt und Möbel gekennzeichnet, empfindliche Geräte einzeln transportiert, dann wieder zusammengestellt, in manchen Fällen repariert und eingeräumt. Vor allem die Planung und Realisierung der Infrastruktur (Rechner- und Kommunikationsnetz, Telefon usw.) machte sehr viel Mühe und dauerte länger als erhofft. Bis Mitte des Sommersemesters 1991 waren die größten Probleme gelöst; die Nachwehen sind auch am Ende des Berichtsjahres noch zu spüren.

In einem sehr fröhlichen Fest am 18. Oktober 1991 hat die Fakultät ihren erfolgreichen Um- und Einzug gefeiert. Dabei war auch Gelegenheit, all denen innerhalb und außerhalb unserer Fakultät zu danken, die sich in der Planung und Vorbereitung, während des Umzugs und bei der Auseinandersetzung mit den vielen kleinen Widrigkeiten einer solchen Aktion mit ganzer Kraft eingesetzt haben.

Beispielhaft sollen zwei Ereignisse aus dem Bereich der Forschung des Instituts erwähnt werden:

Im Rahmen der Konzertierten Aktion *Technology and Blindness* der Europäischen Gemeinschaft wurde vom 16.–17. Dezember ein Workshop *New Problems and future solutions for Man-Machine Interaction* durchgeführt. 28 Wissenschaftler aus Europa (einschl. Rußland) und den USA diskutierten über Benutzungsoberflächen mit photorealistischen Darstellungen, mit 3D-Akustik und mit Generierungssystemen für natürliche Sprache, insbesondere für sehgeschädigte Benutzer.

In der Abteilung Dialogsysteme begann im November 1991 ein von der Europäischen Gemeinschaft im Rahmen von TIDE gefördertes Projekt *Access to textual and graphical user interfaces for the blind* (GUIB). Es entwickelt Konzepte, wie existierende (graphische) Benutzungsoberflächen nachträglich modelliert werden können, damit eine Integration von akustischen Ausgaben (Sprache, Geräusche) an taktilen Anzeigen möglich wird. Insgesamt sieben Partner aus sechs europäischen Ländern und ein Industriepartner arbeiten hier zusammen.

Als junge Wissenschaft hat die Informatik überwiegend jüngere Fachvertreter. Stuttgarter Professoren sind bisher nur durch Wegberufungen ausgeschieden. Mit der ersten Emeritierung hat sich unser Institut nun auch in dieser Hinsicht normalisiert: Prof. Knödel, der die Informatik in Stuttgart aufgebaut und wesentlich geprägt hat, wurde zum Ablauf des Sommersemesters 1991 verabschiedet. Im Rahmen eines Festkolloquiums am 4. Juli 1991 wurde er vom Rektor und zahlreichen Kollegen geehrt. Der Emeritus hat sich aber keineswegs in den Ruhestand begeben, sondern baut nun tatkräftig die Informatik an der Universität Leipzig auf.

Durch diese Emeritierung wären Lehre und Forschung in der Theoretischen Informatik zusammengebrochen, wenn nicht seit August 1991 Prof. Diekert den anderen Theorie-Lehrstuhl besetzt hätte; dieser war seit 1985 leer geblieben, als Prof. Schwabhäuser starb.

Damit sind in der Fakultät Informatik nun vier von dreizehn Lehrstühlen unbesetzt; auf das Institut für Informatik entfallen davon neun Lehrstühle, von denen drei frei sind. Gegen Ende des Berichtsjahres hat Herr Dr. Roller einen Ruf auf den Lehrstuhl *Grundlagen der Informatik* angenommen, so daß wir zu Beginn des Jahres 1992 noch die beiden vakanten Lehrstühle *Programmiersprachen und ihre Übersetzer* sowie *Formale Konzepte der Informatik* haben. Wir hoffen sehr, im nächsten Jahresbericht melden zu können, daß alle Professorenstellen besetzt sind.

Februar 1992

Jochen Ludewig

1 Selbstverwaltung und Organisation

1.1 Leitung des Instituts für Informatik

Geschäftsführender Direktor

Prof. Dr. E. Lehmann	(bis 31.03.)
Prof. Dr. J. Ludewig	(ab 01.04.)

Stellvertretender Geschäftsführender Direktor

Prof. Dr. J. Ludewig	(bis 31.03.)
Prof. Dr. E. Lehmann	(ab 01.04. bis 30.09.)
Prof. Dr. R. Gunzenhäuser	(ab 01.10.)

Vorstand

Prof. Dr. W. Burkhardt	
Prof. Dr. V. Diekert	(ab 01.08.)
Prof. Dr. R. Gunzenhäuser	
Prof. Dr. W. Knödel	(bis 30.09.)
Prof. Dr. K. Lagally	
Prof. Dr. E. Lehmann	
Prof. Dr. J. Ludewig	

Geschäftsstelle

Akad. Oberrat D. Martin	
Frau I. Kolb	(bis 28.02.)
Frau R. Martin	(ab 01.03.)

Abteilungen und ihre Leiter

Betriebssoftware	Prof. Lagally
Computer Systeme	Prof. Burkhardt
Dialogsysteme	Prof. Gunzenhäuser
Formale Konzepte (ab 1.10.)	Prof. Gerber (<i>Lehrstuhlvertr. ab 1.10.</i>)
Intelligente Systeme	Prof. Lehmann
Kombinatorik und Numerische Verfahren (bis 30.9.)	Prof. Knödel (bis 30.9.)
Programmiersprachen und ihre Übersetzer	Dr. Sebesta (<i>Lehrstuhlvertr. 2.4. bis 31.7.</i>) Prof. Ludewig (<i>kommissarisch sonst</i>)
Software Engineering	Prof. Ludewig
Theorie der Informatik	Prof. Knödel (<i>kommissarisch bis 31.7.</i>) Prof. Diekert (ab 1.8.)

1.2 Mitarbeit und Mitgliedschaft in Gremien

1.2.1 Universität Stuttgart

Burkhardt, W. H.	Mitglied im erweiterten Fakultätsrat Mitglied der Studienkommission
Diekert, V.	Mitglied des Fakultätsrats Informatik (ab 1.8.91) Mitglied der Studienkommission (ab 13.11.91)
Gerber, S.	Mitglied im Fakultätsrat Informatik Mitglied im Sektionsrat Informatik an der Universität Leipzig Mitglied im Prüfungsausschuß Informatik an der Universität Leipzig
Gunzenhäuser, R.	Mitglied im Fakultätsrat der Fakultät Informatik Vorsitzender der Studienkommission Informatik Mitglied im Prüfungsausschuß Techn.-orientierter Diplomkaufmann Mitglied der Berufungskommission <i>Praktische Informatik: Bildverstehen</i> Mitglied der Berufungskommission <i>Theoretische Linguistik</i> Mitglied der Senatskommission <i>Koordinierungsausschuß Informatik</i> Mitglied der Berufungskommission <i>Computersysteme</i> (Nachfolge Burkhardt)
Knödel, W.	Direktor der Sektion Informatik an der Universität Leipzig Mitglied in der Mathematisch-Naturwissenschaftlichen Fakultät an der Universität Leipzig Kooptiertes Mitglied des Fakultätsrats Mathematik
Lagally, K.	Dekan der Fakultät Informatik (ab 1.10.) Prodekan der Fakultät Informatik (bis 30.9.) Mitglied des Fakultätsrats Informatik Kommissarischer Vorsitzender des Prüfungsausschusses Informatik (bis 31.3.) Mitglied des Prüfungsausschusses Informatik

	Mitglied der Berufungskommission <i>Programmiersprachen und ihre Übersetzer</i>
	Mitglied der Berufungskommission <i>Theoretische Informatik</i>
	Mitglied der Berufungskommission <i>Bildverständhen</i>
	Mitglied der Berufungskommission <i>Formale Konzepte der Informatik</i>
	Mitglied der Berufungskommission <i>Computerlinguistik</i> (Fak. 11)
	Senatsberichter der Berufungskommission <i>Fördertechnik</i> (Fak. 6)
Lehmann, Egbert	Geschäftsführender Direktor des IfI (bis 31.3.91) Stellvertretender Geschäftsführender Direktor des IfI (bis 30.9.91) Vorsitzender des Prüfungsausschusses Informatik Vorsitzender der Berufungskommission <i>Bildverständhen</i> Mitglied der Berufungskommission <i>Phonetik</i> (Fak. 11) Senatsberichterstatter zum Berufungsverfahren <i>Textil- und Faserchemie</i> (Fak. 3) Mitglied des Graduiertenkollegs <i>Linguistische Grundlagen für Sprachverarbeitung</i> an der Universität Stuttgart
Lichter, H.	Mitglied der Berufungskommission <i>Programmiersprachen und ihre Übersetzer</i>
Ludewig, J.	Stellvertretender geschäftsführender Direktor des IfI (bis 31.3.1991) Geschäftsführender Direktor des IfI (ab 1.4.1991) Sprecher des IVS (Informatik-Verbund Stuttgart) Mitglied des Koordinierungsausschusses Informatik (Senatskommission) Mitglied des Fakultätsrats Leiter der Software-Kommission (Kommission der Fakultät Informatik) Vorsitzender der Berufungskommission <i>Programmiersprachen und ihre Übersetzer</i> Mitglied der Berufungskommission <i>Formale Konzepte der Informatik</i>

Mitglied der Berufungskommission
Praktische Informatik der Friedrich-Schiller-Universität
Jena

Reuß, W. Mitglied des Prüfungsausschusses

Schneider, M. CSLG-Verantwortlicher für das Institut für Informatik

Tausend, B. Mitglied im erweiterten Fakultätsrat

Mitglied der Studienkommission

1.2.2 Wissenschaftliche Institutionen

1.2.2.1 Mitarbeit

Diekert, V.	Partner der EBRA-Working Group No 3166 <i>Algebraic and Syntactic Methods in Computer Science</i> (ASMICS)
Gunzenhäuser, R.	Vorsitzender des Fakultätentages Informatik (bis 30.04.91) Stellvertretender Vorsitzender des Fakultätentages Informatik (ab 01.05.91) Stellvertretender Sprecher der Fachgruppe 7.0.1 <i>Intelligente Lernsysteme der Gesellschaft für Informatik</i> Mitglied der Kommission <i>Rahmenprüfungsordnung Informatik des Fakultätentages</i> <i>und der Hochschulrektorenkonferenz</i> Mitglied von wissenschaftliche Projektberatungsgremien an den Universitäten Hagen und Karlsruhe Mitglied des wiss. Beirats der Akademischen Software Kooperation (Karlsruhe/Bonn)
Hanakata, K.	Gutachter of the Programming Committee <i>COLING92</i>
Lehmann, E.	Gutachter der DFG für den SFB 314 <i>Künstliche Intelligenz</i> Fachgutachter der DFG für den SFB 314 <i>Künstliche Intelligenz</i>
Lichter, H.	Sprecher des GI-Arbeitskreises 4.3.2 <i>Wissensbasierte Systeme für das Prototyping</i>
Ludewig, J.	Sprecher der GI-FG 4.3.1 <i>Requirements Engineering in der</i> <i>industriellen Anwendung</i>
Mahling, A.	Redaktionelles Mitglied der <i>Studio-Blätter</i> (herausgegeben von der Staatlichen Hochschule für Musik und Darstellende Kunst Stuttgart)

1.2.2.2 Mitgliedschaft

American Association for Artificial Intelligence	Lehmann, Rathke
APL–Club Germany	Schweikhardt
Association for Computing Machinery (ACM)	Burkhardt, Berger, Herczeg, Hohl, Lagally, Lehmann, Rathke, Ressel, Ziegler
ACM–SIG Artificial Intelligence	Lehmann
ACM–SIG Computer for the Physically Handicapped	Weber
Association for Computational Linguistics	Lehmann
Deutsche Mathematiker–Vereinigung	Knödel
DIN–Ausschuß ATHBeh-5 (Ausschuß Technische Hilfen für Behinderte) <i>Kommunikationshilfen für sensorisch Behinderte</i>	Schweikhardt
European Association for Theoretical Computer Science (EATCS)	Diekert, Ebinger, Knödel, Reinhardt
Gesellschaft für Angewandte Mathematik und Mechanik	Gunzenhäuser, Knödel
Gesellschaft für Informatik	Bassler, Berger, Burkert, Burkhardt, Deininger, Eggenberger, Forster, Gunzenhäuser, Hanakata, Knödel, Knopik, Kümmel, Lagally, Lehmann, Lichter, Ludewig, Merkel, Nitsche-Ruhland, Rathke, Ressel, Reuß, K. Schneider, Schöbel-Theuer, Schweikhardt, Schwille, Tausend, Ziegler, Zimmermann, Zipperer
GI–Fachgruppe 1.1.1 <i>Theoretische Informatik</i>	Knödel, Kümmel

GI–Fachgruppe 1.1.4 <i>Wissensrepräsentation</i>	Lehmann
GI–Fachgruppe 1.2.5 <i>Sprachanalyse</i>	Kümmel
GI–Fachgruppe 1.3.1 <i>Natürlichsprachliche Systeme</i>	Lehmann
GI–Fachgruppe 2.1.1 <i>Software Engineering</i>	Bassler, Deininger, Lichter, Ludewig, K. Schneider, Schwille
GI–Arbeitskreis <i>Software-Entwicklungsumgebungen</i> der GI–Fachgruppe 2.1.1 <i>Software Engineering</i>	K. Schneider
GI–Arbeitskreis <i>Informatik und Behinderte</i> der GI–Fachgruppe 2.3.1 <i>Software-Ergonomie</i>	Schweikhardt
GI–Fachausschuß 3.1 <i>Systemarchitektur</i>	Lagally
GI–Arbeitskreis 3.1.2 <i>Architektur von Betriebssystemen</i>	Lagally
GI–Fachgruppe 3.3.1 <i>Rechnernetze</i>	Burkhardt, Schweikhardt
GI–Fachgruppe 3.5.5 <i>Architekturen für hochintegrierte</i> <i>Schaltungen</i>	Kümmel
GI–Fachgruppe 4.3.1 <i>Requirements Engineering in der</i> <i>industriellen Anwendung</i>	Ludewig
GI–Arbeitskreis 4.3.2 <i>Wissensbasierte Systeme für das</i> <i>Prototyping</i>	Lichter
GI–Fachbereich 7 <i>Ausbildung und Beruf</i>	Gunzenhäuser
GI–Fachgruppe 7.0.1 <i>Intelligente Lernsysteme</i>	Gunzenhäuser

GI-Fachausschuß 7.1 <i>Informatik an Hochschulen</i>	Gunzenhäuser
GI-Arbeitskreis 7.1.0 <i>Weiterbildung in Informatik</i>	Gunzenhäuser
GI/NTG Fachgruppe <i>Fehlertolerante Systeme</i>	Burkhardt
Gesellschaft für Mathematisch-naturwissenschaftlichen Unterricht	Gunzenhäuser
Gesellschaft für Mathematik, Oekonomie und Operations Research	Knödel
IEEE	Burkhardt
IEEE Computer Society	Burkhardt, Ludewig, Rathke (Affiliate)
IEEE Design Automation Standards Subcommittee	Burkhardt
ISO/TC 173/SC 4 <i>Aids and Adaptations for Communication</i>	Schweikhardt
Occam User Group	Walter
Österreichische Mathematische Gesellschaft	Knödel
Österreichische Statistische Gesellschaft	Knödel
Schweizerische Informatikergesellschaft	Ludewig
Verein Deutscher Ingenieure (VDI)	Schöbel-Theuer

2 Forschung

2.1 Forschungsvorhaben und Forschungsgruppen

2.1.1 Abteilung Betriebsssoftware

Leiter	<i>Lagally</i>
Professor	<i>Eggenberger</i>
Wiss. Mitarbeiter	<i>Kümmel, Schimpf, Schöbel-Theuer, Ziegler</i>
Programmierer	<i>Merkel, Schlebbe</i>
Verwaltungsangestellte	<i>Kiesel (ab 1.2.)</i>

Parallele Ausführung von Logikprogrammen auf Transputer-systemen

(Stefan Schimpf)

Logikprogramme bieten im wesentlichen zwei Möglichkeiten zur Parallelisierung, ODER-Parallelität (gleichzeitige Berechnung alternativer Klauseln eines Prädikats) und UND-Parallelität (gleichzeitige Berechnung der konjunktiv verknüpften Literale einer Klausel). Bei der UND-Parallelität unterscheidet man weiterhin, ob die parallel berechneten Literale datenunabhängig sind oder nicht. Werden nur datenunabhängige Literale parallel berechnet, so spricht man von unabhängiger UND-Parallelität.

Auf der Basis eines unabhängig UND-parallelen Ausführungsmodells für Multiprozessoren mit gemeinsamem Speicher wurde ein System für Transputer entwickelt, die über keinen gemeinsamen Speicher verfügen. Das System verwaltet die Datenstrukturen zur Repräsentation der Berechnungen in Form von Stapeln, was einerseits die effiziente Berechnung sequentieller Programmfragmente erlaubt, andererseits aber die potentielle Parallelität einschränkt und zusätzliche Kommunikation zur Gewährleistung eines korrekten Stapelaufbaus erfordert. Weiterhin ist ein hohes Maß an Kommunikation zwischen den Prozessoren zur Analyse der Datenabhängigkeiten zwischen Literalen und zum Austausch von Teilergebnissen nötig.

In der aktuellen Implementierung stellt die im Verhältnis zu ihrer Rechenleistung relativ geringe Kommunikationsbandbreite der Transputer einen Engpaß dar: so kann

zwar auf einem System mit vier Transputern ein Speedup von zwei bis drei erreicht werden, die Verwendung von höheren Prozessorzahlen bringt jedoch keine wesentliche Verbesserung mit sich. Die zukünftige Forschung ist daher auf die Entwicklung von Verteilungsstrategien und Kommunikationsprotokollen gerichtet, die speziell für Transputersysteme geeignet sind.

Verfahren zum Erkennen allgemein kontextfreier Sprachen

(Thomas Schöbel-Theuer)

Bekannte Verfahren zum Erkennen kontextfreier Sprachen sind entweder auf eine Teilsprachklasse eingeschränkt und daher sehr effizient, oder sie können die ganze Sprachklasse erkennen und sind dabei ineffizient. Das an unserem Institut entwickelte Verfahren für die gesamte Sprachklasse wurde im Rahmen einer Diplomarbeit mit dem Verfahren von Earley verglichen. Hierzu mußten zunächst aussagefähige Vergleichskriterien gefunden werden. Der Vergleich ergab ein besseres Verhalten gegenüber dem Earley-Verfahren in fast allen Fällen.

Auf der Grundlage der gefundenen Meßkriterien wurden dann in einer zweiten Arbeit weitere Verfahren verglichen: Neben Earley auch dessen verbesserte Versionen von Graham/Harrison/Ruzzo sowie von Bouckaert/Pirotte/Snelling, das Tomita-Verfahren sowie dessen Verbesserung von Rekers, unser Verfahren sowie zwei experimentelle Varianten davon. Das Ergebnis dieser Untersuchung war teilweise überraschend: Bei einfachen Testgrammatiken lagen die Tomita-Verfahren oft um eine Größenordnung schlechter, teilweise bis $O(n^4)$ Zeit im Gegensatz zu $O(n^3)$ bei allen anderen Verfahren. Hingegen ergab der Test mit einer von Tomita entwickelten Grammatik für natürliche Sprache einen deutlichen Vorteil bis zum Faktor 10 für das Tomita-Verfahren. Dieses Verhalten scheint jedoch an der verwendeten Grammatik zu liegen: Eliminiert man einen großen Teil der darin verborgenen Redundanz durch Rechtsfaktorisieren, steigt die Effizienz aller anderen Verfahren bis teilweise zum Faktor 5. Dadurch konnte eine Variante unseres Verfahrens, die mit einer FIRST-Restriktion arbeitet, fast gleich gute Werte wie das Tomita-Verfahren erzielen. Da diese experimentelle Variante jedoch noch weiter verbesslungsfähig ist (Einbau einer FOLLOW-Restriktion), müssen endgültige Aussagen erst noch erarbeitet werden.

Weitere Untersuchungen beziehen sich auf die Komplexität bei stark mehrdeutigen Grammatiken: Die UBDA-Grammatik von Earley ($S \rightarrow SS \mid a$) hat eine exponentiell wachsende Anzahl von möglichen Ableitungsbäumen und benötigt bei allen bisher bekannten Parsingverfahren mindestens $\Omega(n^3)$ Zeit, sofern man einmal von Bitvektor-Implementierungen und Algorithmen auf Basis des aufwendigen Strassen-Verfahrens absieht, die im Extremfall bis auf $\Omega(n^2 \log n)$ kommen können. Bei dieser Grammatik ist es nun erstmals gelungen, $O(n^2)$ mit Hilfe eines experimentellen Verfahrens zu erzielen, das die transitive Hülle direkt abspeichert statt sie wiederholt zu berechnen. Die Untersuchungen zu diesem Thema werden zur Zeit intensiv fortgeführt.



جُحا وَحِمَارُهُ

أَتَى صَدِيقٌ إِلَى جُحا يَطْلُبُ مِنْهُ حِمَارًا لِيَرْكَبُهُ فِي سَفَرٍ قَصِيرٍ وَقَالَ لَهُ :

سَوْفَ أُعِيدُهُ إِلَيْكَ فِي الْمَسَاءِ، وَأَدْفَهُ لَكَ أُجْرَةً .

فَقَالَ جُحا :

أَنَا آسِفٌ جِدًا أَنِّي لَا أَسْتَطِيعُ أَنْ أَحْقِقَ لَكَ رَغْبَتَكَ، فَالْحِمَارُ لَيْسَ هُنَا الْيَوْمَ .

وَقَبْلَ أَنْ يُتَمِّمَ جُحا كَلَامَهُ بَدَأَ الْحِمَارُ يَنْتَقُ فِي اصْطَبَلِهِ .

فَقَالَ لَهُ صَدِيقُهُ :

إِنِّي أَئَمُّ حِمَارَكَ يَا جُحا يَنْتَقُ .

فَقَالَ لَهُ جُحا :

غَرِيبٌ أَمْرُكَ يَا صَدِيقِي ! أَتُصَدِّقُ الْحِمَارَ وَتُكَذِّبُنِي ؟

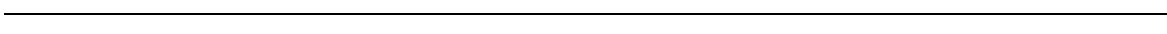


Projekt ArabTEX: Verarbeitung arabischer Texte

(Klaus Lagally)

Die im Vorjahr begonnenen Arbeiten zur Entwicklung eines Systems zur Verarbeitung arabischer Texte wurden weiter fortgeführt. Es handelt sich dabei um eine Erweiterung von L^AT_EX um ein umfangreiches T_EX-Makropaket und einen dazugehörenden Zeichensatz.

Die arabische Schrift ist eine verbundene Schreibschrift, und sie ist linksläufig. Dies bedingt, daß der normale Zeilenumbruchmechanismus von T_EX umgangen werden muß, und daß die Form eines einzelnen Zeichens von der Umgebung abhängt. Dazu kommt, daß Kurzvokale, sofern sie überhaupt geschrieben werden, als diakritische Zeichen über oder unter den Anfangskonsonanten einer Silbe gesetzt werden. Dies und eine Reihe von Sonderregeln, auf die hier nicht eingegangen wird, bedingt eine recht aufwendige Vorverarbeitung eines zu setzenden Textes.



Der im Vorjahr entwickelte Nash-Zeichensatz wurde durch eine neue Version ersetzt, die sich mehr an den üblichen Standard hält. Anfang August des Jahres wurde eine erste Fassung des Systems für Testzwecke sowohl in einer UNIX-Version wie auch in einer PC-Version freigegeben.

Die weiteren Arbeiten gehen in mehrere Richtungen:

- ArabTEX soll mit PlainTEX zusammenarbeiten.
- Ein Zeichensatz hoher Qualität ist zu entwickeln.
- ArabTEX soll auf Wunsch alle gebräuchlichen Ligaturen erzeugen.
- ArabTEX soll auch Texte in Farsi, Urdu und Pashto verarbeiten können. Ein wesentliches Problem dabei ist die Festlegung einer geeigneten Eingabenotation.

Echtzeitdatenverarbeitung auf dem Personal Computer

(*Otto Eggenberger, Uwe Berger*)

Das Programmmpaket MTS (MultiTasking Support) wurde weiterentwickelt. MTS erlaubt die quasi-parallele Ausführung mehrerer Programme als Prozesse auf einem Standard-PC. Es stellt Funktionen zur Synchronisation und zur zeit- und ereignisbezogenen Ablaufsteuerung von Prozessen bereit. Der Betriebssystemkern wurde neu strukturiert in reinen Programmcode und dynamische Datenstrukturen. Dadurch wurde zum einen der Kern ROM-fähig, zum anderen kann der Kern bei der Installation an die Erfordernisse der Anwendung angepaßt werden. Außerdem wird jetzt beim Prozeßwechsel ein numerischer Co-Prozessor berücksichtigt. Für spezielle Anwendungen, die nicht das vollständige MTS-System (interaktive Benutzerschnittstelle, virtuelle Terminals) benötigen, kann der Kern als speicherresidentes Programm installiert werden.

Von April bis Juni 1991 hatten wir als Gast einen Studenten der Ecole Central Paris, der als Praktikumsarbeit ein Programm zur Verwaltung von Druckaufträgen für MTS entwickelte.

Forschungskontakte

Hewlett Packard, Böblingen (*Rochlitzer*)
Kernforschungszentrum Karlsruhe (*Trauboth*)
Kernforschungszentrum Karlsruhe, IDT (*Jaeschke*)
Rausch+Partner GmbH, Stuttgart (*Rausch*)
MPI für Biologische Kybernetik, Tübingen (*Reichardt*)
GMD-Forschungsstelle Karlsruhe (*Grosch*)
Universität Erlangen-Nürnberg (*Fischer*)
Universität München (*Krüger*)

2.1.2 Abteilung Computer–Systeme

Leiter

Burkhardt

Mitarbeiter

*Krause, Moser, Sticcotti, Weiss (bis 30.09.),
Walter, Zimmermann, Zipperer*

Übersicht

Die Forschungsvorhaben konnten im Berichtszeitraum erfolgreich weitergeführt werden. Zusätzlich wurde ein Projekt zur Bearbeitung Neuronaler Netze auf der Basis von Hardwaremethoden neu aufgenommen.

Leistungsverhalten Paralleler Systeme

(*Walter Burkhardt*)

Die Rechnerleistung paralleler Systeme wird sehr stark durch die Einflüsse von Ein- und Ausgabe, Kommunikation zwischen den Komponenten und der Lastverteilung im System beeinflußt. Theoretisch und experimentell konnte die Abhängigkeit der Systemleistung von diesen Parametern für verschiedene Systemkonfigurationen untersucht und quantifiziert werden. Dabei hat sich gezeigt, daß z.B. die Systemleistung als Funktion des Grades der Kommunikation nach einer negativen Potenz stark abfällt. Ein weiteres wichtiges Ergebnis ist der Nachweis der Gültigkeit des Gesetzes vom verschwindenden Nutzen für die Systemleistung in Abhängigkeit vom Dimensionsgrad der Systemkonfiguration.

Parallele Abarbeitung logischer Programme auf einem Transputersystem

(*Jörg Walter*)

Ein großes Hindernis für die breite Anwendung logischer Programmierung ist deren langsame Abarbeitung auf sequentiellen Rechnern. Der Grund dafür ist, daß hier der Programmierer das zu lösende Problem nur logisch formuliert und es der Maschine überläßt, einen Weg zur Lösung zu finden. Dies geschieht durch sequentielles Absuchen des Beweisbaums nach Lösungen. Die geringe Effizienz dieses Verfahrens führt zu langen Ausführungszeiten. Man versucht deshalb seit langem, das Absuchen des Beweisbaums zu parallelisieren und so die Rechenzeiten zu verkürzen. Auf der Grundlage eines Datenflußmodells wurde ein ODER-paralleles Ausführungsmodell entwickelt, mit dem sich logische Programme auf einem Transputersystem abarbeiten lassen. Dabei bereitet vor allem die zu geringe Übertragungsleistung zwischen den Prozessoren

Probleme, da eine parallele Abarbeitung nur dann Gewinn bringt, wenn die Übertragungszeit geringer ist als die Rechenzeit einer Task. Diese ist aber nicht vorher bekannt. Gleichzeitig ist es wünschenswert, die Rechenzeit der einzelnen Tasks so gering wie möglich zu halten, um möglichst die volle Parallelität ausschöpfen zu können. Ein für diese Implementierung gewählter Kompromiß besteht darin, jeweils eine Klause des logischen Programms in eine Task abzubilden. Damit ist mit den gewählten Datenstrukturen im Allgemeinen ein Gewinn durch Parallelverarbeitung erreichbar. Von der Hardware-Seite aus ist das Problem der zu geringen Übertragungsleistung mit einer DMA-Einheit lösbar, die parallel zu jedem seriellen Link einen parallelen Datenpfad für große Datenmengen bereitstellt. Damit kann die Gesamt-Übertragungsleistung aller Links theoretisch bis zur Übertragungsleistung des Speicherbusses des Transputers gesteigert werden. Eine solche DMA-Einheit wurde entworfen und simuliert, die Realisierung auf programmierbaren Gate-Arrays war aufgrund der noch zu geringen Gatterzahl und mangelnden Verbindungs möglichkeiten auf dem Chip nicht möglich. Durch die Unterstützung von Wandel & Goltermann bei der Leiterplattenfertigung konnte erstmals auch eine komplexe Schaltung im Rahmen einer Diplomarbeit realisiert werden. Für die entworfene PC-Einsteckkarte wurde der in Deutschland entwickelte 32-Bit RISC Prozessor Hyperstone E1 verwendet, von dem ein Prototyp zur Verfügung stand. Dadurch war praktisch mit der Serienverfügbarkeit des Prozessors eine funktionsfähige Anwendung vorhanden.

Hardwareimplementation Neuronaler Netze

(Axel Zimmermann)

Der Einsatz Neuronaler Netze in so vielschichtigen Bereichen wie Automatisierungs technik, Wirtschaftsanalytik, Bildverarbeitung oder Sprachverarbeitung gewinnt zunehmend an Bedeutung. Man versucht damit einen Ausweg aus den engen Grenzen herkömmlicher Computerarchitektur zu finden, um zum Beispiel bei Mustererkennungsaufgaben wenigstens annähernd die Fähigkeiten biologischer Systeme nachbilden zu können.

Bisher werden fast ausschließlich Softwarelösungen verwendet, d.h. die eigentlich in der Natur der Neuronalen Netze liegende parallele Verarbeitungsweise wird auf konventionellen Ein-Prozessor-Systemen lediglich simuliert. Der Flexibilität einer solchen Lösung steht jedoch der gravierende Nachteil gegenüber, daß sich die Verarbeitungsgeschwindigkeit umgekehrt proportional zur Netzwerkgröße verhält und damit die Realisierung von Echtzeitsystemen praktisch unmöglich ist.

Das Forschungsprojekt befaßt sich daher mit dem Aufbau von Hardwarelösungen zur Realisierung Neuronaler Netze. In mehreren Teilprojekten (Neural Bit Slice Prozessor, Neuronen Prozessor System, stochastische Neuronen) werden Integrierte Schaltungen auf der Basis von programmierbaren Gate-Arrays entwickelt und in ihrer Lei-

stungsfähigkeit untersucht. Da die Lautsignalverarbeitung als Hauptanwendungsbe-
reich vorgesehen ist, werden außerdem noch Modelle Neuronaler Netze zur Spracher-
kennung untersucht.

Entwurf integrierter Schaltungen

(*Hans-Georg Zipperer*)

In Zusammenarbeit mit dem Institut für Mikroelektronik (Herr M. Jurczyk, Abtlg. SP&T) wurde die Architektur eines 16-bit-RISC-Prozessors für Steuerungsaufgaben (embedded control) definiert. Der Prozessor hat Harvard-Architektur und lehnt sich an die bekannte RISC-Philosophie an. Er soll insbesondere in sicherheitskritischen Systemen eingesetzt werden und ist deshalb fail-safe ausgelegt.

Zur Softwareentwicklung für diesen Prozessor wurde ein portabler Simulator und Debugger implementiert. Durch den Universal-Cross-Assembler, der in der Abteilung CS entwickelt wurde, steht auch ein einfacher Assembler zur Verfügung. Im nächsten Schritt sollen ein leistungsfähiger Assembler und ein Compiler für eine höhere Programmiersprache erstellt werden. Ferner wird zur Zeit eine Experimentier- und Demo-Platine entwickelt, mit der der Prototyp des Prozessors erprobt werden kann.

Für die programmierbaren Gate-Arrays (Xilinx LCAs) wurde ebenfalls eine Experimentier- und Demo-Platine mit flexiblen I/O-Möglichkeiten entwickelt und dank der Kooperation mit der Firma Wandel & Goltermann gefertigt. Die Platine verfügt über Batteriepufferung zur dauerhaften Speicherung der Konfiguration und soll in Lehre und Forschung eingesetzt werden.

Zur Erweiterung des LCA-Entwicklungssystems wurde ein Logiksyntheseprogramm entwickelt, das Wahrheitstabellen unter Berücksichtigung von don't-cares in minimierte Boolesche Gleichungen umwandelt.

2.1.3 Abteilung Dialogsysteme

Leiter	<i>Gunzenhäuser</i>
Wiss. Mitarbeiter	<i>Dilly (ab 1.02.), Hanakata, Herczeg, Hohl, Knopik (bis 30.09.), Kochanek (ab 1.10.), Mahling, Maier (bis 31.05.), Nitsche-Ruhland, Ressel, Schweikhhardt</i>
Programmierer	<i>Kreppein, Werner</i>
Verwaltungsangestellte	<i>Wieland</i>
Gastwissenschaftler	<i>Geltz (ab 1.07.), Weber, Schmitt do Carmo (ab 15.08.)</i>
Hilfskräfte	<i>Ayan, Bayer, Behrens, Beutelschieß, Bildstein, Dörre, Fischer, Flaisch, Förch, Geltz, Herrigel, Koppenhöfer, Koschke, Kreissl, Krieger, Lorch, Mackamul, Milan, Otto, Pauer, Rapp, Recker, Ruhland, Sander, Schneider, Schwörer, Strobel, Wichert, Zerrer</i>

1. Grundlagen, Methoden und Anwendungen der Mensch–Computer–Kommunikation

1.1 Überblick

In den Forschungsarbeiten der Abteilung Dialogsysteme werden Methoden aus der praktischen und der theoretischen Informatik sowie der kognitiven Psychologie auf Aufgabenstellungen der Mensch-Computer-Kommunikation angewandt.

Im Berichtsjahr befaßten sich die einzelnen Vorhaben mit

- (1) prototypische Methoden und Werkzeuge zur Gestaltung von wissensbasierten Benutzerschnittstellen für Anfragesysteme (Projekt DRUID im Rahmen zweier europäischer RACE-Projekte),
- (2) der Erstellung eines „intelligenten“ Tutors für den Kompositionunterricht (DFG-Forschungsvorhaben),
- (3) rechnerunterstützten Hilfsmitteln für Blinde und Sehbehinderte,
- (4) der Entwicklung und Erprobung moderner Benutzeroberflächen für Blinde und Sehgeschädigte (gemeinsam mit Fa. Papenmeier, Schwerte) sowie Vorbereitung eines für ab 1.11.92 bewilligten Projekt GUIB im Programm TIDE,
- (5) der Weiterentwicklung von objektorientierten Sprachen zur Wissensrepräsentation.

Das Berichtsjahr war gekennzeichnet durch die Fortführung der unter (1), (2) und (4) genannten Vorhaben.

Das von der DFG geförderte Projekt befaßt sich mit neuen Methoden des rechnerunterstützten Lernens im Kompositionssunterricht. Es wird gemeinsam mit der Musikhochschule Stuttgart durchgeführt.

Wesentliche Ergebnisse der seit 1978 bestehenden Forschungsgruppe „Angewandte Informatik für Blinde“ konnten auch in das mit der Firma Papenmaier begonnene Kooperationsprojekt übernommen werden. Hier wurde 1990 die Grundlagen für ein europäisches Projekt GUIB gelegt, das am 1.11.1991 beginnen konnte.

Kleinere Vorhaben der Abteilung befaßten sich ferner mit der Anwendung von Hypertext- und Hypermediensystemen sowie mit der Fortführung einer Studie über die Entwicklung und den Einsatz von qualitativ hochwertiger Lernsoftware im Bereich der Informatik. Arbeiten im Zusammenhang mit der Akademischen Software Kooperation (ASK) werden von der Firma IBM Deutschland GmbH im Rahmen eines Kooperationsvertrags unterstützt und finanziell gefördert.

Schließlich wirkte die Abteilung im Rahmen von Diplomarbeiten an einem Vorhaben über „Ethik der Informationsverarbeitung“ mit, das das Institut für Sozialforschung der Universität Stuttgart gemeinsam mit der Firma IBM Deutschland GmbH durchführt.

Auch im Berichtsjahr wurden von der Abteilung Dialogsysteme neue Aktivitäten im Bereich der Informatikausbildung initiiert. Lehrveranstaltungen „Grundlagen der Informatik“ für die Studiengänge Luft- und Raumfahrttechnik, Physik, Vermessungswesen und andere technische Studiengänge sowie parallel dazu für den Studiengang „technisch orient. Diplomkaufmann“ wurden im SS 1991 und im WS 1991/92 erneut abgehalten.

Die Diskussionen über einen neuen Studienplan im Grundstudium Informatik konnte Ende 1991 erfolgreich abgeschlossen werden. Sie finden ihre Fortsetzung in der Beratung eines neuen Studienplans für das Hauptstudium Informatik.

1.2 Forschungsgruppe DRUID: GUIDANCE

gefördert von der Firma SEL

(Jürgen Herczeg, Hubertus Hohl, Matthias Ressel)

In diesem Vorhaben arbeitet die Forschungsgruppe DRUID seit 1.4.1989 mit einer Forschungsgruppe der Firma SEL, Pforzheim, im Rahmen der europäischen RACE-Projekte MCPR (R1038) und GUIDANCE (R1067) zusammen.

Am Gesamtprojekt sind verschiedene internationale Industriepartner beteiligt. Ziel ist es, schnelle Breitbandkommunikationstechniken zu entwickeln und anhand von konkreten Anwendungssystemen zu demonstrieren. Die Breitbandkommunikation soll zum Austausch von Multimedia-Information genutzt werden in Form von Texten, Festbildern, Bewegtbildern (Videos) sowie akustischer Information (Sprache und Musik).

Darüber hinaus sind in das System auch Kommunikationstechniken wie beispielsweise Videokonferenz zu integrieren. Als Anwendung wird ein Informationssystem zum Abrufen und Erstellen von Reiseinformation entwickelt. Die zugrundeliegenden Daten sind objektorientiert repräsentiert und bilden ein Netz aus Informationsknoten, die Verweise auf verschiedenartig gespeicherte Multimedia-Informationen enthalten. Beim Abrufen der gewünschten Informationen werden menü- und formulargesteuerte Anfragen und Browsingtechniken, wie sie in Hypertextsystemen verwendet werden, miteinander verknüpft. Die formalen Anfragen dienen dabei im wesentlichen dazu, geeignete Einstiegspunkte für das Browsing zu finden. Zum Erstellen und Verknüpfen der Informationsknoten wird ein System mit graphischer Benutzerschnittstelle entwickelt, mit dem mehrere Autoren an verschiedenen Workstations miteinander kommunizieren und Dokumente bearbeiten können. Innerhalb des Projektes GUIDANCE werden formale Benutzerschnittstellenspezifikationen für ein *Reader Interface* (zum Abrufen von Informationen) und ein *Production Interface* (zum Erstellen von Informationen) entworfen, anhand derer die Systeme implementiert werden.

Folgende Arbeiten wurden von der Forschungsgruppe DRUID im Berichtsjahr durchgeführt:

Die Implementierung des Reader Interface wurde abgeschlossen und auf die Zielarchitektur SUN SPARCstation mit speziellem Videoboard und angeschlossener Bildplatte und Kamera portiert. Die Grundfunktionalität des Production Interface mit verschiedenen Kommunikationsmöglichkeiten wurde realisiert.

Die Systeme wurden in Common Lisp (einschließlich CLOS) implementiert. Die Implementierung der Benutzerschnittstellen erfolgte mit Hilfe des von der Forschungsgruppe DRUID entwickelten objektorientierten Benutzerschnittstellen-Baukastens XIT für das X Window System.

1.3 Forschungsprojekt: Computereinsatz im Kompositionunterricht

gefördert von der Deutschen Forschungsgemeinschaft

(Heinz-Dieter Böcker (GMD Darmstadt), Andreas Mahling, Rainer Wehinger (Staatl. Hochschule für Musik und Darst. Kunst Stuttgart))

Projektschwerpunkte bilden die musikalische Wissensrepräsentation sowie die Gestaltung intelligenter computergestützter Kompositionswerkzeuge und deren Einsatz in der Lehre.

Nachdem 1990 bereits ein Modul zur Umwandlung von MIDI¹-Daten in die herkömmliche Notenschrift implementiert wurde, folgte in diesem Jahr die Umsetzung von Notenschrift in MIDI Ereignislisten.

¹Musical Instrument Digital Interface ist eine standardisierte serielle Schnittstelle zur Kopplung von elektronischen Musikinstrumenten und/oder Computern.

Zur detaillierteren Bearbeitung einzelner MIDI Ereignisse wurde ein graphischer *MIDI-EventList-Editor* implementiert, mit dem selbst einzelne Parameter einer MIDI-Nachricht im Gesamtkontext und übersichtlich bearbeitet werden können.

Der 1990 entstandene Prototyp des Phrasenstruktur-Editors wurde konzeptionell neu gestaltet und ein weiterer Prototyp implementiert. Mit ihm kann musikalisches Material, beispielsweise eine Melodie oder eine Akkordfolge, durch Netze von rechteckigen Kästchen, die Prozeduren visualisieren, manipuliert bzw. generiert werden (siehe Abb. 2.1).

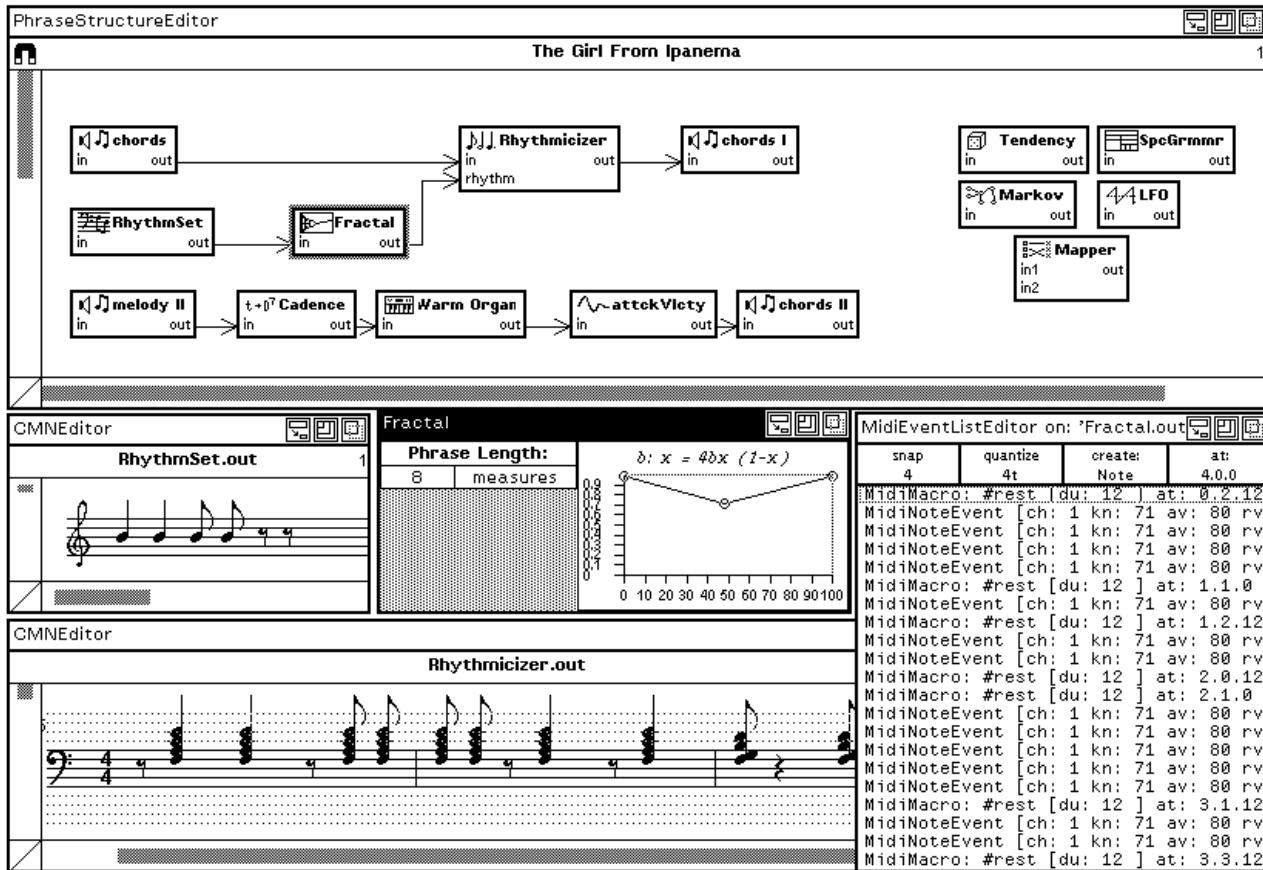


Abbildung 2.1: Ausschnitt einer Arbeitssitzung mit dem Phrasenstruktur-Editor.

Die Ankopplung des Phrasenstruktur-Editors an den MIDI-Sequenzer wurde verbessert. Ein Arrangement-Editor erlaubt die Gestaltung des formalen Aufbaus einer Komposition aus den mit Hilfe des Phrasenstruktur-Editors entstandenen Phrasen bzw. den im Sequenzer enthaltenen Patterns. Während im Sequenzer nur bereits existierende Kompositionsfragmente (Patterns) arrangiert werden können, erlaubt der Arrangement-Editor das Arrangieren virtueller Phrasen, d.h. von Platzhaltern, die noch mit „Inhalt“ gefüllt werden müssen. Die mit Hilfe des Arrangement-Editors entworfene musikalische Form dient dann dem Phrasenstruktur-Editor nach der Auswertung einer Phrasenstruktur dazu, die generierten Patterns entsprechend in den Spurenplan des MIDI-Sequenzers einzutragen.

Im Rahmen einer Diplomarbeit wurde ein Tutorensystem entworfen und implementiert. Es unterstützt den Entwurf von tutoriellen Systemen für die unterschiedlichsten musikalischen Gegenstandsbereiche. Am Beispiel eines tutoriellen Systems für den Palestrina-Kontrapunkt wurden die Fähigkeiten des Systems demonstriert. Die zugrundeliegende Musikwissensbasis erlaubt dem System einen wesentlich flexibleren und damit für den Benutzer attraktiveren Dialog. So kann z.B. mittels *semantischer* Anfragen gezielt nach Beispielen gesucht werden, die nach genauem Studium den Lernenden bei der Bewältigung der gestellten Aufgabe unterstützen. Die Ergebnisse einer Anfrage werden graphisch, beispielsweise durch Einrahmung, dargestellt. Ferner stehen ein Notizbuch und ein Musiklexikon zur Verfügung.

Im Gegensatz zum Phrasenstruktur-Editor ist *Masque* ein echtzeitfähiges Kompositionssystem. Mit Hilfe von Bühnen und Akteuren können ganze Szenen entworfen werden, die dann letztendlich die Komposition bzw. ihre Struktur repräsentieren. Je nach Aufgabe besitzen die Akteure ein anderes Erscheinungsbild. Der Rahmen reicht von Akteuren, die nur mit einer einzigen Note umgehen können, bis zu solchen, die das komplette Scheduling von „Szenenabläufen“ übernehmen. *Masque* verfügt auch über die Fähigkeit, neue Akteure mit Hilfe des Benutzers anzulernen und dies, ohne daß hierzu Kenntnisse der zugrundeliegenden Implementierungssprache notwendig sind. Das Repertoire eines Akteurs kann in speziellen grafischen Editoren kontrolliert und manipuliert werden. Das Anlernen geschieht durch schlichtes Vorführen (der Definition von Makros ähnlich).

Tonika ist ein System für die computergestützte harmonische Analyse tonaler Musik und macht von der Musikwissensbasis sehr starken Gebrauch. Das System ist u. a. ein sehr gutes Beispiel für die Flexibilität des Tutorensystems, auf dessen Klassenhierarchie es aufsetzt. *Tonika* belegt sehr anschaulich, daß mit einem *wissenbasierten* Musiksystem ein qualitativ höheres Niveau in der computergestützten musikalischen Arbeit erreicht wird. Auf der Grundlage einer flexiblen Analysestrategie ist das System in der Lage, harmonische Zusammenhänge in einem Musikstück zu identifizieren und zu bezeichnen. Sogar Modulationen werden in vielen Fällen richtig erkannt. Fehlerhafte oder für das Programm nicht analysierbare Stellen können vom Benutzer „nachgesetzt“ bzw. ergänzt werden; das System versucht dann, die Analyse aufgrund der neuen Angaben mit den weiterhin gültigen Teilen der Analyse in Einklang zu bringen. In speziellen Browsern können die Bezüge zwischen den harmonischen Funktionen verfolgt und ihre Repräsentanten, die Akkorde, inspiziert werden.

Der Austausch von Daten zwischen unterschiedlichen Musiksystemen ist eine sehr wichtige Eigenschaft, da die Ansätze, die die jeweiligen Systeme zur Unterstützung der musikalischen Arbeit verfolgen, recht unterschiedlich und in der Regel nicht für jede Aufgabenstellung gleich gut geeignet sind. Zu diesem Zweck wurde das MIDI-Standard-File Format (Typ 0, 1 und 2) in das bestehende Musiksystem integriert, das von nahezu allen Herstellern kommerzieller Musikprogramme unterstützt wird.

Bei *Pinkie* handelt es sich um ein regelbasiertes System zur teilautomatischen Generierung von Fingersätzen. Es ist noch in einem vorläufigen und nicht gänzlich einsatzfähigen Zustand. Das System ist in der Lage, Fakten der Wissensbasis bereits vor der Abarbeitung der Regelbasis mit Bedingungs- bzw. Aktionsteilen von potentiell betroffenen Regeln zu verknüpfen. Das System erlaubt, mittels eigens hierfür entworfener Werkzeuge die Abarbeitung einer Regelmenge zu verfolgen (Tracer, Stepper) und erlaubt dem Benutzer, interaktiv in diesen Ablauf einzugreifen und Veränderungen an den Regeln bzw. der Wissenbasis vorzunehmen.

1.4 Forschungsgruppe „Angewandte Informatik für Blinde“

(*Waltraud Schweikhardt, Alfred Werner,
Wolfgang Kreissl, Gertrud Pauer (wissenschaftliche Hilfskräfte)*)

Tastbare Wiedergabe gedruckter Dokumente:

Unsere Arbeiten, Blinden gedruckte Materialien in tastbarer Form zugänglich zu machen, wurden weitergeführt. Wir benützten zur Weiterentwicklung unserer Algorithmen einen Rechner vom Typ IBM PS/2, an den ein neubeschaffter Scanner angeschlossen wurde. Zur Texterkennung verwenden wir ein gekauftes OCR-Programm, das Fließtext sehr gut in ASCII-Zeichen umsetzt.

Wir beschäftigten uns in diesem Jahr insbesondere damit, den Aufbau einer Seite zu erfassen, indem sie per Programm in Teile zerlegt wird, die durch waagrechte und senkrechte leere Bereiche isoliert sind. Weiter können zusammenhängende schwarze Bereiche oder Linien entnommen werden. So können besondere Zeichen wie z. B. Wurzelzeichen gefunden werden, was Voraussetzung für das Interpretieren von Formeln ist.

Unsere Überlegungen dazu, wie Blinde selbst im Dialog an der Gestaltung einer Vorlage in tastbarer Form mitwirken können, führte zum Entwurf einer Dialogschnittstelle, die in APL2 implementiert wird.

Studieninformationen über Btx:

In das Kommunikationssystem Bildschirmtext (Btx) der Deutschen Bundespost Telekom wurden eine Reihe von Informationsseiten über den Diplomstudiengang Informatik eingebracht. Zusätzlich wurde eine „Antwortseite“ geschrieben, über die uns Mitteilungen geschickt werden können.

1.5 Forschungsprojekt: Zugang zu graphischen Benutzeroberflächen für Blinde

gefördert von der Fa. F. H. Papenmeier

(Gerhard Weber, Dirk Kochanek (ab 07.10.91))

Graphische Benutzungsoberflächen sind bisher Blinden nahezu nicht zugänglich. Weder (1) der Zugang zu Text, (2) der Zugang zu Grafiken noch (3) das Arbeiten mit einer Maus ist für Blinde sinnvoll möglich.

Eine Pilotstudie wurde entwickelt, die es erlaubt, alle drei Hürden an einem konkreten Beispiel zusammen mit blinden Benutzern zu untersuchen. Zum ersten Mal kann in dieser Pilotstudie Text, der in Pixel zerlegt vorliegt, umgesetzt werden, das Layout (preview) eines Textes interpretiert werden und gleichzeitig neben der Tastatur mit einer Maus gearbeitet werden. Es werden Konzepte entwickelt, wie existierende Benutzerschnittstellen nachträglich modelliert werden können, damit eine Integration von akustischen Ausgaben (Sprache, Geräusche) und taktilen Anzeigen möglich wird. Die Pilotstudie GRABRA konnte auf einer Ausstellung für Rehabilitationstechnologie einem breiten Publikum vorgestellt werden.

Das von der Europäischen Gemeinschaft im Rahmen von TIDE geförderte Projekt “Access to textual and graphical user interfaces for the blind” (GUIB) baut auf den Erfahrungen aus dieser Pilotstudie auf und entwickelt einen allgemeineren Ansatz zur Anpassung beliebiger Anwendungsprogramme, die unter den graphischen Benutzeroberflächen X-Windows oder MS-Windows arbeiten (Beginn: 1. Dezember).

Ein Workshop “New Problems and Future Solutions for Man-Machine Interaction” wurde vom 16.–17. Dezember im Rahmen der Konzertierten Aktion “Technology and Blindness” der Europäischen Gemeinschaft durchgeführt. 28 Wissenschaftler aus Europa (einschl. Russland) und den USA diskutierten den Zugang z. B. zu photorealistischen Darstellungen durch Einsatz von Hyperrendering, von 3D-Akustik oder von Generierungssystemen für natürliche Sprache und Graphiken.

1.6 Forschungsthema: Integration von X-tools in die objektorientierte Sprache COOL

(Kenji Hanakata)

Im Berichtsjahr wurden nach der erfolgreichen Entwicklung des Prototyps von COOL-Browser in X-intrinsics Xt-Bibliotheksfunktionen in COOL integriert. Somit können jetzt die COOL-Anwender X-Window *interaktiv* in COOL programmieren. Das heißt, die sonst sehr zeitaufwendige X-Programmierungsarbeit für z. B. Benutzerschnittstellen individueller Anwendungen in C ist dadurch sehr einfach geworden. Die kleinsten

routinierten Arbeiten am Bildschirm können durch “**Kick**” eines dafür kreierten Buttons erledigt werden. Zur Erprobung dieser mächtigen graphischen Umgebung wurde ein linguistisches “**Workbench**” prototypisch entwickelt.

Als weitere Belastungsprobe in der Praxis soll nun eine umfangreiche Benutzerschnittstelle für das Expertensystem “*Digester*” auf der Basis der Doktorarbeiten von Frau Csima (1983) und Herrn Riekert (1986) entwickelt werden, welche damals auf **Lisp** implementiert wurden.

Das integrierte COOL-Programmierungssystem weist auf die Möglichkeit hin, die bisher hauptsächlich im Lisp-Kulturkreis entwickelten KI-Technologien in die harten industriellen Umgebungen zu transferieren. Das COOL-System findet ein reges Echo in der Industrie.

1.7 Ein Hypermedia-Lern- und Autorensystem

(Doris Nitsche-Ruhland)

Hypermediasysteme finden in den letzten Jahren zunehmend Einsatz im Bereich des computerunterstützten Lernens. Durch den Einsatz von Hypermedia im Bereich CAI ergeben sich neue interessante Möglichkeiten, aber auch neue Probleme, da sowohl Lernende als auch Autoren von Lernsystemen den Umgang und sinnvollen Einsatz von Hypermedia lernen müssen. Da Autoren von Lernsystemen oftmals auf dem Gebiet des Lehrstoffs Experten sind, kann man nicht von ihnen erwarten, daß sie auch Designexperten für Hypermediasysteme sind und dieses Medium geeignet einsetzen.

Im Berichtsjahr wurden Konzepte für ein Autorensystem entwickelt, das den Autor bei der Entwicklung eines Hypermedialernsystems unterstützt. Ausgehend von einem Designmodell wird dem Autor die für das Designstadium notwendige Unterstützung in Form eines Kritikers geboten. Der Schwerpunkt bei diesem System liegt dabei im Aufbau und in der Strukturierung der Wissensbasis und in der Fragestellung, wieviel Systemführung in welchen Lernsituationen angebracht ist, da sich Hypermediasysteme im Gegensatz zu tutoriellen Systemen gerade durch die Lernerkontrolle auszeichnen. Mit der Implementierung eines prototypischen Systems in HyperNeWS unter Open-Windows wurde begonnen.

1.8 Wissensbasierte Visualisierung von allgemeinen Datenstrukturen

(Markus Geltz)

Durch geeignete Visualisierung können dem Benutzer Datenstrukturen so sichtbar gemacht werden, wie er selbst sie versteht und anderen versinnbildlichen würde. Meist können bildliche Darstellungen von Daten schneller erfaßt werden, zudem dienen sie einer Komprimierung von Information („Ein Bild sagt mehr als tausend Worte“).

Ziele der Arbeit sind dabei:

- Entwicklung einer Taxonomie von „Datenkonzepten“
- Visualisierung sollte kontextabhängig erfolgen
- Beschreibung der graphischen Datenstrukturen über Constraints
- Entwicklung eines Beispiel-Editors, durch den Regeln extrahiert werden können.

Die Implementierung eines Prototypen erfolgt in Smalltalk-80 Release 4.0 unter X-Windows. Das Projekt ist eine Kooperation zwischen der Abteilung Dialogsysteme und der Abteilung ZWI der Robert Bosch GmbH.

2. Informatikausbildung

Im Berichtsjahr wirkte der Abteilungsleiter wiederum in mehreren Gremien mit, die sich mit der Informatikausbildung an Hochschulen und mit dem Informatikunterricht an Gymnasien befassen, so im Fachbereich „Ausbildung und Beruf“ und im Fachausschuß „Informatikausbildung in Studiengängen an Hochschulen“ – und mehreren Untergremien – der Gesellschaft für Informatik.

Als Vorsitzender bzw. stellvertretender Vorsitzender des Fakultätentages Informatik hatte er manigfache Aufgabenstellungen der Informatikausbildung und -Weiterbildung initiativ zu verfolgen und in unterschiedlichen Gremien – insbesondere der Hochschulrektorenkonferenz (HRK) – zu vertreten. Im Berichtsjahr konzentrierten sich die Arbeiten auf den Abschluß einer neuen Rahmenprüfungsordnung Informatik an wissenschaftlichen Hochschulen sowie auf die weiterführende Integration von wissenschaftlichen Hochschulen in den neuen Bundesländern in das bisherige Hochschulsystem. Nach Abschluß eines umfangreichen Begutachtungsverfahrens konnten drei Technische Universitäten in den Fakultätentag Informatik aufgenommen werden. Weitere Hochschulen sollen 1992 aufgenommen werden. Ende 1991 wurde Herr Gunzenhäuser in die Fachkommission Informatik der Technischen Universität Dresden berufen. Dieser Fachkommission wurde die Aufgabe übertragen, im Jahre 1992 sämtliche Hochschullehrer und Wissenschaftliche Mitarbeiter fachlich und persönlich auf ihre Eignung für die Lehraufgaben an der Universität zu überprüfen. Auf einer Vortrags- und Studienreise konnte Herr Gunzenhäuser die notwendigen Kontakte in Dresden knüpfen sowie erweiterte Förderungsmaßnahmen für die Informatikausbildung initiieren.

Im Berichtsjahr war Herr Gunzenhäuser ferner als Mitglied von Beratungsgremien an den Universitäten Hagen (Thema: Forschungsvorhaben im Bereich des rechnerunterstützten Lernens) und Karlsruhe (Ausbildung blinder und sehgeschädigter Informatikstudenten) tätig.

Gemeinsam mit Prof. Mandl (Universität München) wurde zum 5. Mal ein Workshop über „Intelligente Lernsysteme“ vorbereitet und auf Schloß Rauischholzhausen bei Gießen durchgeführt. Ein weiteres Fachgespräch zu diesem Thema wird im Rahmen der 22. Jahrestagung der Gesellschaft für Informatik e. V. stattfinden; sie wird von Herrn Gunzenhäuser als Sprecher des Programmausschusses vorbereitet.

Forschungkontakte

Advanced Telecommunications Research Institute International (ATR)
(*Mr. M. Suzuki, Miss M. Hosaka*)

Carnegie Mellon Universität, Dep. of Computer Science, Pittsburgh, Pa, USA
(*Prof. A. Newell, Prof. H.A. Simon, Prof. G. Stelle, Prof. J. Carbonell,
Prof. Ph. Hayes*)

Deutsche Blindenstudienanstalt Marburg a.d. Lahn
(*K. Britz und R. F. W. Witte*)

Deutsches Taubblindenvwerk Hannover (*F. Zekel*)

Electronical Laboratories Tsukuba, Japan (*Prof. Yokoyama, Prof. Ishizaki*)

Firma F. H. Papenmeier GmbH, Schwerte (*J. Bornschein*)

Forschungsinstitut für Anwendungsorientierte Wissensverarbeitung an der
Universität Ulm (FAW)

(*Prof. W. Radermacher, Dr. D. Rösner, Dr. W.-F. Riekert*)

Gesellschaft für Mathematik und Datenverarbeitung (GMD) St. Augustin
(*Dr. Hoschka, Dr. Dzida, Dr. Wißkirchen*)

Hewlett Packard Research Laboratory Böblingen (*Frau A. Sonntag, Herr Ritter*)

IBM Deutschland GmbH Stuttgart, Bereich Wissenschaft (*Dr. W. Glatthaar*)

IBM Deutschland GmbH, Labor Schönaich (*Dr. Th. Fehrle*)

Industrial Products Research Institute, Tsukuba, Japan (*Dr. M. Akamuso*)

Institut für Arbeitswirtschaft und Organisation der Fraunhofer-Gesellschaft
Stuttgart (*Prof. Bullinger, J. Ziegler und wissenschaftliche Mitarbeiter*)

Institut für Informatik, FU Berlin (*Prof. Dr. Th. Strothotte*)

Institut für Informatik der Universität Karlsruhe

(*Prof. A. Schmitt, Modellversuch Informatik für Blinde*)

Institut für Maschinelle Sprachverarbeitung der Universität Stuttgart
(*Prof. Chr. Rohrer*)

Institute of Computer Science - FORTH, Heraklion, Griechenland
(*Dr. C. Stephanidis*)

International Association of Pattern Recognition (*Prof. T. Pavlidis*)

METEC Ingenieur GmbH, Medizintechnik - Feinwerktechnik, Stuttgart

Overbrook School for the Blind, International Program, Philadelphia, Pa., USA
(*Frau A. Rimrott*)

Queen Mary and Westfield College, London, England
(*Dr. P. Johnson, H. Johnson*)

SEL Forschungszentrum Pforzheim
(*A. Lesch, M. Weiß, Dr. P. Byerley und wissenschaftliche Mitarbeiter*)

TA Triumph Adler AG, Forschungszentrum Nürnberg
(*Dr. R. Lutze und wissenschaftliche Mitarbeiter*)

Technische Universität Dresden (*Prof. Kalfa, Prof. Liskowsky*)

Universität des Saarlands, Saarbrücken (*Prof. Wahlster*)

Universität Uppsala, Schweden (*Prof. Dr. G. Jannson*)

University of Colorado at Boulder, Dep. of Computer Science
(*Prof. G. Fischer, Prof. C. Kintsch, Dr. A. Lemke*)

University of Kyoto, Japan (*Prof. N. Nagao, Prof. Nakamura, Prof. Tsujii*)

University of Western Ontario, Dep. of Computer Science, London, Kanada
(*Prof. H. Jürgensen*)

Uppsala Universitet Matematiska Institutionen (*Prof. S. Christofferson*)

Uppsala Universitet, Institutionen för Teknisk Databehandling (*Prof. Sjöberg*)

2.1.4 Abteilung Formale Konzepte der Informatik

bis 30.09.91:

Abteilung Kombinatorik und Numerische Verfahren

Leiter	<i>Knödel (bis 30.09.1991)</i> <i>Gerber (Lehrstuhlvertretung ab 01.10.1991)</i>
Mitarbeiter	<i>Berger, Ebinger, Liedtke, Mijderwijk, Photien</i>
Hilfskräfte	<i>Bertol, Eberhardt, Fritz, Ketelhut, Raichle, Schreiner, I. Weber</i>

Funktionale Programmiersysteme

(Siegmar Gerber)

Es liegt eine algebraische Beschreibung funktionaler Programmiersysteme und deren Anwendung für funktionale Spezifikationen vor. Ausgehend von der Behandlung algebraischer Theorien im Rahmen von Gleichungskalkülen und einem entsprechenden Modellbegriff wurden funktionale Programmiersysteme auf der Basis einer Zweiebenenstruktur (Funktions- und Datenebene) und deren Spezifikation studiert. Rekursive Erweiterungen dieser Spezifikation sind durch entsprechende Funktionstyperweiterungen charakterisiert. Für so eingeführte funktionale Spezifikationen existieren initiale Modelle, die durch Termalgebren beschreibbar sind.

In Anwendung der theoretischen Vorarbeiten sollen konkrete Spezifikationssysteme implementiert werden. Diese Systeme sollen zur Entwicklung von Hilfsmitteln für den Entwurf und die Interpretation funktionaler Programme beitragen.

Zufallsorakel und probabilistische Modelle

(Werner Ebinger)

Eine Orakelturingmaschine rechnet „relativ“ zu einer Orakelsprache. Ein zufällig gewähltes Orakel heißt Zufallsorakel. Es wurde nach Zusammenhängen zu probabilistischen Berechnungsmodellen und interaktiven Beweissystemen gesucht. Durch R. V. Book wurde die Aufmerksamkeit auch auf algorithmisch zufällige Sprachen gelenkt, wo ein enger Zusammenhang zur Kolmogorov-Komplexität besteht.

Mazurkiewicz-Spuren: Charakterisierung von Sprachen aus unendlichen Spuren und allgemeineren Graphen

(Werner Ebinger)

Die Theorie der Mazurkiewicz-Spuren (Traces) ist ein wichtiges Werkzeug für die Erforschung nebenläufiger Systeme. Sie geht zurück auf die Arbeit von A. Mazurkiewicz (1977), der sie als Halbordnungssemantik für elementare Systeme einföhrte. Die Spur-Theorie steht in enger Verbindung zur Automatentheorie, und so betrachtet man als erstes auch erkennbare (reguläre) Sprachen von Mazurkiewicz-Spuren. Die Theorie von unendlichen Spuren als Modell für Systeme, die nicht terminieren, ist noch nicht weit entwickelt.

In der zweiten Jahreshälfte definierten P. Gastin und A. Petit asynchrone Büchi-Automaten für reguläre Sprachen aus unendlichen Spuren, eine Verallgemeinerung der Büchi-Automaten auf unendlichen Wörtern. Mit diesen neuen Automatentypen kann die Charakterisierung der erkennbaren Spur-Sprachen durch Logik monadischer zweiter Stufe auf unendliche Spuren erweitert werden.

Methoden der Programmverifikation

(Thomas Liedtke)

Endliche Interpretationen (Datenbereiche) können durch Zeichenreihen codiert werden. Das Spektrum einer partiellen Korrektheitsformel ist diejenige Sprache, die aus den Interpretationen besteht, in denen eine (vorgegebene) Formel gilt. Die Spektren aller partiellen Korrektheitsformeln, die mit Programmen aus einer Programmiersprache gebildet werden können, werden zu einer Klasse von Sprachen zusammengefaßt.

Interessant sind nun die Beziehungen zwischen Komplexitätsklassen und Spektrenklassen. Insbesondere können damit Hoarsche Beweissysteme bezüglich Ihrer Komplexität untersucht werden.

Weiterhin wird versucht, für „unkonventionelle“ Hoarsche Methoden Standardisierungseigenschaften nachzuweisen. Vielfach können dabei komplizierte Programmtransformationen helfen, komplexe Beweissysteme (i.e. Erstellung eines Ableitungsbaumes) zu vereinfachen.

Im Vordergrund sollen dabei stets die Applikativität auf algolähnliche Programmiersprachen, sowie die Komplexität bei der Automatisierbarkeit interpretationsunabhängiger Beweissysteme, betrachtet werden.

Separation der Eigenwerte von Matrizenpolynomen

(*Sven Mijderwijk*)

Ziel der momentanen Untersuchungen ist es, mit Hilfe der bei der Bestimmung eines größten gemeinsamen rechten Teilers zweier Matrizenpolynome auftretenden Größen, etwas über die Signatur der verallgemeinerten Bezoutiante dieser beiden Matrizenpolynome auszusagen. Außerdem sollen die Analogien zwischen Euklidischem Algorithmus und dem Vorgehen im Matrixfall festgestellt werden.

Die Bearbeitung des Problems hat zu einem Algorithmus geführt, der die Signatur einer s.a. verallgemeinerten Bezoutiante bestimmt. Außerdem erhält man eine Faktorisierung s.a. verallgemeinerter Bezoutianten in der Form P^*DP ohne Benutzung des Spektrums, wobei D einfache Struktur besitzt.

Der Algorithmus ist auch geeignet für die Signaturbestimmung s.a. Blockhankelmatrizen.

Die Arbeit wird von Herrn Prof. G. Heinig an der Universität Leipzig betreut.

Echtzeitdatenverarbeitung auf dem PC

(*Uwe Berger*)

Dieses Forschungsprojekt wird in Zusammenarbeit mit der Abteilung Betriebssoftware durchgeführt und ist in Abschnitt 2.1.1, Seite 14, beschrieben.

Forschungskontakte

Universität Ulm (*Köbler, Schöning, Thierauf*)

Universität Würzburg (*Buntrock, Hertrampf, Wagner*)

University of California at Santa Barbara (*Book, während seines Deutschlandaufenthalts als Humboldt-Preisträger*)

Université PARIS 6 (*Gastin*)

Université PARIS SUD, Orsay (*Petit*)

Universität Leipzig

(*Prof. Dr. W. Knödel, Prof. G. Heinig, Dr. Scheiter, Dr. Herre*)

DB Entwicklungszentrum Ulm (*Dr. B. Hohlfeld*)

2.1.5 Abteilung Intelligente Systeme

Leiter	<i>Lehmann</i>
Wiss. Mitarbeiter	<i>Burkert, Forster, Rathke, Tausend</i>
Programmierer	<i>Langjahr, Schullerer</i>
Verwaltungsangestellte	<i>Frau Castro (bis 31.3.)</i>

Wissensrepräsentation und Inferenz

Repräsentation terminologischen Wissens

(*Gerrit Burkert, Peter Forster*)

Terminologische Repräsentationsformalismen (auch Termbeschreibungssprachen genannt) basieren auf dem Wissensrepräsentationssystem KL-ONE. Ihr Schwerpunkt liegt auf der Repräsentation von definitorischem Wissen. Termbeschreibungssprachen zeichnen sich durch eine klar definierte modelltheoretische Semantik aus. Durch Inferenzmechanismen kann Wissen, das in den Beschreibungen nur implizit enthalten ist, abgeleitet werden. Die Beschreibung von Wortbedeutungen ist auch Gegenstand der Definitionen in Wörterbüchern. Aus der Untersuchung einer Reihe von Wörterbuchdefinitionen wurden Anforderungen an einen Repräsentationsformalismus abgeleitet. Dabei ist zu beachten, daß Wörterbuchdefinitionen, die für menschliche Benutzer gedacht sind, oft nur einen kleinen Teil der für die Bedeutung eines Begriffs wesentlich erscheinenden Informationen enthalten.

Es wurde ein terminologisches „Kernsystem“ TED (für term description language) mit wohldefinierter Semantik und effizienten Inferenzverfahren entworfen und implementiert. TED wurde um eine assertionale Komponente namens ALAN (assertional language) erweitert (Studienarbeit). ALAN ermöglicht es, Instanzen von Konzepten zu beschreiben und Inferenzen über solche Instanzen durchzuführen.

Um wenigstens einen Teil der Anforderungen zur Repräsentation von semantischem Wissen zu erfüllen, muß die terminologische Komponente geeignet erweitert werden. Dabei sind eine eindeutige semantische Beschreibung der Repräsentationskonstrukte und die Entwicklung korrekter und effizienter Inferenzmechanismen anzustreben. Zugunsten eines akzeptablen Antwortverhaltens muß auf die Vollständigkeit der Inferenzen verzichtet werden. Im einzelnen wurde an folgenden Erweiterungen gearbeitet:

- Erweiterung von Konzeptbeschreibungen durch Informationen die typischerweise gelten, sog. *Defaults*
- Integration eines Regelsystems in die assertionalen Komponente. Mit solchen Regeln kann u. a. Wissen repräsentiert werden, das in der terminologischen Komponente zu Unentscheidbarkeitsproblemen führen würde (z. B. inverse Rollen).

- Kopplung der terminologischen Komponente mit konkreten Weltausschnitten, z. B. Integration einer Logik über Zeitintervalle

Wichtigstes Ziel für den nächsten Forschungsabschnitt ist es, eine allgemeine Ontologie aufzubauen, mit deren Hilfe einige Aspekte von Begriffdefinitionen darstellbar sind.

Frame-Talk und Perspektiven

(*Christian Rathke*)

FrameTalk ist eine frame-basierte Repräsentationssprache, die als Erweiterung von CLOS, dem objektorientierten Teil von Common Lisp, entwickelt wird. Die Erweiterungen sollen im Sinne objektorientierter Konzepte realisiert werden, d. h. sie sollen selbst die Form von Klassen, Instanzen und Methoden haben.

Objekte in FrameTalk sollen aus „Perspektiven“ zusammengesetzt sein. Perspektiven sind Inkarnationen ein und desselben Objektes in unterschiedlichen Kontexten. So kann eine konkrete Person als Arbeitnehmer und in einem anderen Kontext als Arbeitgeber, Reisender, Musiker, etc. fungieren. Perspektiven heben einen Teil des betrachteten Objekts hervor. Sie hängen *auch* vom Betrachter ab. Darüber hinaus tragen sie zur Definition von Objekten und deren Eigenschaften bei. Objekte werden im wesentlichen als eine Menge von Ausprägungen verschiedener Perspektiven definiert. Im Berichtszeitraum wurde eine Perspektivenmechanismus für FrameTalk entworfen und implementiert (Diplomarbeiten Kerstin Drehmann und Frank Grotewall).

Termbeschreibungssprachen und Perspektiven

(*Christian Rathke*)

Die Beschreibung von Individuen im assertionalen Teil einer Termbeschreibungssprache führt i. a. zur Zuordnung mehrerer Terme des terminologischen Teils. Es liegt nun nahe, solche multiplen Beschreibungen als Perspektiven aufzufassen und das gegebene Individuum allen Beschreibungen zuzuordnen. Um solche und ähnliche Modellierungsmöglichkeiten mit Perspektiven zu untersuchen, wurde eine einfache Termbeschreibungssprache realisiert (Diplomarbeit Andreas Waidelich) und um Perspektiven ergänzt (Diplomarbeit Petra Schmidt).

Wissensbasiertes Sprachverstehen

ASK-ME — ein Auskunftssystem über Faktenwissen

(*Egbert Lehmann*)

Anknüpfend an frühere eigene Arbeiten wurde das System ASK-ME entwickelt. Dies ist ein effizient arbeitender Prototyp eines in normalem Deutsch abfragbaren Vielzweck-Auskunftssystems zur Abfrage von Fakten aus recht unterschiedlichen Wissensgebieten. Eine Reihe von Möglichkeiten zur Erweiterung dieses Systems in Richtung eines generelleren sprachverstehenden Systems wurden untersucht.

Hilfsmittel zur Analyse deutscher Texte

(*Gerrit Burkert, Egbert Lehmann*)

Das langfristige Vorhaben des Aufbaus eines generellen textverstehenden Systems wurde durch den Aufbau eines für die weitere Arbeit geeigneten Textkorpus, Ergänzung des Lexikons durch Bedeutungsdefinitionen von Nomen, eine interaktive Benutzerschnittstelle zur Bearbeitung lexikalischer Einträge, Erweiterung der Regelgrammatik und Entwurf und Erprobung von Softwaremodulen für das Indexieren und statistische Auswerten umfangreicher Texte vorangetrieben.

Expertensysteme

Constraints in einem Expertensystem zur Steuerberatung

(*Christian Rathke*)

In Fortsetzung der Arbeiten im Bereich interaktive Expertensysteme entstand ein System, das es seinen Benutzern erlaubt, interaktiv Hilfen für die Erstellung eines Teilbereichs der Steuererklärung zu erhalten (Diplomarbeit Birgit Grün). Steuerberater sind Experten, die Einkommenssteuerpflichtigen bei der Abfassung ihrer Steuererklärungen helfen. Sie können die Lebensumstände der Betroffenen, soweit sie für die Aufgabe relevant sind, erkennen, hinterfragen und die richtigen, d. h. vorteilhaften Konsequenzen für das Abfassen der Steuererklärung ziehen. Das Wissen von Steuerberatern erstreckt sich dabei in erster Linie auf entsprechende Steuergesetze. Zusätzlich fließen Erfahrungen aus dem Abfassen von Steuererklärungen in den Prozeß der Beratung ein.

Die Abhängigkeiten zwischen den Lebensumständen und Feldern in Formularen für Steuererklärungen können in erster Näherung als sog. Constraints aufgefaßt werden. Diesem Modell liegt die Vorstellung zugrunde, die Lebensumstände und die Einträge

im Formular in einer Abhängigkeitsstruktur miteinander zu verbinden. Ein Änderung der Lebensumstände (z. B. Heirat, Wechsel des Arbeitsplatzes oder der Wohnung) hat eine Veränderung der Einträge im Steuerformular zur Folge. Umgekehrt können Voraussetzungen und Konsequenzen eines Eintrags auf die Lebensumstände (z. B. ist es finanziell vorteilhaft zu heiraten?) nachgeprüft werden.

Maschinelles Lernen

Analoges und induktives Lernen von Hornklauseln

(Birgit Tausend)

Zum Lernen von Hornklauseln aus Beispielen sind im Bereich des Maschinellen Lernens sowohl induktive als auch analoge Verfahren entwickelt worden. Ziel dieser Verfahren ist es, aus einer Menge von Beispielen für ein zu lernendes Konzept und einer Menge von bekannten Prädikaten, die als Hintergrundwissen vorhanden sind, eine Definition für das neue Konzept zu bestimmen. Die neue Definition wird als Hornklausel ausgedrückt.

Induktive Verfahren gehen von einer Menge von Beispielen für ein zu lernendes Konzept aus. Aus den Beispielen wird eine induktive Hypothese gebildet, die zusammen mit dem Hintergrundwissen die Beispiele impliziert. Die Menge von Hypothesen für die neue Hornklausel spannt einen Suchraum auf, wenn die Hypothesen durch eine genereller-als-Relation angeordnet werden können. Aus diesem Suchraum muß mit Hilfe eines Präferenzkriteriums die beste Hypothese ausgewählt werden. Da die Menge der Hypothesen im allgemeinen sehr groß ist, werden verschiedene Methoden angewendet, um den Suchraum einzuschränken, beispielsweise die Beschränkung auf Hypothesen bestimmter Formen. Die erlaubten Formen können durch Schemata beschrieben werden, mit denen dann die Suche im Hypothesenraum gesteuert wird.

Beim Lernen durch Analogie wird die neue Hornklausel gelernt, indem auf bekannte Prädikate zurückgegriffen wird, die von der Form her ähnlich zu dem neuen Prädikat sind. Auf diese Weise wird der Hypothesenraum für das neue Konzept so beschränkt, daß nur Hypothesen erlaubt sind, die ähnlich zu einem bekannten Konzept sind.

Vergleicht man beide Vorgehensweisen, so lassen sich folgende Unterschiede feststellen. Bei analogen Verfahren wird das Hintergrundwissen besser ausgenutzt als bei induktiven Ansätzen, da die Information über die Form der Hintergrundprädikate zur Definition der neuen Hornklausel verwendet wird. Diese Information wird bei induktiven Verfahren nicht ausgenutzt, da die Schemata in der Regel vorgegeben sind. Bei der Verwendung von Schemata für induktive Lernverfahren muß ein Schema gefunden werden, das die neue Hornklausel exakt matcht, während bei analogen Verfahren keine exakte Übereinstimmung notwendig ist, d.h. auch partielle Matches zwischen bekannter und neuer Klausel erlaubt sind. Ein wesentlicher Nachteil analoger Verfahren ist

der große Suchraum bei der Auswahl eines ähnlichen Prädikats, da alle Hintergrundklauseln betrachtet werden, während bei induktiven Verfahren die Schemata meist viele Klauseln subsumieren.

Zur Integration von induktivem und analogen Vorgehen wurde ein Verfahren CAN mit folgenden Eigenschaften entwickelt und teilweise implementiert. Zur Beschränkung des Suchraums verwendet CAN Schemata, die flexibel angelegt sind, so daß sie viele Prädikate subsumieren. Vorläufige Hypothesen werden gebildet, indem die Beispielmengen anhand der Struktur der Argumentterme unterteilt und für jede der Teilmengen eine Hypothese gebildet wird (Diplomarbeit Irene Stahl). Diese Aufteilung der ursprünglichen Beispielmenge wird durchgeführt, bevor mit Hilfe der Schemata die neuen Hornklauseln gelernt werden. Die Schemata werden danach schrittweise instanziert, bis eine neue Hornklausel aufgebaut wurde, die zusammen mit dem Hintergrundwissen die Beispiele impliziert. Während des Lernens kann ein Schema für das neue Prädikat angepaßt werden, indem es mit vordefinierten Modifikationsoperatoren geändert wird. Die Operatoren nehmen geringfügige Veränderungen an einem Schema vor, wie z. B. das Einfügen eines neuen Literals.

Das entwickelte Verfahren CAN hat sowohl Eigenschaften analoger als auch induktiver Verfahren. Im Vergleich zu analogen Verfahren wird die Suche nach ähnlichen bekannten Konzepten dadurch vereinfacht, daß die Hintergrundkonzepte in Schemata zusammengefaßt werden. Die Schemata können von CAN automatisch aus einer gegebenen Wissensbasis extrahiert werden. Da die Schemata flexibel sind und auch modifiziert werden können, wird eine neue Hornklausel gelernt, die ähnlich zu den bekannten Hintergrundklauseln ist, die das Schema matchen. Von den induktiven Verfahren wurde die gezielte Suche durch den durch Schemata beschränkten Hypothesenraum übernommen. Den induktiven Verfahren gegenüber hat CAN aber die Vorteile, daß auch gelernt werden kann, wenn ein Schema nicht exakt für die neue Hornklausel paßt und daß die vorhandene Wissensbasis durch die automatische Extraktion von Schemata besser ausgenutzt wird.

CAN soll in der nächsten Zeit besonders im Bereich der Modifikation von Schemata verbessert werden und mit verschiedenen Anwendungen getestet werden.

Forschungkontakte

Computer Science Department, Boulder, Colorado, USA (*Fischer, Lewis*)

Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI)
(*Barth, Wahlster, Richter*)

Forschungsinstitut für Anwendungsorientierte Wissensverarbeitung (FAW), Ulm
(*Rösner, Wirth, Rieckert*)

IBM Deutschland, Institut für Wissensbasierte Systeme (*Bosch, v. Luck, Novak*)

Siemens AG, Zentralbereich Forschung und Entwicklung,
Fachgebiet Informationstechnische Grundlagen
(*Schütt, Büttner, Block*)

Institut für Maschinelle Sprachverarbeitung der Universität Stuttgart
(*Rohrer, Kamp, Haider*)

Gesellschaft für Mathematik und Datenverarbeitung (GMD), Darmstadt
(*Wanner, Maier, Böcker*)

Universität Erlangen / Nürnberg, Informatik (*Stoyan, Görz*)

Polnische Akademie der Wissenschaften, Warschau (*L. Bolc*)

AND Software B.V., Rotterdam (*van Wessel*)

Vrije Universiteit Amsterdam (*Willy Martin*)

GMD, Bonn (*Wrobel, Kietz*)

Universität Dortmund (*Monk*)

2.1.6 Abteilung Programmiersprachen und ihre Übersetzer

Leiter	<i>Ludewig</i> (kommissarisch bis 31.3. und ab 1.8.)
	<i>Sebesta</i> (Lehrstuhlvertreter vom 1.4. bis 31.7.)
Wiss. Mitarbeiter	<i>Welsch</i> (bis 31.1.)
Programmierer	<i>Kübler, Jenke</i> (ab 1.3.)
Verwaltungsangestellte	<i>Günthör</i>

2.1.7 Abteilung Software Engineering

Leiter	<i>Ludewig</i>
Wiss. Mitarbeiter	<i>Bassler (ab 1.10.), Deininger, Lichter, K. Schneider, Schwille</i>
Programmierer	<i>Georgescu, M. Schneider</i>
Verwaltungsangestellte	<i>Günthör</i>

Projekt: Objektorientierte Software-Entwicklung und Prototyping

Ziel dieser Arbeit ist es, methodische und instrumentelle Voraussetzungen zu schaffen, damit sogenannte Software-Prototypen (besser: Attrappen) mit möglichst geringem Aufwand entwickelt, erprobt, modifiziert und in Zielsysteme überführt werden können. Dabei wird ein Ansatz verfolgt, das System mit objektorientierten Ansätzen zunächst nur grob zu modellieren (Architektur-Prototyping) und das Modell, wenn es stabil geworden ist, sukzessive in konventionellen Zielcode zu überführen (schrittweise Komplettierung).

Im Rahmen dieser Arbeit wurde ein Kernsystem entwickelt, mit dem Software-Architekturen interaktiv erstellt werden können. Spezielle Werkzeuge erlauben es, einen Architektur-Prototyp auszuführen, der gleichzeitig aus Bausteinen besteht, die in der Prototypingssprache Smalltalk-80 oder in der Zielsprache Eiffel realisiert sind. Es ist geplant, für die gesamte Informationsverwaltung eine objektorientierte Datenbank einzusetzen.

Projekt: Software-Prozeß-Modell

Im Rahmen dieses Forschungsprojekts wird ein Simulationspaket in Form eines Computerspiels (SESAM = Software-Engineering-Simulation durch animierte Modelle) entwickelt. Wie bei äußerlich ähnlichen Programmen (z.B. Ökolopoly von F. Vester) spielt eine Person oder eine Gruppe gegen den Rechner, der ein vernetztes, damit schwer überschaubares System simuliert. Hier übernimmt der Spieler im Rahmen eines (simulierten) Software-Projekts die Rolle des Projektleiters. Ziel des Spiels ist es, das Projekt erfolgreich durchzuführen und abzuschließen.

Eine zentrale Rolle spielen die Annahmen über Einflüsse und Wechselwirkungen im Projekt, die hier als Hypothesen bezeichnet werden. Ein Beispiel ist die Feststellung (nach Brooks), daß Neueinstellungen ein laufendes Projekt u.U. nicht beschleunigen, sondern verzögern. Solche Hypothesen müssen für SESAM gesammelt, formuliert und

so integriert werden, daß sie einerseits sehr unterschiedliche Wirkungen haben können, sich aber dennoch leicht ändern oder ersetzen lassen. Die Sammlung und Formulierung der Hypothesen war eine wichtige Aktivität im Jahre 1991.

Heute steht eine Realisierung zur Verfügung, die erstmals nicht mehr den Charakter eines (später wegzuwerfenden) Prototyps, sondern den eines in Zukunft auszubauenden Kernsystems hat.

Projekt: Software Metriken

Im Rahmen einer Industrie-Kooperation sollen Software-Metriken in einem realen Projekt eingesetzt werden.

Zur Bewertung von Software und dem Prozeß, der sie hervorbringt, gibt es eine Vielzahl theoretischer Ansätze. Ziel dieser Kooperation ist es, einige dieser Ansätze auszuwählen und auf ihren tatsächlichen Nutzen in der Praxis zu überprüfen.

Die Kooperation wurde im Dezember 1991 begonnen.

Forschungskontakte

GMD St. Augustin (*R. Budde u.a.*)

ETH Zürich (*N. Wirth, C.A. Zehnder*)

Universität Mainz (*F. Wankmüller*)

ABB Heidelberg (*Chr. Welsch, J. Heger*)

Zahlreiche Anbieter von Software-Werkzeugen (CASE-Tools)

weitere Kontakte im Rahmen des GI-FA 4.3 (Requirements Engineering)
und der GI-FG 2.1.1 (Software Engineering)

2.1.8 Abteilung Theoretische Informatik

Leiter	<i>Knödel (kommissarisch bis 31.7.91), Diekert (ab 1.8.91)</i>
Mitarbeiter	<i>Münchow, Muscholl (ab 1.10.91), Reinhardt, Reuß, Volkert (ab 1.12.91)</i>
Hilfskräfte	<i>Drehmann, Ehlert, Faulstich, Herrmann, Ketelhut, Schmalzl, Schreiner, Schumacher, Sing, Stahl, Weberruß, Zink</i>

Übersicht

Der Lehrstuhl Theoretische Informatik wurde nach über fünfjähriger Vakanz im August 91 besetzt. Der damit verbundene Neuaufbau ist naturgemäß noch nicht abgeschlossen. Dies spiegelt sich sowohl in der gegenwärtigen Personalstruktur wie in den Forschungsvorhaben wider. Erschwerend für die Situation ist, daß fast gleichzeitig mit der Besetzung des Lehrstuhls der andere Theorielehrstuhl durch die Emeritierung Knödel vakant wurde. Die Besetzung dieses Lehrstuhls jetzt mit der Ausrichtung *Formale Konzepte* wird dringend erwartet. Insbesondere ist geplant, zukünftige Forschungsvorhaben in beiden Theorieabteilungen eng abzustimmen.

Forschungsprojekt: Komplexe Spurssprachen

(Volker Diekert)

Unendliche Spuren bilden ein mathematisches Modell, um die Nebenläufigkeit nicht notwendig terminierender Prozesse zu beschreiben und zu untersuchen. Im Gegensatz zu sequentiellen Prozessen reicht die Information über einen Prozeß, ob Termination vorliegt, jedoch nicht aus, um sein Verhalten geeignet zu beschreiben. Unsere Untersuchungen haben ergeben, daß die zu erwartenden zukünftigen Abhängigkeiten ein unverzichtbares Beschreibungsmerkmal eines Prozesses sind. Diese Zusatzinformation reduziert sich in dem Modell der unendlichen Spuren auf die Angabe der Teilmenge eines gewissen Grundalphabets und führt auf den Begriff komplexer Spuren.

Um diese neuen Strukturen praktisch einsetzen zu können, bedarf es einer Theorie erkennbarer und regulärer Spurssprachen. Die Grundfragen dieser Theorie konnten bereits geklärt werden, so konnte etwa eine Verallgemeinerung des Kleene-Ochmanskischen Satzes gefunden werden, der von zentraler Bedeutung ist. Schwierig ist weiterhin die Beschreibung erkennbarer Spurssprachen durch geeignete Automatenmodelle. Insbesondere ist die Frage deterministischer Spurautomaten offen.

Forschungsprojekt: Spurersetzungssysteme

(Volker Diekert)

Ersetzungssysteme sind an vielen Stellen der Informatik gegenwärtig. Insbesondere Termersetzungssysteme werden in der Algebraischen Spezifikation, Semantik von Programmiersprachen, zum Theorembeweisen oder in der Logikprogrammierung verwendet. In dem Forschungsvorhaben werden Spurersetzungssysteme untersucht. Diese Systeme verallgemeinern Wort und Vektorersetzungssysteme, indem sie eine direkte Behandlung gewisser Nebenläufigkeiten ermöglichen und können als Spezialfall von Termersetzungen modulo einer Kongruenz gelesen werden. Daraus ergeben sich viele natürliche und interessante Problemstellungen. Von zentraler Bedeutung ist hier die Frage nach der Entscheidbarkeit der Konfluenz. Es ist bekannt, daß diese Frage im allgemeinen algorithmisch nicht zu beantworten ist. Im Falle von Einregelsystemen könnte die Entscheidbarkeit bis auf wenige Ausnahmen jedoch positiv geklärt werden. Die Frage der Termination bleibt (selbst für Einregel-Wortersetzungssysteme) offen.

Forschungsprojekt: Inhärent kontext sensitive Sprachen

(Volker Diekert)

Ausgehend vom Konzept der Komplexitätskerne für Probleme, die nicht in polynomialer Zeit zu entscheiden sind, wird ein formalsprachliches Analagon über Platzkomplexitätsklassen entwickelt. Betrachtet man eine kontextsensitive Sprache L , die nicht kontextfrei ist, so wird die Existenz einer unendlichen kontextsensitiven Teilsprache H gezeigt, so, daß $H \cap K$ endlich ist, wenn K eine kontextfreie Teilsprache von L ist. Eine Übertragung auf kontextfreie Sprachen ist eng mit dem Begriff von IRS-Sprachen von S. Greibach verknüpft.

Forschungsprojekt: Konfluente Semikommation und Komplexität

(Volker Diekert, Klaus Reinhardt)

In Fortsetzung unserer Arbeit vom letzten Jahr konnten wir einen Zusammenhang zwischen der lokalen Überprüfbarkeit der Synchronisierbarkeit von Spuren und der Konfluenz von Semikommationssystemen beschreiben. Dadurch wird gezeigt, daß ihre Entscheidung ebenfalls $co-NP$ vollständig ist.

Wir betrachten endliche Semikommationssysteme als Unterklasse von Spur-Ersetzungssystemen (eine Verallgemeinerung von Semi-Thue-Systemen) und konnten zeigen, daß deren Konfluenz nun entscheidbar ist, indem wir ein graphentheoretisches

Kriterium entwickelten. Damit konnten wir durch eine Graphkonstruktion auch zeigen, daß das Problem der Existenz endlicher, vollständiger Präsentationen zwischen Spurmonoiden, die sich auf die Existenz eines vollständigen Semikommunikationssystems reduzieren lässt, Σ_2^p -vollständig ist und die Entscheidung der Konfluenz eines gegebenen Semikommunikationssystems $co-NP$ vollständig ist.

Forschungsprojekt: Charakterisierungen von Komplexitätsklassen mittels Monoid- und Gruppoidprogrammen

(Anca Muscholl)

Programme über Monoide stellen ein Berechnungsmodell dar, das in den letzten Jahren interessante Charakterisierungen von parallelen Komplexitätsklassen ($AC^0 \subset ACC^0(q) \subseteq NC^1$) ergeben hat, deren Definition auf Schaltnetze basiert. Dieser Ansatz stellt ein Mittel zur Verfügung, um die offenen Beziehungen zwischen Unterklassen von NC^1 (die als Klasse der effizient parallelisierbaren Problemen bezeichnet werden kann) mit algebraischen Mitteln neu in Angriff zu nehmen.

Die Verwendung von Gruppoiden als grundlegende Struktur hat zur Charakterisierung der Klasse $LOGCFL$ geführt, d.h. der Klasse der Probleme, die sich mit logarithmischen Platz auf eine kontext-freie Sprache reduzieren lassen. Ausgehend von diesem Ergebnis, konnten weitere sequentielle Komplexitätsklassen (LOG , $NLOG$, $LOGDCFL$, NP) in diesem Rahmen charakterisiert werden, wodurch Fragen beantwortet wurden, die bei der Charakterisierung von $LOGCFL$ entstanden waren. Die Resultate wurden bei STRUCTURE 92 eingereicht.

Forschungsprojekt: Kommunikationskonzepte in Verteilten Systemen

(Joachim Münchow)

Der große Nachteil statischer Routing-Algorithmen liegt im Fehlen jeglicher Adaptivität. In der Vergangenheit wurden zahlreiche Verfahren vorgestellt, die den statischen Ansatz und die Berücksichtigung von Auslastungsschwankungen miteinander verbanden. Bedauerlicherweise implizierten derartige Ansätze eine Reihe von Nachteilen, die in besonderem Maße die Stabilität der Netze betrafen.

Als Beispiel werde hier das zentrale Routing genannt, dessen Nachteile in der Verwundbarkeit der Zentraleinheit, der Heterogenität bezüglich der Verbindungsmehrbelastung, besonders aber im hohen zusätzlichen Kommunikationsaufwand für die Übertragung der Auslastungsinformationen und Routingtafelsätze liegen.

Erarbeitet wurde ein Adaptivierungsmechanismus für statisches Routing, basierend auf dem Algorithmus für die Wegstreckenoptimierung in belasteten Netzen (siehe Jahresbericht 90). Um Applikabilität aufzuzeigen war unterschiedlichen Fragestellungen

nachzugehen. So etwa die Behandlung bzw. Eliminierung von Pfadmehrdeutigkeiten, die Bindung von Informationen hinsichtlich der Auslastungsstufen und die Reaktion auf den Ausfall von Hardware-Ressourcen.

Forschungsprojekt: Synchronisation von Semi-Spuren

(Klaus Reinhardt)

Betreiben nebenläufige Prozesse Kommunikation miteinander, so ist es nötig diese zu synchronisieren; sind diese durch Semi-Spuren in einem Semikommunikationssystem beschrieben, so wird dies durch den Schnitt der Semi-Spuren ausgedrückt. Dies ermöglicht die modulare Beschreibung nebenläufiger Systeme. Die Schnittabgeschlossenheit der Semi-Spuren in einem Semikommunikationssystems und die Synchronisationsabgeschlossenheit im allgemeinen Fall können wir anhand eines Graphkriteriums entscheiden, das $co-NP$ vollständig ist, wohingegen die Synchronisierbarkeit nur $NLOGSPACE$ vollständig ist. Unter Ausnutzung der Tatsache, daß die Inklusion von Semi-Spuren in TC^0 entscheidbar ist, kann man die Synchronisierbarkeit auch in TC^0 testen, falls eine lokale Überprüfungsbedingung gilt, die selbst zu testen jedoch wiederum $co-NP$ vollständig ist.

Forschungsprojekt: Schnelles in-place Sortieren

(Klaus Reinhardt)

Sortierverfahren wie Merge-sort, Merge-Insertion oder Insertion-sort, die mit $n \log n + O(n)$ Vergleichen auskommen, benötigen entweder $O(n^2)$ Transporte oder arbeiten mit einer Zeigerstruktur und benötigen somit $2n$ Platz. Bei einem *in-place* Sortierverfahren von Huang und Langston steigt die Anzahl der Vergleiche auf $O(n \log n)$. Heap-sort braucht $2n \log n$ Vergleiche, Wegener konnte kürzlich zeigen, dass Mc Diarmid und Reed's Variante von Bottom-up-Heap-sort $n \log n + 1.1n$ Vergleiche benötigt.

Wir fanden zunächst ein neue Variante von Merge-sort, die mit $1.25n$ Platz auskommt, indem freiwerdender Platz bereits im gleichen Sortierdurchgang wiederbenutzt wird. Hierbei kommt man mit $n \log n - 0.913n$ Vergleichen aus. Sortiert man zuvor jeweils Blöcke der Länge 21 mit Merge-Insertion, was nur linear mehr Transporte kostet, läßt sich dies auf $n \log n - 1.211n$ verbessern, was der informationstheoretischen Unterschranke von $n \log n - 1.443n$ schon recht nahe kommt. Würde man Blöcke der Länge 85, 341, 1365, ... verwenden, könnte man den Faktor theoretisch bis auf $1 - \log(\frac{2 \ln 2}{3}) - \frac{1}{\ln 2} = -1.328966$ verbessern.

Durch Zusammensortieren von 2^k Teillisten in einem Schritt lässt sich die Anzahl der Transporte um jeden beliebigen Faktor k und somit auf $\epsilon n \log n + O(1)$ für alle $\epsilon > 0$ verringern.

Mit Hilfe eines Merge-Algorithmus , der Listen der Länge n und m mit $\lfloor \frac{n}{2^k} \rfloor + (k+1)m - 1$ Vergleichen zusammensortiert, konstruierten wir ein iteratives in-place Verfahren, das jeweils einen Teil der unsortierten Elemente sortiert und mit den bereits sortierten Elementen zusammensortiert. Unsortierte Elemente treten dabei an die Stelle des freien Platzes. Dieses Verfahren kostet nur $0.0785n$ Vergleiche mehr, womit -1.1324 (theoretisch -1.2504) als linearer Faktor erreicht werden kann.

Da logarithmisch viele Iterationsschritte nötig sind, würden $O(n \log n)$ Transporte zusätzlich benötigt, dies lässt sich jedoch auf $O(n)$ reduzieren, indem man nach jeweils konstanter Anzahl von Iterationsschritten eine Quicksortiteration auf den unsortierten Elementen bezüglich des mittleren Elementes der sortierten Liste durchführt.

Forschungkontakte

University of California, Santa Barbara, USA (*Ron V. Book, C. Wrathall*)

Universität Leiden, Niederlande (*H.-J. Hoogeboom, G. Rozenberg*)

Université de Paris 6, Frankreich (*P. Gastin, J. Sakarovitch*)

Université de Paris Sud, Frankreich (*A. Petit*)

Université de Bordeaux I, Frankreich (*Y. Métivier, R. Cori*)

Université de Lille, Frankreich

(*M. Clerbout, D. Gonzalez, M. Latteux, Y. Roos, P. A. Wacrenier*)

University of Milano, Italien (*G. Mauri, N. Sabadini*)

Universität Barcelona, Spanien (*C. Alvarez*)

Polnische Akademie der Wissenschaften in Warschau, Polen

(*A. Mazurkiewicz, E. Ochmanski*)

Universität Lund, Schweden (*O. Peterson*)

Universität Kiel (*W. Thomas*)

Universität Ulm

(*J. Köbler, M. Mundhenk, U. Schöning, R. Schuller, Th. Thierauf*)

Technische Universität München

(*B. Jenner, K.-J. Lange, I. Niepel, P. Rossmanith*)

Universität Würzburg (*G. Buntrock, K. Wagner*)

Humboldt-Universität Berlin (*C. Damm*)

2.2 Veröffentlichungen

- Appelrath, H.J.** *Skriptum Informatik – eine konventionelle Einführung*
Ludewig, J. vdf, Zürich, und Teubner, Stuttgart, 1991
- Bauer, D.** *Definition von Vorgängen mit dem Editor VED*
Herberg, H. v. d. In Lutze, Kohl (Hrsg.): Wissensbasierte Systeme im
Maier, D. Büro,
Rathke, M. Oldenbourg Verlag, München Wien, 1991
Ressel, M.
Schwab, Th.
- Burkert, G.** 1) *Representation of Semantic Knowledge with Term Subsumption Languages*
Forster, P. Proceedings of the ACL Workshop “Lexical Semantics and Knowledge Representation”
Berkeley, California, USA, 1991
- Burkert, G.** 2) *siehe auch Forster, P.*
- Burkhardt, W.** *Performance Penalty by Communication in Multiprocessor Systems*
Proceedings 1991 International Conference on Parallel Processing
St. Charles, Il, USA, August 1991
- Diekert, V.** 1) *On confluent semi-commutations-decidability and complexity results*
Ochmanski, E. Proc. of 18th ICALP, LNCS, 510, Springer 1991,
Reinhardt, K. 229-241.
- Diekert, V.** 2) *Recognizable complex trace languages*
Gastin, P. Proc. of 16th Symposium on MFCS’91, Springer 1991,
Petit, A. 131-140
- Diekert, V.** 3) *On “inherently context sensitive” languages.
An application of complexity cores*
Book, R. V. Inform. Proc. Letters, 1991
- Forster, P.** 1) *Term Subsumption Languages and Lexical Semantics*
Burkert, G. Proceedings of the Terminological Logic Users Workshop,
KIT Project
Technische Universität Berlin, 1991

2) *siehe auch Burkert, G.*

Frühauf, K.
Ludewig, J.
Sandmayr, H.

Software-Prüfung – eine Fibel
vdf, Zürich, und Teubner, Stuttgart, 1991

Gunzenhäuser, R.

- 1) *Dialogmodelle und Benutzermodelle – neuere Aspekte und Anwendungen der wissensbasierten MCK in:*
Handbuch der Modernen Datenverarbeitung, Heft 160
Forkel-Verlag, Wiesbaden, 1991
- 2) *Mensch-Computer-Kommunikation (Einführung) und Auswirkungen des Verbundprojekts WISDOM auf die Grundlagenforschung und Lehre an einer Hochschule* (gemeinsam mit H.D. Böcker) in:
Wissensbasierte Systeme im Büro
Oldenburg-Verlag, München, 1991
- 3) *Computerunterstütztes Lernen für die Aus- und Weiterbildung*
In Office Management, Heft 9/1991, Baden-Baden

Hanakata, K.
Yokoyama, S.

Construction of abstract conceptual network based on definition sentences in dictionaries
Computational Linguistics in Japan, Ed. Ishiwata,
1991

Herczeg, J.
Hohl, H.

Building Browsers for the Common Lisp Object System
In: Proceedings of EastEurOOPe'91, Conference, Tutorials and Exhibition on Object-Oriented Programming, Bratislava, CSFR, September 1991.

Herczeg, J.
Hohl, H.
Ressel, M.

HyperQuery – Ein Anfragesystem mit Graphischer Benutzeroberfläche
In Fuhr (Hrsg.): Information Retrieval, GI/GMD Workshop, Darmstadt, Juni 1991, Informatik-Fachberichte 289, Springer-Verlag

Herczeg, J.
Hohl, H.
Schwab, Th.

XIT – A Multi-Layered Tool for User Interface Design
In: Proceedings of the Fourth International Conference on Human-Computer Interaction, Stuttgart, Elsevier, Amsterdam - London - New York - Tokyo, 1991

Hohl, H.

siehe Herczeg, J.

Kieback, A. *Prototyping in industriellen Software-Projekten — Erfahrungen und Analysen*
Lichter, H.
Schneider-
Hufschmidt, M.
Züllighoven, H.

GMD-Studie Nr. 184, 1991

Lichter, H. *siehe Kieback, A.*

- Ludewig, J.**
- 1) *Wie gut ist die Software? Qualitäts- und Komplexitätsmetriken, subjektive Schätzungen*
Technische Rundschau 83, Heft 3, 46-53, 1991
 - 2) *Software Engineering und CASE – Begriffsklärung und Standortbestimmung*
it (Informationstechnik), 33 (3), 112-120, 1991
 - 3) *Software Engineering in der Praxis: Voraussetzungen für Fortschritte*
Technische Rundschau 83 , Heft 33, 26-31, 1991
 - 4) *siehe auch Appelrath, H.J.*
 - 5) *siehe auch Frühauf, K.*

Ludewig, J. (Hrsg.) *Software- und Automatisierungsprojekte – Beispiele aus der Praxis*
Teubner Stuttgart, 1991

Mackamul, H. *Intelligente tutorielle Systeme für die musikalische Ausbildung*
Mahling, A.
Studioblätter des Studios für Elektronische und Computer Musik der Staatlichen Hochschule für Musik und Darstellende Kunst, Stuttgart, (6):4-37, 1991

- Mahling, A.**
- 1) *How to Feed Musical Gestures into Compositions*
In: Proceedings of the International Computer Music Conference 1991, S. 258-265, Montreal, 1991. Faculty of Music McGill University
 - 2) *siehe auch Mackamul, H.*

Maier, D. *siehe Bauer, D.*

Rathke, Ch.

- 1) *Implementing Frames in an Object-oriented Programming Language*
Proceedings of the First East European Conference on Object-Oriented Programming, pp 88-94
Bratislava, CSFR, September 1991

- 2) *A Construction Kit Approach to Knowledge-based Design*
Proceedings of the Forth International Symposium on Artificial Intelligence, pp 259-268
Cancun, Mexico, November 1991

**Rathke, Ch.
König, R.**

Combining Rules and State Space Objects in a Configuration Expert System
Proceedings of the 7th IEEE Conference on Artificial Intelligence Applications, pp 275-279
Miami, USA, Februar 1991

**Rathke, Ch.
Negele, A.**

- 1) *Konex+: A Knowledge-based Design Environment*
Proceedings of the 11th International Workshop on Expert Systems and their Applications, pp 147-159
Avignon, Frankreich, Mai 1991

- 2) *Konex+: An Interactive Design Expert*
Proceedings of the 4th International Conference on Human-Computer Interaction, pp 844-848
Stuttgart, September 1991

Reinhardt, K.

siehe Diekert, V.

Ressel, M.

siehe Bauer, D.

Schneider, K.

Systematische Evaluierung von CASE-Tools
Tagungsband zur TOOL 91, 27.-28. Nov. 1991,
Karlsruhe; vde-Verlag
Berlin, 1991

**Tausend, B.
Bell, S.**

- 1) *Analogical Reasoning for Logic Programming*
Proc. of First International Workshop on Inductive Logic Programming ILP-91
Viana de Castelo, Portugal, März 1991
-

- 2) *Analogical Reasoning for Logic Programming*
Proceedings of European Working Session on Learning
EWSL-91
Porto, Portugal, 1991

Walter, J. *Progress towards the Implementation of RAPiD on a Multi-Transputer Architecture*
Proceedings Transputing '91 Conference
Mountain View, CA, USA, April 1991

- Weber, G.**
- 1) *Interaktionsformen neuerer Eingabegeräte für die Graphikanimation*
In: P. Lorenz, Chr. Klöditz (Hrsg.) 3. Fachtagung Computeranimation Magdeburg, S. 70-82, 1991
- 2) *Concurrent Gestures*
In Quéinnec, Y. und Daniellou, F. (Hrsg.) Designing for Everyone
Taylor&Francis: London, S. 701-703, 1991

Weber, G.
Stephanidis, C.
et. al. *Access to graphical user interfaces for blind people*
Royal National Institute for the Blind: London,
68 Seiten, 1991

2.3 Berichte

Burkert, G.

siehe Forster, P.

Forster, P.

Wissensrepräsentation mit TED und ALAN

Burkert, G.

Interner Arbeitsbericht, Institut für Informatik

Eck, O.

Universität Stuttgart

Hanakata, K.

1) *COOL reference Manual*

2) *Quick guide to COOL Browser*

Liedtke, Th.

1) *3-Schichten-Verbindungen*

Universität Stuttgart, Bericht der Fakultät Informatik,
Nr. 6/91

2) Skript zur Vorlesung *Programmverifikation*

3) Skript zur Vorlesung *Programmtransformationen*

Stahl, I.

1) *General-to-specific Learning of Horn Clauses from Positive Examples*

Universität Stuttgart, Bericht der Fakultät Informatik,
Nr. 8/91

2) *Induction of Disjunctive Concepts via Partitioning the Example Set*

Universität Stuttgart, Bericht der Fakultät Informatik,
Nr. 9/91

Tausend, B.

siehe Stahl, I.

Walter, J.

Skript zur Vorlesung *Rechnerarchitektur, Teil 2*

2.4 Vorträge

Burkert, G.

- 1) *Representation of Semantic Knowledge with Term Subsumption Languages*
ACL Workshop “Lexical Semantics and Knowledge Representation”
Berkeley, California, USA, Juni 1991
- 2) *Repräsentation semantischen Wissens mit Termbeschreibungssprachen*
GWAI-91 Workshop “Wissensrepräsentation in natürlichsprachlichen Systemen”
Bonn, September 1991

Diekert, V.

- Infinite Traces*
REX-Concurency Day
Leiden (Niederlande), September 1991

Forster, P.

- 1) *Term Subsumption Languages and Lexical Semantics*
Terminological Logic Users Workshop
Berlin, Oktober 1991
- 2) *Representation of Semantic Knowledge with Term Subsumption Languages*
DFKI
Kaiserslautern, August 1991

Gunzenhäuser, R.

- 1) *Software-Ergonomie*
Eröffnungsreferat eines Lehrgangs von IBM Deutschland GmbH
Labor Böblingen, März und September 1991
 - 2) *Datenstrukturen und Algorithmen*
Seminar IBM Deutschland GmbH
Hauptverwaltung Stuttgart, März und September 1991
 - 3) *Lehrgang Software-Ergonomie*
Lehrgangsleitung und Referate
Technische Akademie Esslingen, Mai 1991
 - 4) *Vom Computer zur Informatik*
Festvortrag Schelztor Gymnasium Esslingen
Mai 1991
-

- 5) *Dialogmodellierung bei interaktiven Systemen*
Universität Bamberg, Mai 1991
- 6) *Neuere Entwicklungen der wissensbasierten Mensch-Computer-Kommunikation*
Elektrotechnisches Kolloquium
Universität Stuttgart, Juli 1991
- 7) *Wissensbasierte graphische Benutzungsoberflächen*
Informatik-Kolloquium Techn. Universität Dresden
September 1991
- 8) *Zum Stand der Informatikausbildung in Deutschland*
Arbeitskreis Informatik und Ausbildung
Universität Stuttgart, Oktober 1991
- 9) *Neuere Entwicklungen des rechnerunterstützten Lernens*
Informatik-Kolloquium der Humboldt-Universität und
der Freien Universität Berlin
Oktober 1991
- 10) *Knowledge based human-computer interaction*
Workshop Concerted Action: “Technology and blind-
ness”
Universität Stuttgart, Dezember 1991

Hanakata, K.

Object oriented programming in mechanical engineering
Laboratory of Matsushita Electric Work
Osaka, 11.9.1991

Herczeg, J.

- 1) *XIT – A Multi-Layered Tool for User Interface Design*
HCI International '91, Fourth International Conference
on Human-Computer Interaction
Stuttgart, September 1991
 - 2) *A Design Environment for Graphical User Interfaces*
NATO Advanced Research Workshop on User-Centred
Requirements for Software Engineering Environments
Chateau de Bonas (Toulouse), Frankreich, September
1991
 - 3) *siehe auch Rathke, Ch.*
-

- Herczeg, J.** *An Introduction to the Common Lisp Object System*
Hohl, H. Tutorial at the EastEurOOPe'91 Conference on
Rathke, C. Object-Oriented Programming
Bratislava, CSFR, September 1991
- Hohl, H.**
- 1) *Ein graphik-orientiertes Werkzeug zum Browsing in Objektstrukturen*
Eingeladener Vortrag zum Kolloquium des GMD-IPSI
Darmstadt, Februar 1991
 - 2) *Building Browsers for the Common Lisp Object System*
EastEurOOPe'91, Conference on Object-Oriented Programming
Bratislava, CSFR, September 1991
 - 3) *siehe auch Herczeg, J.*
 - 4) *siehe auch Rathke, Ch.*
- Lehmann, E.** *Inhaltsorientiertes Sprachverstehen — methodische Ansätze und Probleme*
Elektrotechnisches Kolloquium
Universität Stuttgart, Juni 1991
- Lichter, H.**
- 1) *Prototyping in industriellen Software-Projekten*
Workshop Benutzer-orientierte Systemmodellierung,
Marburg, 11.-12. April 1991
 - 2) *Ergebnisse des AK Wissensbasierte Systeme für Prototyping*
Mitgliederversammlung der GI FG 4.3.1
Berlin, 17.-18. September 1991
- Ludewig, J.**
- 1) *Software Engineering als Lehr- und Forschungsgebiet an der Universität Stuttgart*
Vortrag im Elektrotechnischen Kolloquium der Universität Stuttgart
 - 2) *Software Engineering: Die wichtigsten Grundlagen. Software-Prüfung durch Tests*
Beiträge zur 2. TR-Werkstatt Software Engineering,
Thun, Schweiz, 10.-11. Oktober 1991

- 3) *Vorträge, Seminare und Tutorien in der Industrie über Software Engineering, Datenstrukturen und Algorithmen, Programmiersprachen, CASE, Objektorientierte Programmierung, Software-Metriken*

Mahling, A.

- 1) *Wissensbasierte Manipulation musikalische Gestalten*
KlangArt
Osnabrück, Mai 1991
- 2) *How to Feed Musical Gestures into Compositions*
International Computer Music Conference 1991
Montreal, Canada, Oktober 1991

Rathke, Ch.

- 1) *Konex+: A Knowledge-based Design Environment*
Computer Science Department, University of Colorado
Boulder, USA, März 1991
 - 2) *Konex+: A Knowledge-based Design Environment*
11th International Workshop on Expert Systems and
their Applications
Avignon, Frankreich, Mai 1991
 - 3) *Konex+: An Interactive Design Expert*
4th International Conference on Human-Computer
Interaction
Stuttgart, September 1991
 - 4) *Implementing Frames in an Object-oriented Program-
ming Language*
First East European Conference on Object-Oriented
Programming
Bratislava, CSFR, September 1991
 - 5) *Wissensbasiertes Entwerfen mit Software-Baukästen*
Kolloquiumsvortrag, FAW
Ulm, Oktober 1991
-

- 6) *A Construction Kit Approach to Knowledge-based Design*

Forth International Symposium on Artificial Intelligence
Cancun, Mexico, November 1991

- 7) *siehe auch Herczeg, J.*

Rathke, Ch.

CLOS-Tutorial

Herczeg, J.

First East European Conference on Object-Oriented

Hohl, H.

Programming

Bratislava, CSFR, September 1991

Reinhardt, K.

- 1) *On confluent semi-commutations-decidability and complexity results*

18. ICALP

Madrid, Spanien, 9. Juli 91

- 2) *Sorting in-place with a worst case complexity of*

$n \log n - 1.2n + O(\log n)$ *comparisons and* $\varepsilon n \log n + O(1)$ *transports*

Ulm-Workshop

Ulm, 1. Oktober 91

- 3) *Ein in-place Sortierverfahren mit im worst case*

$n \log n - 1.1n + O(\log n)$ *Vergleichen und* $\epsilon n \log n + O(1)$

Transporten

15. Workshop Komplexitätstheorie und effiziente Algorithmen

Würzburg, 29. Oktober 91

- 4) *On the synchronization of semitraces*

Lille, Frankreich, 18. November 91

- 5) *Sorting in-place with a worst case complexity of*

$n \log n - 1.2n + O(\log n)$ *comparisons and* $\varepsilon n \log n + O(1)$ *transports*

Lille, Frankreich, 21. November 91

Ressel, M.

- 1) *HyperQuery – Ein Anfragesystem mit Graphischer Benutzeroberfläche*

GI/GMD Workshop on Information Retrieval

Darmstadt, Juni 1991

- 2) *A Framework for Modelling Dialogues in Interactive Systems*
HCI International '91, Fourth International Conference
on Human-Computer Interaction
Stuttgart, September 1991

Schneider, K. *Systematische Evaluierung von CASE-Tools*
Vortrag auf der TOOL 91 - Konferenz
Karlsruhe, 27.-28. November 1991

- Schweikhardt**
- 1) *Zusammenfassende Darstellung der Arbeit für Blinde und Sehbehinderte an der Universität Stuttgart seit 1977*
Universität Karlsruhe, 18./19. März 1991
- 2) *Integration von sehgeschädigten Studierenden und Hochschulabsolventen in die Arbeitswelt*
Universität Karlsruhe, 18./19. März 1991
- 3) *From Printed Media to Tactile Documents*
Workshop on Future Problems and Solutions for Man Machine Interaction
Concerted Action on Technology and Blindness
Stuttgart, 16./17. Dezember 1991

- Tausend, B.**
- 1) *Analogical Reasoning for Logic Programming*
First International Workshop on Inductive Logic Programming ILP-91
Viana de Castelo, Portugal, März 1991
- 2) *Analogical Reasoning for Logic Programming*
European Working Session on Learning
Porto, Portugal, März 1991
- 3) *Induktion und Analogie*
4. Arbeitstreffen der GI-FG Maschinelles Lernen
Stuttgart, Juli 1991

Walter, J. *Transputersystem am SCSI-Bus einer Workstation*
Transputer-Anwender-Treffen '91
Aachen, 17.-18. September 1991

Weber, G.

- 1) *Interaktionsformen neuerer Eingabegeräte für die Graphikanimation*
3. Fachtagung „Computeranimation“
Magdeburg, Februar 1991
 - 2) *Concurrent Gestures*
11th Congress of the International Ergonomics Association
Paris, Frankreich, Juli 1991
 - 3) *3D Technology*
EG Konzertierte Aktion „Technology and Blindness“,
Workshop “New Problems and Future Solutions of
Man-Machine Interaction”
Stuttgart, Dezember 1991
-

2.5 Tagungen

- Gunzenhäuser, R.** 1) *Mitglied des Programmausschusses des Fachgesprächs Intelligente Lernsysteme*
Gießen - Rauschholzhausen, Juni 1991
- 2) *Mitglied des Programmausschusses der 3. internationalen Tagung Computers and Handicapped People (ICCHP)*
Wien, 1992
- 3) *Mitglied des Programmausschusses der GI-Tagung Informatik in Schule und Ausbildung*
Oldenburg, 1991
- 4) *Mitglied des Programmausschusses der IFIP-Tagung Human Computer Interaction (HCI)*
Stuttgart, 1991
- Ludewig, J.** *Mitglied im Programmkommittee der GI-Fachtagung Requirements Engineering 91*
Marburg, Mai 1991
- Tausend, B.** *Organisation des 4. Arbeitstreffen der GI-FG Maschinelles Lernen*
Bollinger, T. Stuttgart, 30.-31. Juli 1991

2.6 Herausgabe von Zeitschriften

1. *Artificial Intelligence in Medicine — An International Journal*
Burgverlag : Tecklenburg
Lehmann [Mitherausgeber]
2. *Computing — Archiv für Informatik und Numerik*
Springer-Verlag : Wien, New York
Knödel [Mitherausgeber]
3. *LOG IN : Informatik in Schule und Ausbildung*
Verlag Oldenburg : München
Gunzenhäuser [Mitherausgeber]
4. *Wirtschaftsinformatik*
Verlag Vieweg : Braunschweig
Gunzenhäuser [Mitglied des Herausgeberrates]

2.7 Implementierungen

Abteilung Betriebsssoftware

ARABTEX

ArabTEX — eine Erweiterung von LATEX zur Verarbeitung arabischer Texte (Version 1)
Sprache: Metafont, TEX
Klaus Lagally

Vergleichsstudie verschiedenem Parsingverfahren
Sprache: Common Lisp
Thomas Schöbel-Theuer, Steeb

Abteilung Computersysteme

OS8-PC

Portierung des Betriebssystems OS8 V3D vom Rechner PDP-8 auf den PC.
Implementierung des OS8-Filesystems auf dem PC.
Turbo-Pascal + Assembler
Klemens Krause

FELIKS

Graphische und funktionale Emulation der Sprossenradrechenmaschine ‘Feliks’ auf dem PC.
Erweiterung zum Vierspeziessuperautomaten Quadratwurzelalgorithmus nach Denon.
Turbo-Pascal
Klemens Krause

Abteilung Dialogsysteme

AP55

Ein Hilfsprozessor für APL2 zur Aufbereitung von optisch erfassten Bilddaten für eine tastbare Ausgabe
Sprache: Assembler
Alfred Werner

GRABRA

Dialogumsetzung von graphischen Elementen im Texteditor WORD 5.0 mittels Sprachausgabe und Brailleanzeige
Gerhard Weber

COOL BROWSER

Prototyp Browser für COOL
Sprache: C, Xt
Kenji Hanakata

X-TOOLS IN COOL

Prototyp Graphische Benutzeroberfläche für COOL
 Sprache: C, Xt
Kenji Hanakata

Forschungsgruppe DRUID

XIT

- 1) *User Interface Toolkit für das X Window System*
 Common Lisp

XAM

- 2) *User Interface Metasystem für XIT*
 Common Lisp, XIT
Jürgen Herczeg, M. Wichert

XACT

- 3) *User Interface Construction Kit für XIT*
 Common Lisp, XIT
Jürgen Herczeg, M. Wichert

XGRAPH

- 4) *Werkzeug zur graphik-orientierten Visualisierung*
 Common Lisp, XIT
Hubertus Hohl

XBROWSE

- 5) *Baukastensystem zur Konstruktion anwendungsspezifischer Browser*
 Common Lisp, XIT, XGRAPH
Hubertus Hohl

XCLOS

- 6) *Browsing-Werkzeuge zur Analyse von CLOS-Programmstrukturen*
 Common Lisp, XIT, XGRAPH, XBROWSE
Hubertus Hohl

READER
INTERFACE

- 7) *Multimediales Informationssystem zum Anfordern und Inspizieren von Reiseinformation*
 Common Lisp, XIT

PRODUCTION
INTERFACE

- 8) *System zur Erstellung von Multimedia-Dokumenten*
 Common Lisp, XIT

Abteilung Intelligente Systeme

TED

Terminologische Komponente zur Repräsentation von Wissen
 CommonLisp und Clos
Peter Forster

ALAN	<i>Assertionale Komponente zur Repräsentation von Wissen</i> CommonLisp und Clos <i>Gerrit Burkert, Peter Forster, Oliver Eck</i>
INDICO	<i>Induktion von konjunktiven Konzepten</i> Prolog <i>Birgit Tausend, Irene Stahl</i>
FRAME TALK	<i>Objektorientierte Sprache zur Repräsentation von Wissen</i> CommonLisp und Clos <i>Christian Rathke</i>
ASK-ME	<i>Deutschsprachige Abfrage von Weltwissen</i> Common Lisp <i>Egbert Lehmann</i>

Abteilung Software-Engineering

PRO TEST (Profil-Test)	<i>Fünf Studierende haben ein Ada-Programm für den Einsatz auf PC entwickelt, mit dem man psychologische Tests durchführen und auswerten kann. Dazu gehört das Beantworten von Fragen, das Lösen von Aufgaben unter Zeitdruck und Reaktionstests. Die Ergebnisse der Tests werden verschiedenen Fähigkeiten zugeordnet; so wird ein Fähigkeiten-Profil der Testperson ermittelt. Lediglich Fähigkeiten, die für Mitarbeiter in Software-Projekten relevant sind, sollen untersucht werden.</i>
AUTOPOET	<i>Eine Gruppe von fünf Studierenden hat ein Ada-Programm geschrieben, das moderne Gedichte erzeugen kann. Dazu wird in einer Gedicht-Beschreibungssprache zuerst die Struktur des Gedichts zufällig bestimmt, dann werden Wörter ausgewürfelt, die in diese Struktur eingesetzt werden; schließlich werden diese Wörter noch grammatisch an ihre Stellung im Satz angepaßt. Dadurch entsteht ein zwar inhaltsloser (weil in Struktur und Wortwahl zufälliger), aber grammatisch korrekter Mehrzeiler. Oft wirken diese Erzeugnisse, als sei doch ein geheimer Sinn in ihnen verborgen.</i>

3 Fakultätsbezogene Aufgaben

3.1 Dekanat der Fakultät Informatik

Dekan

Prof. Dr. A. Reuter (IPVR)	(bis 30.09.)
Prof. Dr. K. Lagally	(ab 01.10.)

Prodekan

Prof. Dr. K. Lagally	(bis 30.09.)
Prof. Dr. A. Reuter (IPVR)	(ab 01.10.)

Sekretariat

Frau K. Erz

3.2 Zentrale Fakultätseinrichtungen

3.2.1 Bibliothek

Wissenschaftl. Beauftragter	<i>Ebinger</i>
Bibliothekarin	<i>Röger</i>
Programmbetreuung	<i>Schlebbe</i>
Wissenschaftl. Hilfskräfte	<i>Busch, Behrens, Qian, Radonniklis, Raith, V. Walter, Westermann, Zang, A. Ziegler</i>

Das größte Ereignis im Berichtsjahr war der Umzug in die Breitwiesenstraße. Hier steht für die nächste Zeit genügend Fläche zur Verfügung, so daß der Bibliotheksbestand übersichtlich präsentiert werden kann. Um dies zu unterstützen, wurde begonnen, die Bücher mit gedruckten Rückenschildern zu versehen.

Es gibt drei Terminals in der Bibliothek für den Zugriff auf den elektronischen Katalog, so daß nicht nur die Mitarbeiter der Fakultät von ihren Arbeitsplätzen aus, sondern jeder Benutzer der Bibliothek direkt in der Bibliothek im elektronischen Katalog suchen kann. Zusätzlich wurde im Nebenraum ein Drucker installiert, um die gefundenen Literaturstellen auszudrucken. In der Bibliothek gibt es jetzt mehr Einzelarbeitsplätze

als in der Azenbergstraße. Im Nebenraum stehen erstmals Gruppenarbeitsplätze zur Verfügung. Außerdem gibt es nun im Bibliotheksgebäude Kopierer.

Die Literaturdatenbank „Computer Select“ wurde für einen (für alle Benutzer zugänglichen) PC in der Bibliothek angeschafft. Sie enthält auf CD-ROM Titel und Kurzfassungen des letzten Jahrgangs aus amerikanischen Computerzeitschriften und wird monatlich aktualisiert. Da nun über das Rechenzentrum ein Datex-P-Zugang vorhanden ist, kann auch wieder auf das Literaturrecherchesystem des Südwestverbunds in Konstanz zugegriffen werden. Wir hoffen, daß es dafür (und auch für andere Datenbanken) einmal eine Mehrplatzlizenz geben wird, und daß auch im Bestand der Universitätsbibliothek online gesucht werden kann.

3.2.2 Rechnernetz

Mitarbeiter

*Uwe Berger (ab 01.03.91),
Franz Fabian (bis 31.05.91),
Mircea Fabian, Holger Sammet*

Im neuen Informatik-Gebäude in Stuttgart-Vaihingen wurde das Rechnernetz aufgebaut. Alle Räume des Gebäudes sind mit Koaxial-Kabeln (insgesamt ca. 14 km) und V.24-Leitungen (ca. 20 km) verkabelt. Außerdem sind im Gebäude Lichtwellenleiter für ein Backbone-Netz verlegt. Das im wesentlichen auf Ethernet basierende Rechnernetz ist in dreizehn Subnetze unterteilt. Es werden Multiport-Repeater, Bridges und ein cisco-Router eingesetzt. Die Verbindung zum Campus-Netz wurde zunächst durch eine Standleitung (56 kbps) hergestellt. Im Dezember konnte diese Verbindung auf Glasfaser (10 Mbps) umgestellt werden.

Ende 1991 waren über 300 Rechner an das Rechnernetz der Fakultät Informatik angeschlossen:

- 3 HP Mehrbenutzersysteme (HP 840, HP 845)
 - 2 DEC Server (DEC 5400)
 - 6 Sun Server (4/110, 4/260, 4/370)
 - 8 Apollo Arbeitsplatzrechner
 - 1 Cadmus Arbeitsplatzrechner
 - 28 DEC Arbeitsplatzrechner (DEC 2100, DEC 3100, VS 3100, VAX 3100)
 - 5 HP Arbeitsplatzrechner (HP 320, HP DN400)
 - 7 IBM Arbeitsplatzrechner (RS 6000/320)
 - 105 Sun Arbeitsplatzrechner (Sun 3, Sun 4)
 - 19 Mac II Arbeitsplatzrechner
 - 110 Personal Computer (HP Vectra, IBM PS/2, PC-AT und Kompatible)
 - 13 X-Terminals (NCD, VT 1200, IBM X-Station)
 - 3 Terminalserver (für ca. 50 Terminals)
 - 6 LISP-Maschinen (Symbolics, TI Explorer)
-

- 1 MasPar MP 1216-A
- 1 Sequent Symmetry S27
- 1 IBM 9370
- 1 VAX-Cluster (2×6420)
- 1 Tandem TXP

Auf diesen Rechnern werden überwiegend UNIX-Betriebssysteme (AIX, HP-UX, SunOS, Ultrix) und MS-DOS (auf den Personal Computern) eingesetzt.

Für die Lehrveranstaltungen im Grundstudium sind bisher alphanumerische Terminals vorhanden, die über Terminalserver Zugang zu den HP Mehrbenutzersystemen haben. Die Ablösung dieser Terminals durch Graphikarbeitsplätze wurde geplant und beantragt.

Die Datensicherung für viele der mit eigenen Platten ausgestatteten Rechner erfolgt auf zwei Exabyte-Geräte (je 2 GB pro 8 mm Video-Tape) jede Nacht in der Zeit von 22.00 Uhr – 6.00 Uhr. Es werden 27 Maschinen verschiedener Hersteller (Sun, HP, DEC, IBM) unter verschiedenen Betriebssystemen (AIX, HP-UX, SunOS, Ultrix) gesichert, was einen Bedarf von ca. 13 Tapes (26 GB) monatlich erfordert.

3.2.3 Workstations

Mitarbeiter

Mircea Fabian

Im Laufe des Jahres 1991 sind bei den Sun-Workstations (Sun 3 und 4) insgesamt 23 Service-Fälle aufgetreten. Die erfolgten Reparaturen verteilen sich auf die entsprechenden Hardware-Teile wie folgt:

Service-Statistik 1991

Hardware	CPU	Monitore	Platten	Sonst.	Gesamt
Anzahl Ausfälle	2	14	3	4	23

Global betrachtet kann man davon ausgehen, daß die Rechner zuverlässig liefen und durch die erfolgten Service-Fälle keine ernsthafte Störung des Betriebes stattfand, so daß die Verfügbarkeit der Workstations über die ganze Zeit gewährleistet war.

3.2.4 Elektrotechnik

Mitarbeiter

Holger Sammet

Im Jahre 1991 wurde bei einem Großteil der 3/50 Sun-Workstations ein Austausch der CPU vorgenommen. Durch Einbau der neuen 3/60 CPU konnte die Rechenleistung gesteigert werden. Nach dem Umzug der Fakultät in den „Informatik-Bau“ in

der Breitwiesenstraße konnten endlich alle Rechner den Richtlinien entsprechend angegeschlossen werden. Im Rahmen einer zweieinhalbjährigen Ausbildung haben drei angehende Kommunikationselektroniker ein sechsmonatiges Betriebspraktikum absolviert. Anschließend haben sie die Facharbeiterprüfung vor der Industrie- und Handelskammer zu Stuttgart erfolgreich abgelegt.

3.2.5 Mehrbenutzersysteme

Mitarbeiter

*Heinz Kohl (bis 31.12.91), Otto Finger,
Horst Schneider, Om Parkash Wahi*

Im Laufe des Jahres wurden die zwei für die Lehre im Grundstudium zur Verfügung stehenden HP 9000/835 umgerüstet auf HP 9000/845 und ausgebaut. Sie haben jetzt je 64 MB Haupt- und je 2.4 GB Hintergrundspeicher. An einem der beiden Rechner steht jetzt ein Gigatape, auf dem anderen ein PC-Emulator mit einem schnellen Doppellaufwerk für 3.5" Disketten bereit. Inzwischen wäre es überfällig, die als Terminals angeschlossenen Atari 1040ST durch X-Windowfähige Geräte, geplant sind diskless Workstations, zu ersetzen. Der Beschaffungsantrag ist jedoch noch nicht genehmigt.

3.2.6 PC-Pools

Beauftragter

Wolfgang Hersmann

Wissenschaftl. Hilfskräfte

Milan, Utz, Sing, Waigand, Waitzmann

bis Februar:

*Pätzold, Sacher, Schäfer, Schwen,
Wittig*

bis Juli:

Ly Huynh, Mardassi

ab Oktober:

Kammoun

Im Jahre 1991 wurde zur bestehenden Software keine neue Software installiert.

Wie schon im Jahre 1990 so wurden auch 1991 Praktika zu den Vorlesungen *Einführung in die Informatik*, *Grundlagen der Informatik* und *Informatik III* sowie verschiedene Software-Praktika, Studienarbeiten und Diplomarbeiten auf den PC-Pools durchgeführt.

Im Oktober 1991 wurde das Rechnernetz Office Share modifiziert. Dies ermöglichte die Anbindung von 15 PCs im PC-Pool an das Fakultäts-Hausnetz. Ein aufgetretener Engpaß bei den Atari-Terminals wurde dadurch beseitigt.

3.3 Lehre

3.3.1 Aufbau des Informatikstudiums

3.3.1.1 Diplomstudiengang Informatik

Der seit dem Wintersemester 1974/75 gültige und in den Jahren 1987 bis 89 aktualisierte Studienplan sieht nach der Diplomprüfung in Informatik eine Auffächerung in verschiedene Studienschwerpunkte vor:

- Theorie der Informatik
- Software-orientierte Informatik
- Hardware-orientierte Informatik
- Anwendungsorientierte Informatik I: Ingenieursysteme
- Anwendungsorientierte Informatik II: Mensch-Maschine-Kommunikation

Dazu tritt ein Nebenfach, das die Studierenden mit Methoden und Anwendungen eines anderen Fachgebietes vertraut macht. In enger Zusammenarbeit mit den betreffenden Fakultäten werden derzeit die Nebenfächer

- Bauingenieurwesen/Verkehrs wesen
- Betriebswirtschaftslehre
- Biologie
- Elektrotechnik
- Energietechnik
- Linguistik
- Mathematik
- Steuerungstechnik
- Technische Kybernetik
- Verfahrenstechnik

angeboten. In Einzelfällen kann der Prüfungsausschuß Informatik Ausnahmegenehmigungen für andere Nebenfächer erteilen.

Das Studium wird mit dem akademischen Grad eines Diplom-Informatikers (Dipl.-Inform.) abgeschlossen.

Die Prüfungsordnung und der Studienplan Informatik können bei der Fakultät Informatik oder der Studienberatung angefordert werden.

Auf Grund der landesweiten Kapazitätsberechnung ergab sich für das Wintersemester 1991/92 für die Universität Stuttgart eine Aufnahmezahl von rund 185 Informatik-Studienanfängern, die von Senat und Ministerium – wie in den Vorjahren – auf 215

Plätze erhöht wurde. Von fast 400 Bewerbern haben knapp 210 mit dem Studium in Stuttgart begonnen.

Damit ergab sich – dem Trend in den alten Bundesländern folgend – erstmals seit mehr als fünf Jahren keine weitere Überlast für die Stuttgarter Informatik, die allerdings auf Grund der hohen Aufnahmезahlen in den Vorjahren mit fast 1400 Studierenden stark belastet bleibt.

Im Jahre 1991 waren insgesamt 11 Stellen für Professoren besetzt. Für 4 weitere Stellen konnten die laufenden Berufungsverfahren teilweise abgeschlossen werden. Ein größerer Teil der Lehre konnte von den drei Honorarprofessoren, auswärtigen Lehrbeauftragten sowie erfahrenen Mitarbeiterinnen und Mitarbeitern der Fakultät Informatik abgedeckt werden.

3.3.1.2 Nebenfachstudium Informatik

In der Universität Stuttgart wird Informatik als Nebenfach bzw. Technisches Schwerpunktstudium in den Studiengängen Mathematik und „technisch orientierter Diplomkaufmann“ sowie in der Studienrichtung Linguistik angeboten, seit Beginn des Wintersemesters 1991/92 auch im Studiengang Computer-Linguistik und Technik-Pädagogik.

Die Fakultät Informatik übernimmt darüberhinaus die Ausbildung in „Grundlagen der Informatik“ für die Studiengänge „technisch-orientierter Diplomkaufmann“, Mathematik, Luft- und Raumfahrt, Vermessungswesen, technische Biologie sowie für die Studiengänge der Maschinenbaufakultäten.

3.3.2 Lehrveranstaltungen

3.3.2.1 Lehrangebot im Sommersemester 1991

A. Grundstudium für Hörer anderer Fakultäten

Grundlagen der Informatik II (für Studiengang techn. orient. Diplomkaufm.)	2 V 1 Ü	Glatthaar (LA) Nitsche-Ruhland
Grundlagen der Informatik II (für Studiengang Physik, Luft- und Raumfahrt)	2 V 1 Ü	Gunzenhäuser Nitsche-Ruhland
Grundlagen der Informatik II (für VD Maschinenwesen und Verfahrenstechnik)	2 V	Baitinger, Dammert
Grundlagen der Informatik II	1 P	Böhm

B. Pflichtlehrveranstaltungen

2. Semester

Analysis II für Informatiker	2 V 1 Ü	Werner, Lesky jun. Werner, Lesky jun.
Lineare Algebra	4 V 1 Ü	Degen Degen
Differentialgleichungen (für Informatiker)	2 V 1 Ü	Werner, Lesky jun. Werner, Lesky jun.
Einführung in die Informatik II	4 V 2 Ü	Lagally Lagally
Physikalische und elektrotechnische Grundlagen II	2 V 1 Ü	Baitinger, Rettig Baitinger, Rettig

4. Semester

Numerik	3 V 1 Ü	Berger Berger
Software-Praktikum I	4 P	Eggenberger, Ziegler u.a.
Aufbau von Datenverarbeitungsanlagen	3 V 1 P	Burkhardt Burkhardt

Hardware-Praktikum	4 P	Burkhardt, Walter, Zipperer
--------------------	-----	--------------------------------

6. und 8. Semester

Entwurf und Analyse von Algorithmen und Datenstrukturen	2 V 1 Ü	Knödel Knödel
Berechenbarkeit und Komplexitätstheorie	3 V 1 Ü	Ambos-Spieß Ambos-Spieß
Betriebssysteme	4 V 1 Ü	Lagally Lagally

C. Wahlpflichtveranstaltungen

Kombinatorik	2 V	Münchow
Prädikatenlogik	3 V 1 Ü	Mijderwijk Mijderwijk
Implementierung von Datenbanksystemen	3 V 1 Ü	Reuter Reuter
Graphische Datenverarbeitung II	2 V	Grieger
Nichtprozedurale Programmierung	2 V	Ludewig, Lichter
Verteilte Systeme	3 V 1 Ü	Rothermel Rothermel
Rechnernetze	4 V 1 Ü	Rothermel Rothermel
EA-Organisation	2 V	Hieber
Mikroprogrammierung	3 V	Ebert
Entwurf kundenspezifischer Schaltungen	2 V 1 Ü	Zipperer Zipperer
Entwurf von Schaltnetzen und Schaltwerken	2 V 1 Ü	Baitinger, Lanchès Baitinger, Lanchès
Rechnergestützter Schaltungsentwurf	2 V 1 Ü	Baitinger, Ryba Baitinger, Ryba
Automatisierung (CAM, CAP, CAD/NC) II	1 V 1 Ü	Storr Storr

Echtzeitdatenverarbeitung	3 V	<i>Eggenberger</i>
Rechnerunterstütztes Lernen	2 V	<i>Gunzenhäuser</i>
	1 Ü	<i>Gunzenhäuser</i>
Intelligente Systeme II	2 V	<i>Lehmann</i>
	1 Ü	<i>Lehmann, Burkert</i>
Verteilte objektorientierte Anwendungen	2 V	<i>Heuser (LA)</i>
Kleinrechnerbetriebssysteme (für PC's)	2 V	<i>Eggenberger</i>
Kleinrechnerbetriebssysteme (UNIX)	2 V	<i>Schimpf</i>
Text- und Listenverarbeitende Verfahren	2 V	<i>Hanakata</i>
Parallele Komplexitätstheorie	2 V	<i>Reinhardt</i>
Informationstheorie	3 V	<i>Reuß</i>
Interaktives Problemlösen	2 V	<i>Schweikhardt</i>
Expertensysteme	2 V	<i>Rathke</i>
Informationssysteme II	2 V	<i>Reuter</i>
Datenbank-Anwendungssysteme	2 V	<i>Reuter</i>
Methoden und Notationen der frühen Phasen der Software-Entwicklung	2 V	<i>Ludewig, K. Schneider</i>
Konzeption und Aufbau objektorientierter Programme	2 V	<i>Ludewig, Lichter</i>
	2 Ü	<i>Ludewig, Lichter</i>
Deduktionsverfahren II	2 V	<i>Schönfeld</i>
Objektorientierte Programmierung	2 V	<i>Hanakata</i>
Programmverifikation	2 V	<i>Liedtke</i>
	1 Ü	<i>Liedtke</i>
Neuronale Netze	2 V	<i>Zell</i>
Datenbanksysteme für die Fabrikautomatisierung	2 V	<i>Kupper</i>
	1 Ü	<i>Kupper</i>
Verteilte Systeme	2 V	<i>Haban (LA)</i>
Implementierung von Programmiersprachen	2 V	<i>Sebesta</i>
	1 Ü	<i>Sebesta</i>
Konzepte höherer Programmiersprachen	2 V	<i>Sebesta</i>
Fachpraktikum: Software-Engineering	4 P	<i>Ludewig u. Mitarb.</i>

Fachpraktikum: Datenbankpraktikum	4 P	<i>Reuter u. Mitarb.</i>
Fachpraktikum: Rechnerarchitektur	4 P	<i>Walter, Zimmermann, Zipperer</i>
Fachpraktikum: Interaktive und intelligente Systeme	4 P	<i>Herczeg, Tausend</i>
Fachpraktikum: Parallele Programmierung	4 P	<i>Bräunl</i>
Fachpraktikum: Simulation neuronaler Netze	4 P	<i>Zell</i>

D. Seminare

Betriebssysteme für Parallelrechner	2 S	<i>Walter</i>
Hardware-Implementation Neuronaler Netze	2 S	<i>Zimmermann</i>
Dialogformen und Interaktionstechniken der Mensch-Computer-Kommunikation	2 S	<i>Gunzenhäuser, Nitsche-Ruhland</i>
Syntaxanalyse	2 S	<i>Schöbel-Theuer</i>
Entwurf mit programmierbaren Logikbausteinen	2 S	<i>Rettig</i>
Seminar über Testing	2 S	<i>Sebesta</i>
Software-Ergonomie	2 S	<i>Fehrle (LA), Hampp</i>
Natürlichsprachliche KI-Systeme	2 S	<i>Lehmann, Rathke</i>
Modelle des Software-Engineering	2 S	<i>Deininger</i>

E. Hauptseminare

Netzwerk-Verwaltung	2 HS	<i>Rothermel</i>
Struktur moderner Großrechnersysteme	2 HS	<i>Ebert</i>
Bildverstehen	2 HS	<i>Hanakata</i>
Probabilistische Komplexitätstheorie	2 HS	<i>Ebinger</i>
Datenverwaltungsmethoden in der Bildverarbeitung	2 HS	<i>Reuter</i>
Wissensrepräsentationssysteme	2 HS	<i>Lehmann, Burkert, Forster</i>
Softwarewartung und -wiederverwendung	2 HS	<i>Ludewig, Schwille</i>
Formale Methoden der Mensch-Computer-Kommunikation	2 HS	<i>Weber (LA)</i>

Grundlagen der Mensch-Computer-Kommunikation	2 HS	<i>Gunzenhäuser, Herczeg</i>
Integrierter Systementwurf	2 HS	<i>Baitinger</i>
Parallele Algorithmen	2 HS	<i>Baitinger, Bräunl</i>
Neue Transaktionskonzepte	2 HS	<i>Reuter, U. Schmidt, Wörner</i>

F. Wahlveranstaltungen

Übungen zur Informationstheorie	1 Ü	<i>Reuß</i>
---------------------------------	-----	-------------

G. Kompaktkurse

OCCAM2	<i>Lanchès</i>
Smalltalk-80	<i>Herczeg, Mahling</i>
COOL	<i>Hanakata</i>
PROLOG	<i>Nitsche-Ruhland</i>
Common LISP	<i>Tausend, Forster</i>
Ada	<i>Sebesta</i>

3.3.2.2 Lehrangebot im Wintersemester 1991/92

A. Grundstudium für Hörer anderer Fakultäten

Grundlagen der Informatik I (für Studiengang techn. orient. Diplomkaufmann)	2 V 1 Ü	Glatthaar (<i>LA</i>) Nitsche-Ruhland
Grundlagen der Informatik I (für Studiengang Physik, Luft- und Raumfahrt)	2 V 1 Ü	Gunzenhäuser Schweikhardt
Grundlagen der Informatik I (für VD Maschinenwesen und Verfahrenstechnik)	2 V 1 Ü	Baitinger Baitinger, Dammert

B. Pflichtlehrveranstaltungen

1. Semester

Analysis I	5 V 2 Ü	Werner, Lesky jun. Werner, Lesky jun.
Mathematische Grundlagen	4 V 2 Ü	Lesky sen. Lesky sen.
Einführung in die Informatik I	4 V 2 P	Lagally Lagally u. Mitarb.
Physikalische und elektrotechnische Grundlagen I	2 V 1 Ü	Burkhardt Burkhardt, Walter

3. Semester

Wahrscheinlichkeitstheorie und Warteschlangen	2 V 1 Ü	D. Schmidt D. Schmidt
Logik	4 V 1 Ü	Diekert Diekert
Kombinatorische und sequentielle Netzwerke	2 V 1 Ü	Eggenberger Eggenberger
Einführung in die Informatik III	4 V 2 Ü	Reuter Reuter

5. Semester

Automatentheorie und formale Sprachen	3 V 1 Ü	<i>Gerber</i> <i>Gerber</i>
Rechnerarchitektur	4 V 1 Ü	<i>Walter, Zipperer</i> <i>Walter, Zipperer</i>
Interaktive und intelligente Systeme	4 V	<i>Gunzenhäuser,</i> <i>Rathke</i>
	1 Ü	<i>Gunzenhäuser,</i> <i>Burkert, Herczeg</i>
Informationssysteme/Datenbanken	4 V 1 Ü	<i>Reuter</i> <i>Reuter</i>
Programmiersprachen und Compilerbau	4 V 1 Ü	<i>Schwinn</i> <i>Schwinn</i>

C. Wahlpflichtveranstaltungen

Codierungstheorie	3 V	<i>Reuß</i>
Entwurf großer Systeme	2 V	<i>Endres</i>
Rechnernetze	4 V	<i>Rothermel</i>
	1 Ü	<i>Barth, Sembach</i>
Software Engineering	4 V	<i>Ludewig</i>
	1 Ü	<i>Deininger</i>
Einführung in die parallele Programmierung	2 V	<i>Bräunl</i>
Aufbau und Einsatz von Mikrocomputern	2 V	<i>Burkhardt</i>
	1 Ü	<i>Burkhardt</i>
Signalverarbeitung durch Neuronale Netze	2 V	<i>Zimmermann</i>
	1 P	<i>Zimmermann</i>
CAM, CAP, CAD/NC – Automatisierung des technischen Informationsflusses I	1 V	<i>Storr</i>
	1 Ü	<i>Storr</i>
Spezifikation digitaler Systeme	2 V	<i>Ryba</i>
Offene integrierte Entwurfssysteme	2 V	<i>Baitinger, Ryba</i>
Musteranalyse und Bildverstehen	2 V	<i>Hanakata</i>
Symbolmanipulation	2 V	<i>Rathke, Forster</i>
	2 Ü	<i>Rathke, Burkert, Forster, Herczeg</i>

Maschinelles Lernen	2 V 1 Ü	Tausend, Wirth Tausend, Wirth
Graphische Datenverarbeitung I	1 V 1 Ü	Grieger Grieger
Petrinetze	2 V	Gerber
Kombinatorik II	2 V	Münchow
Nullstellenseparation	2 V	Mijderwijk
Programmtransformationen	2 V	Liedtke
Rekursionstheorie	2 V	Reinhardt
Aufbau von Speicherstrukturen	2 V	Ebert
Hochgeschwindigkeitsnetze und heterogener Netzverbund	2 V	Ebert
Software-Ergonomie	2 V	<i>M. Herczeg (LA)</i>
Leistungsmessung von Systemen	2 V 1 Ü	Hieber Hieber
Systemprogrammierung (II)	2 V	Eggenberger
Kleinrechner-Betriebssysteme (SOLO)	2 V	Schöbel-Theuer
Effiziente parallele Algorithmen	2 V	Bräunl, Zell
Fehlerbehandlung und Konfiguration in verteilten Systemen	2 V	Sembach
Datenschutz	2 V	Biller
Software-Praktikum II (alte Prüfungsordnung)	4 P	Ziegler u.a.
Parallele Programmierumgebungen	4 P	Reuter, Maier
Rechnernetze und verteilte Systeme	4 P	Rothermel

D. Seminare

Neuere kundenspezifische Schaltungen	2 S	Zipperer
Simulation elektronischer Schaltungen	2 S	Lanchès
Grundlagen und Anwendungen graphischer Informationsverarbeitung	2 S	Schweikhardt
Hypermedien	2 S	Nitsche-Ruhland

Wissensrepräsentationssysteme	2 S	<i>Forster, Burkert</i>
Verteilte Algorithmen	2 S	<i>Kovacs</i>
Lastbalancierung in verteilten Systemen	2 S	<i>Barth</i>
Entwurf von Netzwerken	2 S	<i>Fabian</i>
Hochgeschwindigkeitsnetze	2 S	<i>Sembach</i>
Software-Qualitätssicherung	2 S	<i>Deininger</i>
Software Configuration Management	2 S	<i>Schwille</i>
Logikprogrammierung und Parallelverarbeitung	2 S	<i>Schimpf</i>

E. Hauptseminare

Integrierter Systementwurf	2 HS	<i>Baitinger u. Mitarb.</i>
Theoretische Informatik	2 HS	<i>Diekert, Gerber</i>
Techniken des Entwurfs von Benutzungsoberflächen	2 HS	<i>Gunzenhäuser, Herczeg</i>
Verteilte Betriebssysteme	2 HS	<i>Rothermel</i>
Sprachkonzepte für verteilte Systeme	2 HS	<i>Rothermel</i>
Moderne Analyse- und Entwurfsmethoden	2 HS	<i>K. Schneider</i>
Neuere Entwicklungen in Computersystemen	2 HS	<i>Burkhardt</i>
Objektorientierte Programmierung	2 HS	<i>Hanakata</i>

F. Wahlveranstaltungen

Codierungstheorie	1 Ü	<i>Reuß</i>
Durchführung wissenschaftlicher Arbeiten	2 K	<i>Ludewig u. Mitarb.</i>

G. Kompaktkurse

Kompaktkurs C	<i>Hanakata</i>
Kompaktkurs Common-LISP	<i>Tausend, Forster</i>
Kompaktkurs PROLOG	<i>Nitsche-Ruhland</i>
Kompaktkurs UNIX	<i>Dammert</i>
Kompaktkurs Ada	<i>Schwille</i>
Kompaktkurs Eiffel	<i>Lichter</i>

3.3.3 Informatik-Kolloquium

03.01.	Girgensohn, Andreas Universität of Colorado at Boulder Departement of Com- puter Science and Institute of Cognitive Science	<i>Modifizierbarkeit von Designerum- gebungen durch Endbenutzer</i>
07.01.	Staiger, L. RWTH Aachen, Lehrstuhl für Informatik II	<i>Formale Sprachen, ω-Sprachen und Programmeigenschaften</i>
08.01.	Diekert, Volker TU München, Institut für Informatik	<i>Eine neue Halbwortsemantik für nichthalrende nebenläufige Prozesse</i>
11.01.	Wagner, K. Julius-Maximilians-Universität Würzburg, Institut für Informatik	<i>Interaktive Proof-Systeme</i>
14.01.	Lange, K. TU München, Institut für Informatik	<i>Parallele Komplexitätstheorie</i>
15.01.	Jungnickel, D. Justus-Liebig-Universität Gießen	<i>Zum Datentyp „Endliche Körper“</i>
18.01	Otto, F. Gesamthochschule Kassel, Fach- bereich Mathematik-Informatik	<i>Wortersetzungssysteme als ein Hilfsmittel, um Vermutungen über Termersetzungssysteme zu beweisen oder zu widerlegen</i>
23.01.	Claus, Volker Universität Oldenburg, Fachbe- reich Informatik	<i>Modellierung von Bürovorgängen</i>
25.01.	Meinel, Christoph Humboldt-Universität Berlin	<i>Obere und untere Schranken für Branching-Programm-Berechnungen</i>
12.02.	Posthoff, Christian Universität Chemnitz	<i>Zur Konstruktion optimaler Diagnose-Systeme</i>

30.04.	Borrmann, Lothar Siemens AG München, Zentralabteilung Forschung und Entwicklung	<i>Virtuell gemeinsamer Speicher im Parallelrechnerprojekt EDS</i>
14.05.	Riekert, Wolf-Fritz Forschungsinstitut für anwendungsorientierte Wissensverarbeitung (FAW) Ulm	<i>Wissensbasierte Auswertung von Satellitenbilddaten zur Fernerkundung der Welt</i>
10.06.	Lindsay, Bruce IBM Almaden, USA	<i>Object-Oriented Issues and Relational Technology</i>
11.06.	Graefe, Goetz Universität of Boulder, USA	<i>Encapsulation of Parallelism in the Vocano Query Processing System</i>
02.07.	Rabenseifner, R. Regionales Rechenzentrum der Universität Stuttgart	<i>Das DFN Remote Procedure Call Tool (DFN-RPC)</i>
09.07.	Kalfa, Winfried Technische Universität Dresden	<i>Prozessor-Scheduling in verteilten Betriebssystemen</i>
15.10.	Keller, W. Senior Systems Engineer IBM, Space Shuttle Organization	<i>Software Process for Space Shuttle Software</i>

3.3.4 Habilitationen und Examensarbeiten

3.3.4.1 Dissertationen

Knopik, Thomas *Methoden des maschinellen Lernens für den interaktiven Wissenserwerb*

Hauptbericht : Gunzenhäuser

Mitbericht : Lehmann

Maier, Dieter *Wissensbasierte Planungsunterstützung als Aufgabe der Mensch-Computer-Kommunikation*

Hauptbericht : Gunzenhäuser

Mitbericht : Fischer (Univ. of Colorado at Boulder)

3.3.4.2 Diplomarbeiten

Antz, Dirk	<i>Beschreibung eines Sensorsystems im Kraftfahrzeug auf verschiedenen Abstraktionsebenen mit dem Software-Paket SABER</i> Prüfer: Baitinger (IPVR) Betreuer: Ryba (IPVR)
Bader, Hans-Jürgen	<i>Dokumentation von CAD-Modellen und Übertragung dieser Information über standardisierte Schnittstellen</i> Prüfer: Lagally Betreuer: Mohrmann (debis Systemhaus GmbH)
Bäumer, Dirk	<i>Konzeption und Realisierung eines Werkzeugs zum Entwurf von fensterorientierten Benutzeroberflächen</i> Prüfer: Ludewig Betreuer: Deininger
Banhart, Matthias	<i>Massiv parallele Implementierung einer realen Stücklistenauflösung</i> Prüfer: Reuter (IPVR) Duppel: Duppel (IPVR)
Bayer, Harald	<i>Massiv parallele Simulation der selbstorganisierenden Karten von Kohonen auf einem SIMD-Rechner</i> Prüfer: Reuter (IPVR) Betreuer: Zell (IPVR)
Bayer, Rolf	<i>Massiv parallele Berechnung des optischen Flusses in Bildfolgen</i> Prüfer: Baitinger (IPVR) Betreuer: Bräunl (IPVR)
Bosch, Monika	<i>Beschreibungssprache für graphenorientierte, vermaschte Diagramme</i> Prüfer: Baitinger (IPVR) Betreuer: Dammert (IPVR)
Büttner, Frank	<i>Beschreibung eines Sensorsystems im Kraftfahrzeug auf verschiedenen Abstraktionsebenen mit der Hardware-Beschreibungssprache VHDL</i> Prüfer: Baitinger (IPVR) Betreuer: Ryba (IPVR)

Burkert, Holger	<i>Systemarchitektur und Betriebsmodell für das verteilte Transaktionssystem APRICOT</i> Prüfer: Reuter (IPVR) Betreuer: Wächter (IPVR)
Clauss, Wolfgang	<i>Konzeption, Spezifikation und Implementierung einer Systemmodifikationseinrichtung für vernetzte UNIX-Systeme</i> Prüfer: Eggenberger Betreuer: Meitner (IFF), Reim (IFF)
Dinkelacker, Robert	<i>Paralleler Eigenwertsolver für Transputer-Netzwerk</i> Prüfer: Ebert (AEG-Ulm) Betreuer: Böhm (IPVR)
Dörre, Ingrid	<i>TONIKA — Ein Werkzeug für die computerunterstützte harmonische Analyse</i> Prüfer: Gunzenhäuser Betreuer: Mahling
Drehmann, Kerstin	<i>Perspektiven für FrameTalk</i> Prüfer: Lehmann Betreuer: Rathke
Emmerich, Conny-Michael	<i>Subordinierende Temporale im Deutschen — ein Adetal(V)-basierter Beschreibungsansatz</i> Prüfer: Rohrer (IMS) Betreuer: Kamph (IMS)
Engelhardt, Stefan	<i>Automatische Übersetzung einer massiv parallelen Programmiersprache für sequentielle und parallele Rechnerarchitekturen</i> Prüfer: Baitinger (IPVR) Betreuer: Bräunl (IPVR)
Fabricius, Joachim	<i>Interpolationsverfahren für eine Wetterdatenbank auf dem relationalen Datenbanksystem INGRES</i> Prüfer: Reuter (IPVR) Betreuer: Neugebauer (IPVR)
Feldner, Wolfgang	<i>Abbildung eines Geschäftsmodells in ein Dialogmodell: Ein formaler Ansatz zur Vorgehensbeschreibung</i> Prüfer: Gunzenhäuser Betreuer: Cholewa (IBM)

Förch, Thomas	<i>Eine prozedurale Umsetzung des RAPiD-Modells</i> Prüfer: Burkhardt Betreuer: Walter
Gebhardt, Rolf	<i>Menschen am Computer: Handeln? Kooperieren? Kommunizieren?</i> Prüfer: Gunzenhäuser
Gmeinwieser, Norbert	<i>Entwicklung und Aufbau einer PC-Einsteckkarte mit dem 32-Bit-RISC-Processor Hyperstone E1.</i> Prüfer: Burkhardt Betreuer: Walter
Greiner, Thorsten	<i>Heuristiken für das dual-bin-packing-Problem</i> Prüfer: Knödel
Grotepaß, Frank	<i>Perspektiven für FrameTalk</i> Prüfer: Lehmann Betreuer: Rathke
Grün, Birgit	<i>Ein Expertensystem zur Steuerberatung</i> Prüfer: Lehmann Betreuer: Rathke
Hanisch, Jürgen	<i>Technologieabhängige Logiksynthese für einen 3D-CMOS- Prozeß</i> Prüfer: Baitinger (IPVR) Betreuer: Rettig (IPVR), Roos (IME)
Hanske, Wilhelm Wilfried	<i>Informationstheoretische Grundlagen und neuere Anwen- dungen der elektronischen Kommunikation mit Nachrichten und Dokumenten (Electronic Mail)</i> Prüfer: Gunzenhäuser
Harreus, Thomas	<i>Entwurf und Implementierung der Steuerwerke für einen Radschlupfprozessor</i> Prüfer: Baitinger (IPVR) Betreuer: Rettig (IPVR), Büchner (IME)
Hecker, Christoph	<i>Dreidimensionale Packungen</i> Prüfer: Knödel
Herrmann, Michael	<i>GRACOMP Compiler für ein graphisches Simulationssy- stem</i> Prüfer: Lagally Betreuer: Schmid (IPA)

Hung Cuong, Do *Entwicklung einer CAD-Umgebung für FTS-Simulator*
Prüfer: Ebert (AEG-ULM)
Betreuer: Reising (IPA)

Koch, Martin *Anbindung und Ansteuerung eines Transputer-System über den SCSI-Bus*
Prüfer: Burkhardt
Betreuer: Walter

Kortüm, Gerd *Temporales Schließen in einem natürlichsprachlichen System*
Prüfer: Lehmann
Betreuer: von Luck (IMB)

**Kouroufexis,
Andreas** *Entwurf und Implementierung eines automatischen Codeformatierers*
Prüfer: Ludewig
Betreuer: K. Schneider,

Kreissl, Wolfgang *Interaktives Erkunden gedruckter Dokumente*
Prüfer: Gunzenhäuser
Betreuer: Schweikhardt

Kubiak, Günther *Vorhersage von Börsenkursen mit neuronalen Netzen*
Prüfer: Reuter (IPVR)
Betreuer: Zell (IPVR)

Langbein, Karin *Entwurf und Gestaltung eines Graphikgrundsystems für die Erstellung von NC-Bedienoberflächen*
Prüfer: Storr (ISW)
Betreuer: Junghans (ISW)

Levec, Tatjana *Erweiterung eines verteilten SQL-Datenbanksystems zur Realisierung hoher Datenverfügbarkeit*
Prüfer: Reuter (IPVR)

Liebelt, Sabine *Entwicklung und Untersuchung von massiv parallelen Hidden-Surface- und Raytracing-Algorithmen*
Prüfer: Ludewig
Betreuer: Bräunl (IPVR)

Lieven, Oliver *Der Austausch komplex strukturierter Daten mittels objekt-orientierter Programmierung (C++)*
1. Prüfer: Eggenberger
2. Prüfer: Lagally

Löffler, Peter	<i>Interaktive, modellbasierte Diagnose nachrichtentechnischer Systeme</i> Prüfer: Gunzenhäuser Betreuer: Herczeg
Lutz, Rainer	<i>Transaktionsverwaltung auf der Basis von Transaktionsbeziehungsnetzen</i> Prüfer: Reuter (IPVR) Betreuer: Wächter (IPVR)
Ly, Huynh Dung	<i>Programmierung einer Bedien- und Anzeigoberfläche für einen Qualitätsregelkreis in einem flexiblen Fertigungssystem</i> Prüfer: Storr (ISW) Betreuer: Heger (ISW)
Mackamul, Harald	<i>Ein Tutoren-System für musikalische Stile</i> Prüfer: Gunzenhäuser Betreuer: Mahling
Mardassi, Mohsen	<i>Dialogformen für Blinde mit kombinierter Sprachausgabe und Brailleanzeige</i> Prüfer: Gunzenhäuser Betreuer: Weber
Menz, Falko	<i>Vergleich zweier kontextfreier Parsing-Verfahren</i> Prüfer: Lagally Betreuer: Schöbel-Theuer
Michel, Heiko	<i>Messung des Lastverhaltens paralleler C-Programme auf Transputernetzen</i> 1. Prüfer: Baitinger (IPVR) 2. Prüfer: Mätzel (TU Chemnitz) Betreuer: Lanchès (IPVR)
Müller, Adrian	<i>ICL Integrity Control Language</i> Prüfer: Gunzenhäuser Betreuer: von Luck (IBM)
Müller, Hiltrud	<i>Bewertung verschiedener Replikationsmechanismen für eine verteilte DB-Anwendung mit hohen Verfügbarkeitsanforderungen</i> Prüfer: Reuter (IPVR)

Mumme, Martin	<i>Darstellung des Netzzustandes bei verteilter paralleler Auftragsbearbeitung</i> Prüfer: Reuter (IPVR) Betreuer: Winckler (IPVR)
Naschold, Michael	<i>DOS-Schnittstelle für UNIX</i> Prüfer: Burkhardt
Nguyen, Van Quan	<i>Entwicklung einer Testumgebung für Fertigungsanlagen mit SPS</i> 1. Prüfer: Ebert (AEG-Ulm) 2. Prüfer: Bullinger (IAO) Betreuer: Hengel (IAO)
Ocker, Mathias C.	<i>Realisierung eines graphischen Lastmonitors zur Analyse massiv paralleler Anwendungen</i> Prüfer: Reuter (IPVR) Betreuer: Schiele (IPVR)
Paul, Fritz	<i>Meßdatenerfassung-, Regelungspaket für ATARI ST</i> 1. Prüfer: Ebert (AEG-Ulm) 2. Prüfer: Lagally Betreuer: Homeister (IPVR)
Räumschüssel, Stefan	<i>Konzept und Implementierung eines Systems zur parallelen Ausführung von Logikprogrammen auf Transputern auf der Basis des RAPID-Datenflußmodells</i> Prüfer: Lagally Betreuer: Schimpf
Reczko, Martin	<i>Einsatzmöglichkeiten von Neuronalen Netzen bei medizinischen Echtzeitsignalen zur Störunterdrückung</i> Prüfer: Lehmann Betreuer: Tausend
Rieth, Peter	<i>Verteilte Partitionierung von Finite-Elemente-Modellen</i> Prüfer: Reuter (IPVR) Betreuer: Wörner (IPVR)
Röhl, Ronald	<i>Roco-Compiler zur Erzeugung eines Roboterprogramms</i> Prüfer: Lagally Betreuer: Schmid (IPA)

Ruhland, Michael	<i>Eine Beschreibungssprache für multimediale Interaktionsformen</i> Prüfer: Gunzenhäuser Betreuer: Weber
Sander, Frank	<i>Entwicklung massiv paralleler Verfahren zur Kantenerkennung und Bildsegmentierung</i> Prüfer: Baitinger (IPVR) Betreuer: Bräunl (IPVR)
Schenk, Michael	<i>Graphische Benutzeroberfläche für das PROTOS-L System</i> Prüfer: Gunzenhäuser Betreuer: Jasper (IBM)
Schmidt, Petra	<i>Entwurf und Integration eines Realizers mit Perspektiven in CLASSIFY</i> Prüfer: Lehmann Betreuer: Rathke
Schmidt, Stefan	<i>Ein Gateway zwischen zwei Netzwerken für das Betriebssystem OS/2</i> Prüfer: Eggenberger Betreuer: Kerker (HP)
Schönherr, Iris	<i>Übertragung von Zeichnungsdaten von einem CAD-System auf einen Macintosh-Rechner und Aufbereitung der Daten für die Steuerung eines Schneideroboters</i> Prüfer: Ebert (AEG-Ulm) Betreuer: Riedl (Haaf GmbH)
Schwanzer, Ingrid	<i>Entwurf einer problemangepaßten Datenbank sowie einer graphischen Benutzeroberfläche zur Verwaltung und Bearbeitung von Geometrie- und Auftragsdaten</i> Prüfer: Ebert (AEG-Ulm) Betreuer: Riedl (Haaf GmbH)
Schwenkreis, Friedemann	<i>Zustandsgrößenverwaltung in einem Referenzmodell für numerische Berechnungen</i> Prüfer: Reuter (IPVR) Betreuer: Münch (ISD)
Siehr, Hans-Jürgen	<i>Ähnlichkeitsmaße beim Lernen durch Analogie</i> Prüfer: Lehmann Betreuer: Tausend

Stahl, Irene	<i>Induktion von disjunktiven Konzepten</i> Prüfer: Lehmann Betreuer: Tausend, Wirth (University of California)
Staiger, Ralph-Christoph	<i>Implementierung einer Parallelen Deduktiven Datenbank — Implementierung der Magic Sets</i> Prüfer: Reuter (IPVR) Betreuer: Duppel (IPVR)
Veigel, Andreas	<i>Rotations- und translationssinvariante Erkennung handschriebener Zeichen mit neuronalen Netzwerken</i> Prüfer: Reuter (IPVR) Betreuer: Zell (IPVR)
Waidelich, Andreas	<i>Entwicklung und Implementierung einer Termbeschreibungssprache zur Wissensrepräsentation</i> Prüfer: Lehmann Betreuer: Rathke
Werkmann, Hubert	<i>Schaltungsentwurf und Layout eines Assoziativspeichers mit einem 3D-CMOS-Prozessor</i> Prüfer: Baitinger (IPVR) Betreuer: Roos (IME), Rettig (IPVR)
Wichert, Michael	<i>COUSIN Ein Baukasten für Benutzerschnittstellen</i> Prüfer: Gunzenhäuser Betreuer: Herczeg
Wimpff, Roland	<i>Konzeption und Implementierung eines Kommunikationsbausteins für die Leittechnik flexibler Fertigungssysteme (FFS) auf Basis der international genormten Protokollarchitektur MAP 3.0 (Manufacturing Automation Protocol)</i> Prüfer: Eggenberger Betreuer: Fischer (Mercedes-Benz)
Winnewisser, Roger	<i>Effiziente Implementierung von “Traveling Salesman” Algorithmen auf Personalcomputern</i> Prüfer: Lagally Betreuer: Seeland (debis Systemhaus GmbH)
Wittmann, Holger	<i>Ein Beispiel zur wissensbasierten Produktionsplanung mit PROTO-S-L</i> 1. Prüfer: Schönfeld (IBM Heidelberg) 2. Prüfer: Lehmann Betreuer: Beierle (IBM Stuttgart)

Wollensack, Uta *XTRACT — Ein graphisches Trace-Werkzeug für die Programmiersprache CLOS*
Prüfer: Gunzenhäuser
Betreuer: Herczeg

Zimmerer, Peter *Translations- und rotationsinvariante Erkennung von Werkstücken mit neuronalen Netzen*
Prüfer: Reuter (IPVR)
Betreuer: Zell (IPVR)

Ist bei den Diplomarbeiten nur ein Prüfer benannt, so hat dieser auch die jeweiligen Arbeiten betreut.
Prüfer oder Betreuer ohne Zusatzangabe sind Angehörige des IFI.

3.3.4.3 Studienarbeiten

Andreas, Anita *Darstellung allgemeiner, sich dynamisch ändernder Hypergraphen*
Prüfer: Reuter (IPVR)
Betreuer: Zink (IPVR)

Andree, Ulrike *Entwurf und Realisierung eines Formularbausteins für die Erstellung von NC-Bedienoberflächen*
Prüfer: Storr (ISW)
Betreuer: Junghans (ISW)

Balassa, Friedrich *Integration moderner Window-Techniken in das ISW-OFF-line-Programmsystem*
Prüfer: Storr (ISW)
Betreuer: Angerbauer (ISW)

Balassa, Renate *EFEU — Eine Formularentwicklungsunggebung*
Prüfer: Lehmann
Betreuer: Rathke

Baumgärtel, Jochen *Belastungsorientierte Auftragsfreigabe in einer Fertigungssteuerung*
Prüfer: Böhm (IPVR)
Betreuer: Dammert (IPVR)

Bayer, Harald *Entwurf und Implementierung einer Präsentationskomponente für ψ -Terme*
Prüfer: Reuter (IPVR)
Betreuer: Zell (IPVR)

Behrens, Marcus *Mascue — Ein System zum Experimentieren mit Musikmimen*
Prüfer: Gunzenhäuser
Betreuer: Mahling

Bergfeld, Stephan *Electronic Mail als Anwendung moderner Informations- und Kommunikationstechnologien: Beurteilung unter ethischen Gesichtspunkten*
Prüfer: Gunzenhäuser

Böckh, Matthias *Benutzerorientierte Implementierung der graphischen Transformationen von PHIGS*
Prüfer: Grieger (ISD)

Brahim, Kais	<i>Dünn verbundene neuronale Netze</i> Prüfer: Reuter (IPVR) Betreuer: Zell (IPVR)
Brunner, Jürgen	<i>Programm zur interaktiven Fahrplangestaltung und -abstimmung in Netzen des öffentlichen Verkehrs</i> 1. Prüfer: Knödel 2. Prüfer: Heimerl (EUV) Betreuer: Dobeschinsky (EUV), Schedel (EUV)
Büttner, Axel	<i>Zentrale Ereignisbearbeitung in einem größeren Informationssystem</i> Prüfer: Eggenberger Betreuer: Siegel (Siemens AG)
Clement, Hans-Georg	<i>PINKI — Ein wissensbasiertes Werkzeug zur Generierung von Fingersätzen; Problemlösekomponente</i> Prüfer: Gunzenhäuser Betreuer: Mahling
Czech, Hans	<i>Konzeption und objektorientierter Entwurf eines Projektplanungs-Werkzeugs</i> Prüfer: Ludewig Betreuer: K. Schneider
Dettlaff, Berthold	<i>Aufbau graphischer Benutzeroberflächen unter ergonomischen Gesichtspunkten</i> Prüfer: Böhm (IPVR) Betreuer: Dammert (IPVR)
Dirlewanger, Martin W.	<i>Statusänderungen des Microprozessors 80286 mit dem Befehl LOADALL</i> Prüfer: Eggenberger
Dorna, Michael	<i>Parser-Steuertabellen für Unifikationsgrammatiken</i> Prüfer: Rohrer (IMS) Betreuer: Eisele (IMS)
Eck, Oliver	<i>ALAN — eine assertionale Komponente zur Wissensrepräsentation</i> Prüfer: Lehmann Betreuer: Burkert, Forster

Finkbeiner, Albrecht	<i>Generierung von SILOS-Netzlisten aus Register-Transfer-Beschreibungen ausgewählter Bauelemente</i> Prüfer: Burkhardt Betreuer: Zipperer
Friedrich, Jürgen	<i>Entwicklung einer DMA-Einheit für Transputer-Module zur schnellen Übertragung großer Datenmengen</i> Prüfer: Burkhardt Betreuer: Walter
Gotin, Michael	<i>Inbetriebnahme und funktionelle Erweiterung eines Rad-schlupfprozessors</i> Prüfer: Baitinger (IPVR) Betreuer: Rettig (IPVR), Büchner (IME)
Hasan, Michael	<i>Entwurf und Implementierung eines Visualisierungswerkzeugs zur Leistungsanalyse von parallelen Programmen</i> Prüfer: Reuter (IPVR) Betreuer: J. Maier (IPVR)
Heiden, Elke	<i>Entwicklung eines Werkzeuges zur Analyse des dynamischen Verhaltens von Software-Architektur-Prototypen</i> Prüfer: Ludewig Betreuer: Schwille
Herrmann, Frank	<i>Erlernen von Grammatiken mit Genetischen Algorithmen</i> Prüfer: Lehmann Betreuer: Tausend
Herrmann, Kai-Uwe	<i>Methoden zur Generierung anwendungsspezifischer Testdaten am Beispiel einer Hoteldatenbank</i> Prüfer: Reuter (IPVR)
Hoch, Edgar	<i>Effiziente Implementierung des DRS-Konstruktions-algorithmus für ein Fragment der Diskurs-Repräsentations-Theorie durch Einsatz geeigneter Datenstrukturen</i> Prüfer: Rohrer (IMS) Betreuer: Reyle (IMS)
Hofmann, Eckhard	<i>Entwicklung und Erprobung eines deutschsprachigen automatischen Dialogpartners</i> Prüfer: Gunzenhäuser Betreuer: Mahnkopf (Btx Südwest-Datenbank GmbH Stuttgart)

Hofmann, Peter E.	<i>Ein HyperCard-basiertes Informationssystem der Fakultät Informatik</i> Prüfer: Reuter (IPVR) Betreuer: Zell (IPVR)
Holzmüller, Bernd	<i>SESAM — ein Prototyp zur Simulation von Software-Projekten</i> Prüfer: Ludewig Betreuer: K. Schneider
Jahke, Thilo	<i>Erweiterung des Optimierers im CSQL-System, einem System zur Bearbeitung eingebetteter Prozeduren (Interpolationsroutinen in SQL-Anfragen)</i> Prüfer: Reuter (IPVR) Betreuer: Neugebauer (IPVR)
Keller, Wolfgang	<i>Modellierung von Transportvorgängen mit dem TP-Monitor VAX ACMS</i> Prüfer: Reuter (IPVR) Betreuer: Schmidt (IPVR), Winckler (IPVR)
Kellerhoff, Holger	<i>Leistungsmessung am Filestone MS-DOS</i> Prüfer: Eggenberger Betreuer: Berger
Ketelhut, Oliver B.	<i>Durchschnittskomplexität</i> Prüfer: Ebinger
Kiehne, Klaus	<i>PINKI — ein wissensbasiertes Werkzeug zur Generierung von Fingersätzen; Dialogkomponente</i> Prüfer: Gunzenhäuser Betreuer: Mahling
Knoll, Rolf Pfeilsticker, Konrad	<i>A system vor the manipulation and analysis off difference operators</i> Prüfer: Reuter (IPVR) Betreuer: Thune (Universität Uppsala), Wahlund (Universität Uppsala)
Kolodziej, Harald	<i>Automatische Generierung von Bedienpulten an Montageanlagen</i> 1. Prüfer: Ebert (AEG-Ulm) 2. Prüfer: Bullinger (IAO) Betreuer: Hengel (IAO)

Koser, Stefan	<i>Eine Sprachausgabekomponente für ein auf einer abstrakten Spezifikationstabelle basierendes UIMS</i> Prüfer: Gunzenhäuser Betreuer: Diel (IBM)
Kurz, Hermann	<i>Realisierung eines Prozessüberwachungs- und Steuerungssystems für eine Fertigungszelle unter Verwendung des MAP/MMS-3.0-Protokolls und Ankopplung des Systems an nicht MAP/MMS-fähige Komponenten</i> Prüfer: Eggenberger Betreuer: Kämpf (IPA), Haag (IPA)
Lang, Friedger	<i>Generierung von Druckservicekonfigurationen im Netzwerk</i> Prüfer: Knödel Betreuer: Heller (Hewlett Packard)
Levec, Tatjana	<i>Literaturanalyse von Methoden zur Verwaltung replizierter Daten in verteilten Datenbanksystemen</i> Prüfer: Reuter (IPVR)
Lockowandt, Georg Alexander	<i>Simulation der Leistungsvorhersage von verteilten Systemen</i> Prüfer: Reuter (IPVR) Betreuer: Haban (Mercedes-Benz)
Mast, Klaus	<i>Erweiterungen von Grammatiken durch analoges Lernen</i> Prüfer: Lehmann Betreuer: Tausend
Matsentidis, Stylianos	<i>Entwicklung und Implementierung eines Simulationswerkzeugs für ein wissensbasiertes Entscheidungsunterstützungssystem</i> Prüfer: Gunzenhäuser Betreuer: Zahn (BWL), Greschner (BWL)
Mercz, Thomas	<i>EPSON-LQ PREVIEW</i> Prüfer: Eggenberger
Monschau, Jörg	<i>Programmierung mit X am Beispiel einer Dateiauswahlbox und eines Colormap-Editors</i> Prüfer: Böhm (IPVR) Betreuer: Dammert (IPVR)

Müller, Helmut	<i>Untersuchung des Transaktionsverhaltens für verteilte Anwendungen in einer Client-Server-Umgebung unter dem Betriebssystem UNIX</i> Prüfer: Eggengerger Betreuer: Rauh (Hewlett Packard)
Nakott, Dirk	<i>Entwurf und Implementierung eines Interpreters für grafisch erzeugte Arbeitspläne</i> Prüfer: Reuter (IPVR) Betreuer: Winckler (IPVR), R. Günthör (IPVR)
Nothdurft, Thomas	<i>Aufbau eines multifunktionalen Experimentierboards für Logic Cell Arrays (LCA)</i> Prüfer: Baitinger (IPVR) Betreuer: Rettig (IPVR)
Olderdissen, Jan	<i>Ein Konvertierungsprogramm für Wahrheitstabellen in das PALASM Format</i> Prüfer: Burkhardt Betreuer: Zipperer
Petzold, Andrea	<i>Vergleich verschiedener Lernverfahren für neuronale Netze</i> Prüfer: Reuter (IPVR) Betreuer: Zell (IPVR)
Philipp, Holger	<i>X-BOF-Editor — ein Designwerkzeug für Benutzeroberflächen</i> Prüfer: Gunzenhäuser Betreuer: Brunn (DELTA Industrieinformatik GmbH Fellbach)
Pitz, Joachim	<i>Darstellung und Konfigurierung paralleler Strukturen in verteilten Netzen unter X-Windows</i> Prüfer: Reuter (IPVR) Betreuer: Winckler (IPVR)
Reich, Volker	<i>Erzeugung von Flussbildern aus Energiebilanzen</i> Prüfer: Eggengerger Betreuer: Hanselmann (IER)
Reichert, Matthias Faulhaber, Volker	<i>Entwurf und Implementierung einer Meßumgebung für ein Multiprozessorsystem</i> Prüfer: Reuter (IPVR) Betreuer: Schiele (IPVR)

Reinauer, Georg	<i>Entwicklung eines interaktiven graphischen Editors für den Entwurf von Software-Architekturen</i> Prüfer: Ludewig Betreuer: Lichter
Rüdiger, Klaus	<i>Entwurf eines Benchmarks für Datenbanksysteme im Laborbereich</i> Prüfer: Reuter (IPVR) Betreuer: Sauer (Friedrich & Co)
Scheidel, Herwig-Franz	<i>Wissensbasierte Auswahl und Anordnung von Teilebehältern an manuellen Arbeitsplätzen</i> Prüfer: Ebert (AEG-Ulm) Betreuer: Fischer (IFF), Matthes (IFF)
Scherschel, Christian	<i>Integration von Gedächtnistests beim rechnerunterstützten Vokabellernen</i> Prüfer: Gunzenhäuser
Schiller, Anne	<i>Architektur und Implementierungsaspekte einer Entwicklungsumgebung zur Erstellung von 2-Ebenen-Morphologien</i> Prüfer: Rohrer (IMS) Betreuer: Steffens (IBM)
Schmalz, Michael	<i>Rotations- und translationsinvariante Erkennung maschinengeschriebener Zeichen mit neuronalen Netzen</i> Prüfer: Reuter (IPVR) Betreuer: Zell (IPVR)
Schmid, Peter	<i>Vergleichende Untersuchung und Anwendung von „Tool-Kits“ zur Spezifikation von Benutzeroberflächen</i> Prüfer: Gunzenhäuser
Schmidt, Dieter	<i>Anwendung neuronaler Netzwerkmodelle zur Erkennung und Klassifikation exogener und endogener Komponenten ereigniskorrelierter hirnelektrischer Potentiale</i> Prüfer: Reuter (IPVR) Betreuer: Zell (IPVR), Miltner (Uni-Tübingen)
Schmidt, Petra	<i>Ein Chartparser für eine feature-orientierte Grammatik</i> Prüfer: Lehmann Betreuer: Rathke

Schütz, Holger	<i>Darstellung und Verwaltung von Beziehungen zwischen Transaktionen</i> Prüfer: Reuter (IPVR) Betreuer: Wächter (IPVR)
Schumacher, Guido	<i>Trennung stochastischer von deterministischen Signalkomponenten in akustischen Signalen</i> Prüfer: Böhm (IPVR) Betreuer: Dammert (IPVR)
Seibold, Wolfram	<i>Entwicklung einer DMA-Einheit für Transputer-Module zur schnellen Übertragung großer Datenmengen</i> Prüfer: Burkhardt Betreuer: Walter
Sienel, Jürgen	<i>Kompensation von Störgeräuschen in Spracherkennungssystemen mittels Neuronaler Netze</i> Prüfer: Reuter (IPVR) Betreuer: Hackbarth (SEL), Zell (IPVR)
Staib, Rainer	<i>Vorhersage der Lastfaktoren von Sort-Merge Joins auf einem Multiprozessorsystem</i> Prüfer: Reuter (IPVR) Betreuer: Duppel (IPVR)
Stechow, Arne	<i>Entwurf und Implementierung eines graphischen Auswertungsprogrammes für akustische Meßdaten des Echtzeit-Frequenzanalysatorss B & K 2133</i> Prüfer: Bullinger (IFF) Betreuer: Lauster (IFF)
Stöckle, Martin	<i>Anbindung des Zellrechners einer Fertigungszelle an das Fertigungssteuerungssystem mittels standardisierter Protokolle sowie Umsetzung der Zellrechner-Aktionen auf MMS-Dienste</i> Prüfer: Eggenberger Betreuer: Kämpf (IPA), Haag (IPA)
Trippner, Ruth	<i>Konzeption und Implementierung einer neutralen Benutzeroberfläche eines wissensbasierten Simulationssystem</i> Prüfer: Gunzenhäuser Betreuer: Hartfuss (IPA)
Vogt, Michael	<i>Simulation einer Extended Warren Abstract Machine</i> Prüfer: Lagally Betreuer: Schimpf

Wahl, Peter	<i>Interpretative Abwicklung von Prädikat-Transitionsnetzen</i> Prüfer: Reuter (IPVR) Betreuer: Wächter (IPVR)
Waldmann, Thomas	<i>Entwurf und Aufbau eines Busadapters von MC68000 auf AT-Bus am Beispiel einer VGA-Karte</i> Prüfer: Burkhardt Betreuer: Zipperer
Walter, Volker	<i>Entwurf von massiv parallelen Simulated Annealing Algorithmen</i> Prüfer: Ludewig Betreuer: Bräunl (IPVR)
Walter, Markus	<i>Morphologische und phonologische Information im Stuttgarter LFG-System</i> Prüfer: Rohrer (IMS) Betreuer: Momma (IMS)
Weiß, Beate	<i>Ein HyperCard-basiertes Informationssystem über die Stadt Stuttgart</i> Prüfer: Reuter (IPVR) Betreuer: Zell (IPVR)
Weiß, Christoph	<i>Erstellung von Gedächtnistests für rechnerunterstützte Lernmethoden</i> Prüfer: Gunzenhäuser
Wendt, Wolfgang	<i>Entwurf und Implementierung einer relationalen Datenbank für die Konfigurationsverwaltung eines verteilten Rechnernetzes</i> Prüfer: Baitinger (IPVR) Betreuer: Kropf (IPVR)
Wenzler, Gerhard	<i>Gestaltung einer Benutzeroberfläche unter PHIGS und FORTRAN</i> Prüfer: Gunzenhäuser Betreuer: Heid (IMS)
Wohnhaas, Klaus	<i>Ein System für intelligente Dokumentkonversion zwischen MS Word und LATEX</i> Prüfer: Reuter (IPVR) Betreuer: Zell (IPVR)

Zander, Olaf

Entwicklung eines Simulators und Debuggers für den ausfallsicheren RISC-Prozessor RISC-16 des Instituts für Mikroelektronik Stuttgart
Prüfer: Burkhardt
Betreuer: Zipperer

Ist bei den Studienarbeiten nur ein Prüfer benannt, so hat dieser auch die jeweiligen Arbeiten betreut.
Prüfer oder Betreuer ohne Zusatzangabe sind Angehörige des IFI.

Abkürzungen

ACM	Association for Computing Machinery
AIB	Angewandte Informatik für Blinde (Forschungsgruppe)
AK	Arbeitskreis
ALWR	Arbeitskreis der Leiter wissenschaftlicher Rechenzentren
ASK	Akademische Software Kooperation
BMBW	Bundesministerium für Bildung und Wissenschaft
CSLG	Campuswide Software Licence Grant Program
DA	Dienstauftrag
BMFT	Bundesministerium für Forschung und Technologie
DFG	Deutsche Forschungsgemeinschaft
DLR	Deutsche Forschungsanstalt für Luft- und Raumfahrt
EATCS	European Association for Theoretical Computer Science
EBRA	Esprit Basic Research Action
EUV	Institut für Eisenbahn- und Verkehrswesen
FG	Fachgruppe
GI	Gesellschaft für Informatik
GID	Gesellschaft für Information und Dokumentation
GMD	Gesellschaft für Mathematik und Datenverarbeitung
HBFG	Hochschulbauförderungsgesetz
HP	Hewlett Packard
IAO	Institut für Arbeitswissenschaft und Organisation
IEEE	Institute of Electrical and Electronics Engineers
IER	Institut für Energiewirtschaft und Rationelle Energieanwendung
IEV	Institut für Eisenbahn- und Verkehrswesen
IFF	Institut für Industrielle Fertigung und Fabrikbetrieb
IFI	Institut für Informatik
IFIP	International Federation of Information Processing
IGDD	Interest Group of Distributed Data
ILR	Institut für Luft- und Raumfahrt
IME	Institut für Mikroelektronik Stuttgart
IMS	Institut für Maschinelle Sprachverarbeitung

IMG	Institut für Maschinenelemente und Gestaltungslehre
IPA	Institut für Produktionstechnik und Automatisierung
IPE	Institut für Physikalische Elektrotechnik
IPVR	Institut für Parallele und Verteilte Höchstleistungsrechner
ISD	Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen
ISO	International Organization for Standardization
ISW	Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen
IVS	Informatik–Verbund Stuttgart
LA	Lehrauftrag
MWK	Ministerium für Wissenschaft und Kunst Baden Würtemberg
NTG	Nachrichtentechnische Gesellschaft
RUS	Rechenzentrum der Universität Stuttgart
SC	Subcommittee
SEL	Standard Elektrik Lorenz
SFB	Sonderforschungsbereich
SI	Schweizerische Informatikergesellschaft
SIG	Special Interest Group
SIGCHI	Special Interest Group Computer Human Interaction
TC	Technical Committee
WG	Working Group