



# **JAHRESBERICHT**

## **1989**

**Universität Stuttgart**  
**Institut für Informatik**

Azenbergstraße 12 7000 Stuttgart 1

Herausgeber: Vorstand des Instituts für Informatik  
Institut für Informatik  
Azenbergstraße 12  
7000 Stuttgart 1

Redaktion: Bernhard Ziegler Tel: (0711) 121-1365

Institutsadressen:

Azenbergstraße 12,	7000 Stuttgart 1	Tel: (0711) 121-1379
Herdweg 51,	7000 Stuttgart 1	Tel: (0711) 121-1384
Forststraße 86,	7000 Stuttgart 1	Tel: (0711) 121-1430

# Inhaltsverzeichnis

<b>Das Jahr 1989</b>	<b>1</b>
<b>1 Selbstverwaltung und Organisation</b>	<b>3</b>
1.1 Leitung des Instituts für Informatik . . . . .	3
1.2 Mitarbeit in Gremien . . . . .	4
1.2.1 Universität Stuttgart . . . . .	4
1.2.2 Wissenschaftliche Institutionen . . . . .	6
<b>2 Fakultätsbezogene Aufgaben</b>	<b>9</b>
2.1 Dekanat der Fakultät Informatik . . . . .	9
2.2 Zentrale Fakultätseinrichtungen . . . . .	9
2.2.1 Bibliothek . . . . .	9
2.2.2 Rechnernetz . . . . .	10
2.2.3 Mehrbenutzersysteme . . . . .	11
2.2.4 Workstations . . . . .	11
2.2.5 PC-Pools . . . . .	12
2.3 Lehre . . . . .	13
2.3.1 Aufbau des Informatikstudiums . . . . .	13
2.3.1.1 Diplomstudiengang Informatik . . . . .	13
2.3.1.2 Nebenfachstudium Informatik . . . . .	14
2.3.1.3 Weitere informatikorientierte Studiengänge . . . . .	14
2.3.2 Lehrveranstaltungen . . . . .	15
2.3.3 Informatik-Kolloquium . . . . .	22
2.3.4 Habilitationen und Examensarbeiten . . . . .	25
2.3.4.1 Habilitationen . . . . .	25

2.3.4.2	Dissertationen . . . . .	25
2.3.4.3	Diplomarbeiten . . . . .	27
2.3.4.4	Studienarbeiten . . . . .	35
<b>3</b>	<b>Forschung</b>	<b>45</b>
3.1	Forschungsvorhaben und Forschungsgruppen . . . . .	45
3.1.1	Abteilung Betriebsssoftware . . . . .	45
3.1.2	Abteilung Computer–Systeme . . . . .	50
3.1.3	Abteilung Dialogsysteme . . . . .	54
3.1.4	Abteilung Intelligente Systeme . . . . .	63
3.1.5	Abteilung Kombinatorik und Numerische Verfahren . . . . .	70
3.1.6	Abteilung Programmiersprachen und ihre Übersetzer . . . . .	73
3.1.7	Abteilung Software Engineering . . . . .	80
3.1.8	Abteilung Theorie der Informatik . . . . .	82
3.2	Veröffentlichungen . . . . .	85
3.3	Berichte . . . . .	92
3.4	Vorträge . . . . .	94
3.5	Tagungen . . . . .	107
3.6	Herausgabe von Zeitschriften . . . . .	108
3.7	Implementierungen . . . . .	109
<b>Abkürzungen</b>		<b>113</b>

# Das Jahr 1989

Die Entwicklung der Informatik in der Universität Stuttgart war im Jahre 1989 durch mehrere wichtige Ereignisse geprägt:

Mit Beginn des Wintersemesters 1988/89 ging ein lange gehegter Wunsch der Stuttgarter Informatik in Erfüllung: Nach Inkrafttreten einer entsprechenden Grundordnungsänderung entstanden aus der bisherigen Fakultät Mathematik und Informatik zwei unabhängige Fakultäten. Die neu gegründete *Fakultät 14 Informatik* konnte in einem Festakt am 24. November 1988 unter Beisein des Ministers für Wissenschaft und Kunst und des Rektors der Universität Stuttgart aus der Taufe gehoben werden.

Unter dem Dach dieser neuen Fakultät konnte sich das schon 1988 von Professor Reuter konzipierte und initialisierte neue Institut für Parallele und Verteilte Höchstleistungsrechner (IPVR) tatkräftig entwickeln. Es hat sich im Frühjahr 1989 auch personell und wirtschaftlich vom Institut für Informatik (IfI) abgetrennt: Unter anderem sind die bisher zum IfI gehörigen Lehrstühle und Abteilungen *Anwendersoftware* (Professor Reuter), *Integrierter Systementwurf* (Professor Baitinger) und *Verteilte Systeme* (derzeit noch unbesetzt) an das IPVR übergegangen.

Das 1989 von Bundesbildungsminister Möllemann ins Leben gerufene *Ausbauprogramm für Hochschulen* besitzt einen Schwerpunkt im Bereich der Informatikausbildung. Es ist der Universität Stuttgart gelungen, im Rahmen dieses Programms zwei neue Lehrstühle für die Stuttgarter Informatik einzurichten: Einer dieser Lehrstühle soll am IfI schwerpunktmäßig die Ausbildung in *Grundlagen der Informatik* als Dienstleistung für andere Fakultäten der Universität übernehmen. Der andere soll voraussichtlich am IPVR eingerichtet werden und sich mit *Bildverständhen* in Forschung und Lehre befassen. Für beide Lehrstühle sind die Berufungsverfahren im Gang.

Durch diesen Ausbau hat sich die Ausbildungskapazität der Fakultät Informatik um jährlich etwa 25 zusätzliche Plätze für Studienanfänger erhöht. Sie kletterte zu Beginn des Wintersemesters 1989/90 auf die Rekordzahl von 215 Studienanfängerplätzen, aus denen aus über 550 Bewerbungen allerdings fast 300 neue Informatikstudenten aufgenommen werden mußten.

Eine solche nun schon seit fast 10 Jahren anhaltende Überlast kann von den Professoren, den Dozenten, den Honorarprofessoren und den Lehrbeauftragten sowie den Wissenschaftlichen Mitarbeitern nurmehr unter sehr hohem persönlichem Einsatz verkraftet werden. Erschwerend kommt hinzu, daß es der Stuttgarter Informatik auch 1989 nicht gelungen ist, die vakanten Lehrstühle für *Theorie der Informatik* (*Nachfolge Prof. Schwabhäuser*) und für *Programmiersprachen und ihre Übersetzer* (*Nachfolge Prof. Barth*) zu besetzen. Am Ende des Berichtsjahres bleiben damit von den neun zum IfI zählenden Lehrstühlen drei unbesetzt.

---

Trotz dieser personellen Engpässe konnten zum Wintersemester 1989/90 zwei neue Dienstleistungslehrveranstaltungen aufgenommen werden: Grundlagen der Informatik für die Maschinenbaufakultäten (*Prof. Baitinger, IPVR*) und Grundlagen der Informatik für die Studiengänge der Luft- und Raumfahrttechnik, der Physik, des Vermessungswesens und der technischen Biologie (*Prof. Gunzenhäuser, IfI*).

Für die räumliche Unterbringung der bisher auf drei voneinander entfernt liegende Gebäude verteilten Informatikinstitute konnte eine Lösung gefunden und noch 1989 vertraglich abgesichert werden. Voraussichtlich im Oktober 1990 kann ein bisher von einem DV-Hersteller gemietetes Bürogebäude im Industriegebiet Stuttgart–Vaihingen bezogen werden. Die Erweiterung dieses Gebäudes um ein weiteres Stockwerk soll Anfang 1991 abgeschlossen sein, so daß dann alle Dozenten, Mitarbeiter und Studenten der Stuttgarter Informatik in einem gemeinsamen großzügigen Gebäude untergebracht sein werden. Diesem unbestreitbaren Vorteil, der auch einen Ausbau der Bibliothek und der studentischen Arbeitsplätze einschließt, steht die noch unbefriedigende Verkehrsanbindung entgegen. Davon sind insbesondere die Studierenden betroffen, weil sie — etwa im Rahmen ihres Nebenfachstudiums — zwischen den einzelnen Universitätszentren pendeln müssen.

Die Rechnersituation konnte 1989 im Bereich der Forschung des IfI dadurch verbessert werden, daß aus den Berufungsmitteln der Professoren Lehmann und Ludewig leistungsfähige Workstations beschafft werden konnten. Zur Bewältigung der Überlast in der Informatikausbildung wurden übergangsweise drei Rechnersysteme angemietet; an diese Server wurden 30 einfache PC als Benutzerarbeitsplätze angeschlossen.

Im Berichtsjahr konnten in der Abteilung Dialogsysteme drei Forschungsvorhaben erfolgreich abgeschlossen werden, die durch Drittmittel und im Rahmen einer Industriekooperation finanziert wurden. Im Rahmen eines ESPRIT-Projekts konnte in Zusammenarbeit mit der Firma SEL ein neues Vorhaben begonnen werden.

Im Jahr 1989 sind in der Stuttgarter Informatik auch die Vorbereitungen für die *20. Jahrestagung der Gesellschaft für Informatik* angelaufen, die die beiden Informatikinstitute im Oktober 1990 in Stuttgart durchführen werden; sie wird unter dem Thema „Informatik auf dem Weg zum Anwender“ stehen.

---

Februar 1990

Rul Gunzenhäuser

# 1 Selbstverwaltung und Organisation

## 1.1 Leitung des Instituts für Informatik

### Geschäftsführender Direktor

Prof. Dr. E. Lehmann ()

### Stellvertretender Geschäftsführender Direktor

Prof. Dr. R. Gunzenhäuser (bis 30.11.)  
Prof. Dr. J. Ludewig (ab 1.12.)

### Vorstand

Prof. Dr. W. Burkhardt  
Prof. Dr. R. Gunzenhäuser  
Prof. Dr. W. Knödel  
Prof. Dr. K. Lagally  
Prof. Dr. E. Lehmann  
Prof. Dr. J. Ludewig

### Geschäftsstelle

Akad. Oberrat D. Martin  
Frau E. Zahn (bis 30.9.)  
Frau I. Kolb (ab 1.11)

### Abteilungen und ihre Leiter

Betriebsssoftware	Prof. Lagally
Computer Systeme	Prof. Burkhardt
Dialogsysteme	Prof. Gunzenhäuser
Grundlagen der Informatik	N.N.
Intelligente Systeme	Prof. Lehmann
Kombinatorik und Numerische Verfahren	Prof. Knödel
Programmiersprachen und ihre Übersetzer	Prof. Ludewig (kommissarisch)
Software Engineering	Prof. Ludewig
Theorie der Informatik	Prof. Knödel (kommissarisch) Dr. Schmidt
(Lehrstuhlvertreter ab ??.)	

## 1.2 Mitarbeit in Gremien

### 1.2.1 Universität Stuttgart

<b>Böcker, H.-D.</b>	Mitglied im Prüfungsausschuß der Fakultät Informatik Mitglied im Fakultätsrat der Fakultät Informatik
<b>Burkhardt, W.H.</b>	Mitglied im erweiterten Fakultätsrat Vorsitzender der Studienkommission <i>Informatik</i> Mitglied der Berufungskommission <i>Praktische Informatik: Verteilte Systeme</i> Mitglied der Berufungskommission <i>Praktische Informatik: Integrierter Systementwurf</i> Mitglied der Berufungskommission <i>Theoretische Informatik</i> (bis Okt. 1989) Mitglied einer Berufungskommission der Fakultät Elektrotechnik
<b>Diekert, V.</b>	Mitglied des erweiterten Fakultätsrats <i>Informatik</i> (ab 1.10.)
<b>Eggenberger, O.</b>	Mitglied des Fakultätsrats <i>Informatik</i> Mitglied des Prüfungsausschusses <i>Informatik</i>
<b>Gunzenhäuser, R.</b>	Mitglied im Fakultätsrat der Fakultät Informatik Stellv. Vorsitzender der Studienkommission <i>Informatik</i> Mitglied des Großen Senats Mitglied des Prüfungsausschusses <i>Techn.-orientierter Diplomkaufmann</i> Mitglied der Berufungskommission <i>Bildverständen</i> Mitglied der Berufungskommission <i>Theoretische Informatik</i> Mitglied der Senatskommission <i>Koordinierungsausschuß Informatik</i>
<b>Knopik, Th.</b>	Mitglied im Fakultätsrat der Fakultät Informatik
<b>Knödel, W.</b>	Kooptiertes Mitglied des Fakultätsrats Mathematik Dekan der Fakultät Informatik (bis 30.9.) Prodekan der Fakultät Informatik (ab 1.10.) Mitglied des Großen Senats

	Mitglied des Senats (bis 30.9.)
	Vorsitzender der Berufungskommission für die C4–Professur <i>Theoretische Informatik</i> (Nachfolge Schwabhäuser)
	Mitglied der Senatskommission <i>Informatik</i>
	Mitglied der Kommission <i>Uni–Bau 2000</i>
<b>Krause, K.</b>	Mitglied des Fakultätsrats <i>Informatik</i>
<b>Lagally, K.</b>	Mitglied des Fakultätsrats <i>Informatik</i> Vorsitzender des Prüfungsausschusses <i>Informatik</i>
	Mitglied der Berufungskommission <i>Programmiersprachen und ihre Übersetzer</i>
	Mitglied der Berufungskommission <i>Theoretische Informatik</i>
	Mitglied der Berufungskommission <i>Verteilte Systeme</i>
	Mitglied der Berufungskommission <i>Bildverständen</i>
<b>Lehmann, E.</b>	Mitglied des Fakultätsrats <i>Informatik</i> Mitglied des Prüfungsausschusses <i>Informatik</i>
	Vorsitzender der Berufungskommission <i>Bildverständen</i>
	Mitglied der Berufungskommission <i>Programmiersprachen und ihre Übersetzer</i>
	Mitglied der Berufungskommission <i>Phonetik</i> (Fak. 11)
<b>Lichter, H.</b>	Mitglied der Berufungskommission <i>Programmiersprachen und ihre Übersetzer</i>
<b>Ludewig, J.</b>	Mitglied des Fakultätsrats <i>Informatik</i> Leiter der Software–Kommission der Fakultät Informatik
	Vorsitzender der Berufungskommission <i>Programmiersprachen und ihre Übersetzer</i>
	Mitglied der Berufungskommission <i>Verteilte Systeme</i>
<b>Rathke, C.</b>	Mitglied der Berufungskommission <i>Bildverständen</i>
<b>Reuß, W.</b>	Mitglied des Fakultätsrats <i>Informatik</i> Mitglied der Berufungskommission <i>Theoretische Informatik</i> (ab 1.9.)
<b>Tausend, B.</b>	Mitglied der Studienkommission <i>Informatik</i>
<b>Thierauf, T.</b>	Mitglied der Berufungskommission <i>Theoretische Informatik</i> (bis 31.8.)

---

## 1.2.2 Wissenschaftliche Institutionen

<b>Armbruster, D.</b>	Mitglied der <i>TeX User Group</i>
<b>Bodenschatz, W.</b>	Mitglied der GI-Fachgruppe 2.1.4 <i>Alternative Konzepte für Sprachen und Rechner</i>
	Mitglied der Fachgruppe 3.1.2 <i>Parallel-Algorithmen und -Rechnerstrukturen</i>
	Mitglied der Deutschen Occam-Interessengemeinschaft der Transputeranwender (DOIT)
	Mitglied der <i>OCCAM User Group</i>
<b>Burkhardt, W. H.</b>	Mitglied der GI/NTG Fachgruppe <i>Fehlertolerante Systeme</i>
	Mitglied der GI/NTG Fachgruppe <i>Rechnernetze</i>
<b>Diekert, V.</b>	Mitglied der EATCS
	Partner der EBRA-Working Group No 3166 <i>Algebraic and Syntactic Methods in Computer Science</i> (ASMICS)
<b>Gunzenhäuser, R.</b>	Mitglied im Fachbereich 7 <i>Ausbildung und Beruf</i> der GI
	Mitglied im Fachausschuß 7.1 <i>Informatik in Studiengängen an Hochschulen</i> der GI
	Stellv. Sprecher der Fachgruppe <i>Intelligente Lernsysteme</i> der GI
	Vorsitzender des Fakultätentages <i>Informatik</i>
	Mitglied von wissenschaftlichen Projektberatungsgremien an den Universitäten Hagen, Karlsruhe und Tübingen
<b>Knopik, Th.</b>	Mitglied der GI-Fachgruppe 1.5.1 <i>Knowledge Engineering</i>
<b>Knödel, W.</b>	titl.a.o.Prof. der TU Wien
	Gründungsmitglied der Ges.f.Informatik (GI)
	Mitglied der GI Fachgruppe 1.1.1 <i>Theoretische Informatik</i>
	Mitglied der <i>Europ.Ass.Theoretical Comp.Sc</i> (EATCS)
	Mitglied der <i>Ges.f.Mathematik, Oekonomie und Operations Research</i>
	Mitglied der Deutschen Mathematiker-Vereinigung
	Mitglied der Österr. Mathematischen Gesellschaft

	Mitglied der Österr. Statistischen Gesellschaft
	Mitglied der Gesellschaft für Angewandte Mathematik und Mechanik
<b>Kümmel, P.</b>	Mitglied der GI-Fachgruppe 1.1.1 <i>Theoretische Informatik</i> Mitglied der GI-Fachgruppe 1.2.5 <i>Sprachanalyse</i> Mitglied der GI-Fachgruppe 3.5.5 <i>Architekturen für hochintegrierte Schaltungen</i>
<b>Lagally, K.</b>	Mitglied des GI-Fachausschusses 3.2 <i>Systemarchitektur</i> Mitglied des GI-Arbeitskreises 3.1.2 <i>Architektur von Betriebssystemen</i>
<b>Lehmann, E.</b>	Stellvertretender Sprecher der Fachgruppe 1.2.2 <i>Expertensysteme</i> der GI Mitglied des VDI-Ausschusses <i>Künstliche Intelligenz</i> Mitglied der GI-Fachgruppe <i>Natürlichsprachliche Systeme</i> Mitglied der ACM, der „American Association for Artificial Intelligence“ und der “Association for Computational Linguistics“
<b>Lichter, H.</b>	Mitglied der GI Mitglied des GI-AK <i>Wissensbasierte Systeme für das Prototyping</i> im GI-FA 4.3
<b>Ludewig, J.</b>	Mitglied der GI, IEEE Computer Society, SI Stellvertretender Sprecher der GI-FG 4.3.1 <i>Requirements Engineering für die Automatisierung von Systemen</i> Mitglied der GI-FG 2.1.1 <i>Software Engineering</i>
<b>Rust, S.</b>	Mitglied der GI-Fachgruppe 3.5.1 <i>Methoden des Entwurfs und der Verifikation digitaler Schaltungen</i> Mitglied der GI-Fachgruppe 3.5.4 <i>CAD-Umgebungen für den Entwurf integrierter Schaltungen und Systeme</i>
<b>Schweikhardt, W.</b>	Mitglied im DIN-Ausschuß ATHBeh-5 (Ausschuß Technische Hilfen für Behinderte) <i>Kommunikationshilfen für sensorisch Behinderte</i>

---

---

Mitglied des Arbeitskreises *Informatik und Behinderte* der  
GI-Fachgruppe *Software-Ergonomie*

Mitglied im *APL-Club Germany*

Mitglied der GI-Fachgruppe *Rechnernetze*

**Strothotte, Th.** Mitglied der *Canadian Society for Computational Intelligence* und der *Canadian Information Processing Society*

Mitglied des Arbeitskreises *Informatik und Behinderte* der  
GI-Fachgruppe *Software-Ergonomie*

**Weber, G.** Mitglied des Arbeitskreises *Informatik und Behinderte* der  
GI-Fachgruppe *Software-Ergonomie* und Mitglied der GI-  
Fachgruppe *Personal Computing*

Mitglied der ACM-Fachgruppen *SIGCHI* und *SIGCAPH*

---

## 2 Fakultätsbezogene Aufgaben

## 2.1 Dekanat der Fakultät Informatik

Dekan

Prof. Dr. W. Knödel (bis 31.9.)  
Prof. Dr. A. Reuter (ab 1.10.)

## Sekretariat

Frau K. Erz

## 2.2 Zentrale Fakultätseinrichtungen

### 2.2.1 Bibliothek

Wissenschaftl. Beauftragter	<i>Ebinger</i>
Bibliothekarin	<i>Röger</i>
Programmbetreuung	<i>Schlebbe</i>
Wissenschaftl. Hilfskräfte	<i>Bauer, Bergdolt, Haag-Merz, Nguyen, Ranzinger, Ulm, vom Orde, Westermann</i>

Bedingt durch die starke Entwicklung der Bibliothek bei unveränderter räumlicher Ausstattung leidet die Bibliothek unter Platzmangel. Eine Präsentation des Bestandes, die als benutzerfreundlich bezeichnet werden kann, ist nicht mehr gegeben. Hier muß durch sinnvolle Ausstattung des Bibliotheksraumes im Rahmen des geplanten Umzuges Abhilfe geschaffen werden.

Um einen aussagekräftigen Sachkatalog zu ermöglichen, wurde im Berichtsjahr der Bibliotheksbestand in Bezug auf das aktuelle CR-Schema überarbeitet. Die Ergebnisse sind im Auskunftssystem IFIBIB enthalten. Die Überarbeitung bestimmter Sachgruppen steht noch aus, da bisher kein geeigneter Bearbeiter gefunden werden konnte.

Zur Erweiterung des Funktionsumfanges von IFIBIB wurde mit der Neuimplementierung in C für UNIX–Betriebssysteme auf der Basis eines relationalen Datenbanksystems mit der Anfragesprache SQL begonnen. Gegenüber dem derzeitigen IFIBIB–System ist so ein höheres Maß an Datensicherheit garantiert, werden differenziertere Zugangskontrollen sowie Parallelität von Online–Katalogisierung und –Recherche erst ermöglicht. Die zur Einführung von Dubletten– und Bandzählungskontrolle erforderliche Restrukturierung der Daten und Erweiterung der Zugriffsoperationen sind bereits abgeschlossen. Eine Komponente zur Verwaltung der Benutzerausleihe wurde integriert. Die Ausgestaltung einer fensterorientierten Benutzeroberfläche sowie die Bereitstellung von Transformationsmodulen zur Unterstützung verschiedener Datenaustauschformate (u.a. BibTEX, Süd–West–Verbund) sind derzeit noch in Bearbeitung.

## 2.2.2 Rechnernetz

Mitarbeiter

*F. Fabian, Sammet, Serafin*

Zum Jahresende 1989 waren ca. 200 Rechner unterschiedlicher Größenordnung an das auf Ethernet–Technologie basierende Rechnernetz der Informatik angeschlossen:

6	SUN Server (4/260, 4/370, 4/110)
4	HP Server (320)
5	Mehrbenutzersysteme (HP 835, 840)
54	SUN Arbeitsplatzrechner (3/50, 3/60, 3/80, 4/60)
12	MAC II Arbeitsplatzrechner
4	Cadmus Arbeitsplatzrechner
4	DEC VaxStation 3100
9	Apollo Server und Arbeitsplatzrechner
7	Lisp–Maschinen (Symbolics, TI–Explorer)
90	Personal Computer (HP Vectra, PC–AT)
4	X–Terminals (NCD)
3	Terminalserver

Hinzu kommen noch ca. 50 Terminals, die mittels Terminalserver Zugang zu den Rechnern am Informatik–Netz haben.

Auf allen für die Lehre verwendeten Rechnern (Ausnahme: PCs) wird als Betriebssystem UNIX eingesetzt.

Das Informatik–Rechnernetz wurde im Jahre 1989 weiter ausgebaut:

- Anschluß des Standorts Forststraße über Glasfaser–Leitung (10 Mbit Ethernet).
  - Anschluß an das Universitätsnetz über einen eigenen Gateway–Rechner (zuvor direkter Anschluß).
-

- Unterteilung des Netzes in Unternetze mittels Ethernet–Bridges bzw. Gateway–Rechner. Dies war wegen der stark ansteigenden Netzwerkbelaustung notwendig geworden.

Zusätzlich wurden noch einige abteilungs–interne Netze auf Basis von Token–Ring bzw. Appletalk installiert.

Über das Rechnernetz der Universität Stuttgart bestehen jetzt Kommunikationsmöglichkeiten mit anderen Universitäten in Baden–Württemberg (BelWue Rechnernetz) sowie weiteren nationalen und internationalen Netzen (seit Ende November 1989 auch in die USA) auf der Basis des IP/TCP Protokolls.

Die Informatik besitzt seit diesem Jahr einen eigenen Anschluß an das *Electronic Mail System* des DFN (Deutsches Forschungsnetz), der Zugang zu anderen, weltweit nutzbaren E–Mail Systemen wurde verbessert (Mail–Adressen: *user@informatik.uni-stuttgart.dbp.de* bzw. *user@informatik.uni-stuttgart.de*).

Bei der lokalen Nutzung des Rechnernetzes hat die Verwendung von *X Windows* und darauf basierenden Programmen stark zugenommen.

### 2.2.3 Mehrbenutzersysteme

Mitarbeiter

*Kohl, Finger, H. Schneider, Wahi*

Als Nachfolger der 1982 beschafften 2 VAXen (diese wurden 1989 stillgelegt) wurden für die Ausbildung im Grundstudium 4 HP–Mehrbenutzersysteme (Typ HP 9000/835, je 32 MB Hauptspeicher, 1,1 GB Plattenspeicher) angemietet und in Betrieb genommen. Der Zugang zu diesen erfolgt im wesentlichen mittels Terminals (40 Atari 1040ST) und Terminalserver.

### 2.2.4 Workstations

Mitarbeiter

*M. Fabian*

Für die Ausbildung im Hauptstudium wurden 1989 zusätzlich zu den schon vorhandenen SUN Arbeitsplatzrechnern 4 SUN Mehrbenutzersysteme (Typ 4/260, je 24 MB Hauptspeicher, 1,2 GB Plattenspeicher) in Betrieb genommen, die als Plattenserver für die schon bisher verwendeten Arbeitsplatzrechner und zur Bearbeitung rechenintensiver Programme eingesetzt werden.

---

## 2.2.5 PC-Pools

Beauftragter	<i>Hersmann</i>
Wissenschaftl. Hilfskräfte	<i>Bächtle, Grau, Hofmann, Ly Huynh, Mardassi, Milan, Nguyen, Nietsche, Schäfer, Sing, Schwenzer, Stümpfig, Thaler, Utz, Waigand, Waitzmann F., Waitzmann H., Wenzler</i>

Im Jahre 1988 wurde auf den vorhandenen 4 PC-Servern die Netzwerksoftware Version 1.0 von Office Share durch die Version 3.0 ersetzt. Dabei wurden drei Pools so modifiziert, daß es nun einen großen Pool mit drei Servern gibt. Der Vorteil ist: alle 30 Benutzer können auf alle drei Server und auf alle drei Drucker zugreifen.

Es wurde folgende Software installiert:

Arity-Prolog	Prolog	C-Compiler
FrameWork	MS-Pascal	

Im Jahre 1989 erhielten 30 Arbeitsplätze des PC-Pools graphikfähige Monitore und eine Harddisk von 20 MB, die zur Lösung von Aufgaben mit Graphik erforderlich sind. Vier PC-Server mit 40 PC-Arbeitsplätzen wurden an das ThinLan-Netz des Instituts angeschlossen.

Es wurde folgende Software installiert:

Modula2	Promod	Turing
FrameWork 3.0 LAN	Turbo Pascal 5.0	Turbo C 2.0

Lernprogramme von der Fernuniversität Hagen:

LISP	PROLOG	UNIX
------	--------	------

In den Jahren 1988 und 1989 wurden Praktika zu den Vorlesungen *Einführung in die Informatik I*, *Grundlagen der Informatik* und *Einführung in die Informatik III* sowie verschiedene Software-Praktika, Studienarbeiten und Diplomarbeiten auf diesen Rechnern durchgeführt.

## 2.3 Lehre

### 2.3.1 Aufbau des Informatikstudiums

#### 2.3.1.1 Diplomstudiengang Informatik

Der seit dem Wintersemester 1974/75 gültige, im Jahre 1989 auch im Bereich des Hauptstudiums aktualisierte Studienplan sieht nach der Diplomvorprüfung 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 kommt 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/Verkehrswesen
- 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.

---

---

Zum Wintersemester 1990/91 gab es für 215 Plätze für Studienanfänger 427 Bewerber, von denen 260 mit dem Informatikstudium in Stuttgart beginnen konnten. Die übrigen wurden von der Zentralstelle zur Vergabe von Studienplätzen an andere Universitäten verwiesen. Mit dieser hohen Anfängerzahl studieren nunmehr in Stuttgart mehr als 1400 Studierende im Diplomstudiengang Informatik.

Bei derzeit zehn besetzten Stellen für Professoren — fünf Stellen sind unbesetzt — muß ein größerer Teil der Informatik–Lehre durch die Vergabe von Lehraufträgen an auswärtige Lehrbeauftragte und an wissenschaftliche Mitarbeiter des Instituts abgedeckt werden.

### **2.3.1.2 Nebenfachstudium Informatik**

Informatik wird als Nebenfach bzw. technisches Schwerpunkt fach von zahlreichen Studenten in den Studiengängen „technisch–orientierter Diplomkaufmann“ und Mathematik sowie im Studiengang Linguistik gewählt.

Die Fakultät Informatik übernimmt dazuhin die Ausbildung in „Grundlagen der Informatik“ für die Studiengänge „technisch–orientierter Diplomkaufmann“, Mathematik, Luft– und Raumfahrt, Vermessungswesen, technische Biologie und Physik sowie — seit dem Wintersemester 1989/90 — auch für die Studiengänge der Maschinenbau–fakultäten.

### **2.3.1.3 Weitere informatikorientierte Studiengänge**

An der Universität Stuttgart gibt es weitere Ausbildungsgänge mit Informatikanteilen:

1. ein Studienmodell *Ingenieur–Informatik* mit Abschluß Dipl.–Ing. in der Fakultät für Elektrotechnik,
2. ein Studienmodell *Angewandte Informatik* in den Studienrichtungen Physik–ingenieurwesen und Fertigungstechnik des Maschinenbaustudiums,
3. ein Studienmodell *Luft– und Raumfahrttechnik / Datenverarbeitung* mit dem Abschluß Dipl.–Ing. in der Fakultät für Luft– und Raumfahrttechnik.

## 2.3.2 Lehrveranstaltungen

### Lehre im Sommersemester 1989

Grundlagen der Informatik II	2 V 1 Ü	<i>Gunzenhäuser</i> <i>Gunzenhäuser</i>
Analysis II	2 V 1 Ü	<i>Brenner</i> <i>Brenner</i>
Lineare Algebra	4 V 1 Ü	<i>Leichtweiß</i> <i>Leichtweiß</i>
Differentialgleichungen	2 V 1 Ü	<i>Werner, Drehman</i> <i>Werner, Drehman</i>
Einführung in die Informatik II	4 V 2 Ü	<i>Ludewig</i> <i>Ludewig</i>
Physikalische und elektrotechnische Grundlagen II	2 V 1 Ü	<i>Burkhardt</i> <i>Burkhardt, Walter</i>
Numerik	3 V 1 Ü	<i>Berger (DA)</i> <i>Berger (DA)</i>
Software–Praktikum I	4 P	<i>Eggenberger,</i> <i>Reuter u. Mitarb.</i>
Aufbau von Datenverarbeitungsanlagen	3 V 1 P	<i>Burkhardt</i> <i>Burkhardt,</i> <i>Homeister</i>
Hardware–Praktikum	4 P	<i>Burkhardt,</i> <i>Homeister, Rust,</i> <i>Walter</i>
Betriebssysteme I	3 V 1 Ü	<i>Eggenberger</i> <i>Eggenberger</i>
Komplexitätstheorie	2 V 1 Ü	<i>Thierauf (DA)</i> <i>Thierauf (DA)</i>
Echtzeitdatenverarbeitung	3 V	<i>Eggenberger</i>
Prädikatenlogik	3 V 1 Ü	<i>Knödel</i> <i>Knödel</i>

---

Mikroprogrammierung	3 V	<i>Ebert</i>
E/A–Organisation	2 V	<i>Hieber</i>
Künstliche Intelligenz I	2 V	<i>Lehmann</i>
	1 Ü	<i>Lehmann, Burkert, Tausend</i>
Dialogsysteme II	2 V	<i>Gunzenhäuser, Böcker</i>
Syntaxanalyse	2 V	<i>Göttler</i>
Text– und listenverarbeitende Verfahren	2 V	<i>Hanakata</i>
CAD,CAD/CAM–Automatisierung des technischen Informationsflusses II (EDV–Anwendung in Konstruktion und Entwicklung)	1 V	<i>Storr</i>
	1 Ü	<i>Storr</i>
Graphische Datenverarbeitung	2 V	<i>Grieger</i>
Künstliche Intelligenz III	2 V	<i>Lehmann, C.Rathke</i>
Informationssysteme II	2 V	<i>Reuter</i>
Deduktionsverfahren I	2 V	<i>Schönfeld</i>
Datenbank–Anwendungssysteme	2 V	<i>Reuter</i>
Entwurf kundenspezifischer Schaltungen	2 V	<i>Rust (DA)</i>
	1 P	<i>Rust (DA)</i>
Periphere Geräte	2 V	<i>Böhm (DA)</i>
Software–Ergonomie	2 V	<i>Herczeg (LA)</i>
Konzepte der Datenfernverarbeitung	2 V	<i>Lutz (LA)</i>
Nichtprozedurale Programmierung	3 V	<i>Schwinn (DA)</i>
	2 Ü	<i>Schwinn (DA)</i>
Maschinelles Lernen und Sprachverarbeitung	2 S	<i>Lehmann, Tausend, Wirth</i>
Gesellschaftliche Auswirkungen der Informatik	2 S	<i>Böcker (DA), Gunzenhäuser</i>
Logik–Programmierung und Parallelverarbeitung	2 S	<i>Schimpf (DA)</i>

---

---

Multi–Transputer–Systeme	2 S	<i>Bodenschatz (DA)</i>
Multiprozessoren–Arrays	2 S	<i>Homeister (DA)</i>
RISC/CISC–Multiprozessorsysteme	2 S	<i>Walter (DA)</i>
Modelle des Problemlösens und Planens	2 S	<i>Lehmann, C.Rathke, Burkert, Tausend</i>
Konnektionismus	2 S	<i>Rösner (LA)</i>
Methoden der Softwareentwicklung	2 S	<i>Ludewig, Lichter</i>
Neuere Entwicklungen des rechnerunterstützten Lernens	2 HS	<i>Gunzenhäuser</i>
SMALLTALK 80 — Programmiersprache und Programmierumgebung	2 HS	<i>Böcker (DA)</i>
Mustererkennung	2 HS	<i>Hanakata</i>
Mathematische Grundlagen der Künstlichen Intelligenz	2 HS	<i>Eusterbrock (DA)</i>
Komplexität Boolescher Funktionen	2 HS	<i>Köbler (DA)</i>
Struktur moderner Großrechnersysteme	2 HS	<i>Ebert</i>
Übersetzung und Optimierung in parallelen Systemen	2 HS	<i>Reuter u. Mitarb.</i>
Wortbedeutung und Weltwissen in KI und Linguistik	2 HS	<i>Lehmann, Rohrer</i>
Graph–Grammatiken		<i>Göttler</i>
Compilergeneratoren		<i>Göttler</i>
Programmiersprache SCHEME		<i>Göttler</i>
Datenschutz und Datensicherheit		<i>Biller (LA)</i>
Objektorientierte Programmierung	2 V	<i>Hanakata</i>
Informationstheorie	3 V	<i>Reuß (DA)</i>
	1 Ü	<i>Reuß (DA)</i>
C–Kompaktkurs		<i>Böhm (DA)</i>
PROLOG–Kompaktkurs		<i>Knopik</i>
LISP–Kompaktkurs		<i>C.Rathke</i>
Datenbank–Praktikum	4 P	<i>Reuter u. Mitarb.</i>

---

# Lehre im Wintersemester 1989/90

Grundlagen der Informatik I (für Studiengang technischer Diplomkaufmann)	2 V 1 Ü	Glatthaar (LA) Weber (DA)
Grundlagen der Informatik I (für Studiengang Physik, Luft- und Raumfahrt )	2 V 1 Ü	Gunzenhäuser Schweikhardt (DA)
Grundlagen der Informatik I (für Studiengang Energie-, Fertigungs- und Konstruktionstechnik, Verfahrenstechnik)	2 V 1 Ü	Baitinger Baitinger, Lanchès
Analysis I	5 V 2 Ü	Kolbe
Mathematische Grundlagen	4 V 2 Ü	Roggenkamp Roggenkamp
Einführung in die Informatik I	4 V 2 P	Ludewig Ludewig u. Mitarb.
Physikalische und elektrotechnische Grundlagen I	2 V 1 Ü	Burkhardt Burkhardt
Wahrscheinlichkeitstheorie und Warteschlangen	2 V 1 Ü	Knödel Knödel
Logik	4 V 1 Ü	Reuß (DA) Reuß
Kombinatorische und sequentielle Netzwerke	2 V 1 Ü	Eggenberger Eggenberger
Einführung in die Informatik III	4 V 2 Ü	Lagally Lagally
Automatentheorie und formale Sprachen	3 V 1 Ü	Diekert Diekert
Entwurf und Analyse von Algorithmen und Datenstrukturen	2 V 1 Ü	Zell (DA) Zell (DA)

---

Rechnerarchitektur	4 V 1 Ü	<i>Burkhardt, Rust</i> <i>Burkhardt, Rust</i>
Interaktive und intelligente Systeme	4 V 1 Ü	<i>Gunzenhäuser, Lehmann</i> <i>Knopik (DA), Burkert</i>
Informationssysteme, Datenbanken	4 V 1 Ü	<i>Schiele</i> <i>Gugel (DA)</i>
Programmiersprachen und Compilerbau	4 V 1 Ü	<i>Göttler</i> <i>Göttler</i>
Numerik (für Informatiker)	3 V 2 Ü	<i>Knödel, Münchow</i> <i>Knödel, Münchow</i>
Software–Praktikum II	4 P	<i>Eggenberger u.a.</i>
Entwurf großer Systeme	2 V	<i>Endres</i>
Codierung	3 V 1 Ü	<i>Reuß (DA)</i> <i>Reuß (DA)</i>
Formale Semantik	2 V 1 Ü	<i>Lagally</i> <i>Lagally</i>
Grundelemente des Software–Engineering	2 V	<i>Ludewig</i>
Implementierung von Datenbanksystemen	3 V 1 Ü	<i>Duppel (DA)</i> <i>Duppel</i>
Verteilte Systeme	2 V	<i>Reuter</i>
Implementierung von Programmiersprachen	2 V	<i>Lagally</i>
Leistungsmessung von Systemen	2 V 1 Ü	<i>Hieber</i> <i>Hieber</i>
Aufbau und Einsatz von Mikrocomputern	2 V 1 P	<i>Burkhardt</i> <i>Burkhardt</i>
CAM, CAP,CAD/NC–Automatisierung des technischen Informationsflusses I	1 V 1 Ü	<i>Storr</i> <i>Storr</i>
Graphische Datenverarbeitung I	1 V 1 Ü	<i>Grieger</i> <i>Grieger</i>

---

---

Künstliche Intelligenz II	2 V	<i>C.Rathke, Lehmann</i>
Mustererkennung und Bildverarbeitung	2 V	<i>Hanakata</i>
Symbolmanipulation	2 V	<i>Böcker (DA), C.Rathke</i>
Entwurf und Analyse von Algorithmen von Datenstrukturen	2 V	<i>Knödel</i>
	1 Ü	<i>Knödel</i>
Standarddatenformate zur Berechnung von digitalen Schaltkreisen	2 V	<i>Ryba</i>
Reduktionssysteme	2 V	<i>Diekert</i>
	2 Ü	<i>Diekert</i>
Scientific Computing	2 V	<i>Thunè</i>
Projektmanagement	2 V	<i>Biller (LA)</i>
Konzepte höherer Programmiersprachen	2 V	<i>Schwinn (DA)</i>
Methoden der Systemanalyse	2 V	<i>Seeland (LA)</i>
Systemprogrammierung II	2 V	<i>Eggenberger</i>
Interaktives Problemlösen	2 V	<i>Strothotte (DA)</i>
Rechnerverbund	2 V	<i>Ebert</i>
Analyse natürlicher Sprachen	2 S	<i>Burkert, Tausend (DA)</i>
Anwendungen von Graph–Grammatiken	2 S	<i>Göttler</i>
Auswirkungen der Informatik	2 S	<i>Böcker (DA), Gunzenhäuser</i>
Neue Aspekte von Multiprozessoren	2 S	<i>Walter (DA)</i>
Parallelverarbeitung mit Transputern	2 S	<i>Homeister (DA)</i>
UNIX–Konzepte	2 S	<i>Armbruster (DA)</i>
Anwendungen der Rechnerkommunikation	2 HS	<i>Ebert</i>
Anwendungsneutrale Benutzerschnittstellen	2 HS	<i>Gunzenhäuser, Schweikhardt</i>

---

Implementierung von Netzwerksprotokollen auf Kleinrechnern	2 HS	<i>Berger (DA)</i>
Lastbalancierung von Parallelrechnern	2 HS	<i>Reuter</i>
Wissensrepräsentation	2 HS	<i>Lehmann, C.Rathke</i>
Petri–Netze	2 HS	<i>Diekert</i>
C–Kompaktkurs		<i>Hanakata</i>
Common–LISP–Kompaktkurs		<i>Tausend</i>
Graphik–Normen Kompaktkurs		<i>Grieger</i>
PROLOG–Kompaktkurs		<i>Knopik</i>

### 2.3.3 Informatik-Kolloquium

11.4.	<b>Strothotte, T.</b> Institut für Informatik Universität Stuttgart	<i>Bildhafte Darstellungen in interaktiven wissensbasierten Systemen</i>
18.4.	<b>Kowalk, W.</b> Philips AG Nürnberg	<i>ATD Kommunikationstechnik der Zukunft</i>
26.4.	<b>Walter, B.</b> RWTH Aachen	<i>Koordination in verteilten Anwendungssystemen</i>
2.5.	<b>Krumm, H.</b> Universität Karlsruhe	<i>Spezifikation und Analyse von Systemen kommunizierender Instanzen</i>
9.5.	<b>Reischuk, R.</b> Technische Hochschule Darmstadt	<i>Uniformer Ansatz zur effizienten Lösung algorithmischer Probleme</i>
13.6.	<b>Rothermel, K.</b> IBM Deutschland GmbH	<i>Datenbankfernzugriff in Rechnernetzen</i>
20.6.	<b>Meyer-Wegener, K.</b> Universität Kaiserslautern	<i>Konzepte für verteilte DB/DC-Systeme und ihre Nutzung beim Einsatz</i>
21.6.	<b>Below, Y.</b> Universität Kiew	<i>Prototypenbildung und Abschätzung der Effektivität</i>
22.6.	<b>Mühlhäuser, M.</b> Universität Karlsruhe	<i>Software Engineering für verteilte Anwendungen: Sprachen, Werkzeuge</i>
27.6.	<b>Böttcher, S.</b> IBM Deutschland GmbH	<i>Regeln, Synchronisation und Integritätsbedingungen in Datenbanksystemen</i>
4.7.	<b>Morasca, S.</b> Politecnico Mailand/Italien	<i>ER Nets: a Petri Net Based Formalism for Time-dependent Systems</i>
11.7.	<b>Zell, A. und Sonntag, Chr.</b> Institut für Informatik Universität Stuttgart	<i>Hypercard-Informationsprogramm zur Struktur der Fakultät Informatik</i>

20.7.	<b>Zell, A.</b> Institut für Informatik Universität Stuttgart	<i>Computerviren des Apple Macintosh an der Fakultät Informatik</i>
15.8.	<b>Stelovsky, J.</b> Universität Hawaii	<i>Hypermedia Applications and Research</i>
18.8.	<b>Mohan, C.</b> IBM Almaden, Research Center, USA	<i>Efficient and High Concurrency Index Management Method</i>
21.8.	<b>Garcia-Molina, H.</b> Princeton Universität/USA	<i>Management of a Remote Backup Copy for Disaster Recovery</i>
5.9.	<b>Hollbach-Weber, S.</b> Universität Rochester/New York	<i>Structal Connectionist Approach to Direct Inferences</i>
12.9.	<b>Weber, C.</b> Universität Rochester/New York	<i>A Parallel Algorithm for Statistical Belief Refinement</i>
17.10.	<b>Ryba, M.</b> Institut für Parallele und Verteilte Höchstleistungsrechner	<i>Offene integrierte CAD–Systeme für den Entwurf integrierter Schaltungen</i>
24.10.	<b>Ambos-Spies, K.</b> Universität Heidelberg	<i>Minimale Paare für Polynomialzeit-reduzierbarkeit</i>
7.11.	<b>Reischuk, R.</b> Technische Hochschule Darmstadt	<i>Layout-effiziente Verfahren zum Design fehlertoleranter Schaltkreise</i>
14.11.	<b>Brandenburg, F.J.</b> Universität Passau	<i>Polynomiale Graph Grammatiken und Layouts von Graphen</i>
21.11.	<b>Weihrauch, K.</b> Fern Universität Hagen	<i>Berechenbarkeit aus separablen metrischen Räumen</i>
27.11.	<b>Lange, K.-J.</b> Technische Universität München	<i>Eindeutigkeit bei Schaltnetzen</i>

---

- 
- |        |  |   |
|--------|--|---|
| 28.11. | <b>Otto, F.</b><br>Universität Kassel                | <i>Kanonische Ersetzungssysteme — ein Paradigma zum Entwurf von Algorithmen</i> |
| 5.12.  | <b>Ottmann, Th.</b><br>Universität Freiburg          | <i>Varianten von Prioritätssuchbäumen</i>                                       |
| 12.12. | <b>Diekert, V.</b><br>Technische Universität München | <i>Freie partiell kommutative Monoide</i>                                       |
| 12.12. | <b>Maass, W.</b><br>Universität Illinois             | <i>Komplexität von lernenden Maschinen</i>                                      |

## 2.3.4 Habilitationen und Examensarbeiten

### 2.3.4.1 Habilitationen

**Strothotte, Thomas** *Interaktive und wissensbasierte Methoden zur integrierten Bild- und Sprachkommunikation*  
Hauptbericht : Gunzenhäuser  
Mitbericht : Fellbaum (Technische Universität Berlin)

### 2.3.4.2 Dissertationen

**Bodenschatz, Wolfgang** *Multi-Transputer-Maschine zur parallelen Reduktion von Funktionsprogrammen*  
Hauptbericht : Burkhardt  
Mitbericht : Ebert (AEG Ulm)

**Bräunl, Thomas** *Ein Modell der parallelen Programmierung mit funktionaler Spezifikation der Netzwerktopologie*  
Hauptbericht : Barth  
Mitbericht : Ludewig

**Dengel, Andreas** *Automatische visuelle Klassifikation von Dokumenten*  
Hauptbericht : Barth  
Mitbericht : Reuter (IPVR)

**Fehrle, Thomas** *Menüorientierte, wissensbasierte Klärungsdialoge für ein natürlichsprachliches Auskunftssystem*  
Hauptbericht : Gunzenhäuser  
Mitbericht : Lehmann

**Hampp, Albrecht** *Bedienergesteuerte Anpassung einer Benutzungsoberfläche unter Berücksichtigung verschiedener Dialogformen und Interaktionstechniken*  
Hauptbericht : Gunzenhäuser  
Mitbericht : Lauber (Universität Stuttgart)

**Köbler, Johannes** *Strukturelle Komplexität von Anzahlproblemen*  
Hauptbericht: Schöning (EWH Koblenz)  
Mitbericht: Knödel

---

---

<b>Schwab, Thomas</b>	<i>Methoden zur Dialog- und Benutzermodellierung in adaptiven Computersystemen</i> Hauptbericht : Gunzenhäuser Mitbericht : Ludewig
<b>Weber, Gerhard</b>	<i>Interaktive Dialogtechniken für blinde Rechnerbenutzer</i> Hauptbericht : Gunzenhäuser Mitbericht : Hoepelmann (Universität Stuttgart)
<b>Wirth, Rüdiger</b>	<i>Lernverfahren zur Vervollständigung von Hornklauselmen- gen durch inverse Resolution</i> Hauptbericht : Lehmann Mitbericht : Schönfeld
<b>Zell, Andreas</b>	<i>Metaebenen-Architekturen logischer Programme</i> Hauptbericht : Barth Mitbericht : Reuter (IPVR)

### 2.3.4.3 Diplomarbeiten

Ackermann, Ralf	<i>Entwurf und Realisierung einer kartographischen Datenbank</i> Prüfer : Reuter (IPVR) Betreuer : Kremers (Geodätisches Institut)
Auer, Günter	<i>Auftragsabwicklungssystem</i> Prüfer : Ebert (AEG-Ulm), Lagally
Barth, Boris	<i>Entwicklung einer Normteilebibliothek für Wälzlager auf einem zweidimensionalen CAD-System</i> Prüfer : Lagally Betreuer : Becker, (Hewlett Packard)
Bitter, Bernhard	<i>Konzeption und Implementierung eines zentralen Informationssystems für die kommerzielle DV-System-Entwicklung</i> Prüfer : Ebert (AEG-Ulm)
Börkel, Manfred	<i>Knowledge Engineering für ein wissensbasiertes natürlichsprachliches System</i> Prüfer : Studer (IBM), Lehmann Betreuer : v. Luck (IBM)
Bosch, Jürgen	<i>Ein System Design Tool für ASICs</i> Prüfer : Burkhardt Betreuer : Rust
Brösamle, Klaus	<i>Ein Übersetzer für PROLOG</i> Prüfer : Lagally Betreuer : Schimpf
Brunn, Nicola	<i>Automatische Schnittaufteilung bei Einstechbearbeitungen</i> Prüfer : Eggenberger, Gunzenhäuser Betreuer : Lichtinger (Daimler Benz)
Buhl, Lothar	<i>Modularisierung attributiver Grammatiken</i> Prüfer : Endres (IBM) Betreuer : Steinhoff (IBM)
Burkert, Bernd	<i>Entwicklung einer Laienprogrammiersprache für die NICE-Compakteinheit</i> Prüfer : Eggenberger, Gunzenhäuser Betreuer : Isbruch (Nixdorf)

---

---

<b>Czadek, Michael</b>	<i>Prototyp für eine MS-WINDOWS-Benzeroberfläche der objektorientierten Sprache CTALK unter UNIX</i> Prüfer : Gunzenhäuser Betreuer : Hanakata
<b>Czymmeck, Michael</b>	<i>Subsumption für eine erweiterete attributive Sortenbeschreibungssprache</i> Prüfer : Studer (IBM) Betreuer : Hettstück (IBM)
<b>Dammert Jürgen</b>	<i>Paralleles RAY-Tracing</i> Prüfer : Baitinger (IPVR) Betreuer : Homeister
<b>Deininger, Markus</b>	<i>Wissenserwerb durch Analogien im wissensbasierten Beratungssystem Keystone</i> Prüfer : Gunzenhäuser Betreuer : Knopik
<b>Dieterich, Hartmut</b>	<i>Benutzermodellierung durch Codeanalyse</i> Prüfer : Gunzenhäuser
<b>Dieter, Jochen</b>	<i>Leistungsuntersuchungen am relationalen DBMS Ingres</i> Prüfer : Reuter (IPVR) Betreuer : Haberhauer (IPVR)
<b>Dilly, Jutta</b>	<i>Ein graphikorientiertes System zum Erstellen und Lesen elektronischer Bücher</i> Prüfer : Gunzenhäuser
<b>Du, Van Tham</b>	<i>Lastbalanzierung in einem Parallelrechensystem</i> Prüfer : Reuter (IPVR)
<b>Ebinger, Werner</b>	<i>Orakelturingmaschinen und Zufallsorakel</i> Prüfer : Knödel Betreuer : Köbler
<b>Elsner, Sigrun</b>	<i>Entwurf und Implementierung graphischer Zoom-Techniken für Schnittstelle-Prototypen komplexer Systeme</i> Prüfer : Gunzenhäuser Betreuer : Strothotte, Rohr (IBM)

---

<b>Genikomsidis, Dinmitros</b>	<i>Entwicklung eines Software-Werkzeugs zur rechnergestützten Vorgabezeitermittlung an manuellen Montagearbeitsplätzen</i> Prüfer : Bullinger (IAO), Lagally Betreuer : Brunn (IAO), Menges (IAO)
<b>Gerstl, Peter</b>	<i>Ermittlung von Handlungshypothesen in einem Mehrragente-Szenario</i> Prüfer : Barth Betreuer : Zell
<b>Grote, Armin</b>	<i>Logische Programmierung durch kommunizierende veränderliche Objekte</i> Prüfer : Göttler Betreuer : Welsch
<b>Gugel, Dieter</b>	<i>Entwurf geeigneter Basismechanismen</i> Prüfer : Reuter (IPVR) Betreuer : Schiele (IPVR)
<b>Herczeg, Jürgen</b>	<i>Ein grafischer, visueller Trace</i> Prüfer : Gunzenhäuser Betreuer : Böcker
<b>Hönes, Frank</b>	<i>AMS — Ein Informationssystem für Anwendungs- und Anwenderdaten in Großrechenzentren</i> Prüfer : Eggenberger Betreuer : Krumrain (USU Software-Haus)
<b>Hoffmann, Marcus</b>	<i>Entwickl. eines interaktiven Analyseprogramms zur Erkennung von Interruptsperrern und Bestimmung von Interruptdatenzeiten</i> Prüfer : Eggenberger
<b>Hospach, Hans-Joachim</b>	<i>Analyse und Realisierung einer Anwenderschnittstelle zwischen CAD-System und Geometrie-Datenband</i> Prüfer : Lagally Betreuer : Seeland (Daimler Benz)
<b>Kapteina, Christoph</b>	<i>Entwurf und Implementierung eines Übersetzer-Frontends für eine einheitliche explizite Fertigungszellen-Programmiersprache</i> Prüfer : Storr Betreuer : Frager

---

<b>Kutschker, Siegmar</b>	<i>Entwickl. und Implementierung eines Verfahrens zur Minimierung unvollständig spezifizierter Mealy–Automaten</i> Prüfer : Eggenberger, Knödel Betreuer : Berger
<b>Lautz, Thomas</b>	<i>Konzept und Implementierung eines User–Profils für die individuelle Datenverarbeitung auf IBM–Großrechnern</i> Prüfer : Ebert (AEG–Ulm)
<b>Litty, Klaus</b>	<i>Ein graphikorientiertes System zum Erstellen und Lesen elektronischer Bücher (Teil 1)</i> Prüfer : Gunzenhäuser
<b>Maier, Joachim</b>	<i>Integration von Constraints in die Lilog Sortenbeschreibung</i> Prüfer : Studer (IPVR) Betreuer : v. Luck (IBM)
<b>Mailänder, Andreas</b>	<i>Ein TMS–basierter Problemlöser in der objektorientierten Programmiersprache C–Talk</i> Prüfer : Hanakata
<b>Mergl, Jürgen</b>	<i>ObjTalk Examiner ein Analysewerkzeug für ObjTalk–Programme</i> Prüfer : Gunzenhäuser Betreuer : v.d. Herberg
<b>Meyer, Gregor</b>	<i>Regelauswertung auf Datenbanken im Rahmen des PROTOS–L–Systems</i> Prüfer : Reuter (IPVR), Studer (IBM) Betreuer : Böttcher (IBM)
<b>Müller, Bernd</b>	<i>Ein Resolutionskalkül für ordnungssortierte Prädikatenlogik mit Sortenliteralen</i> Prüfer : Studer (IBM) Betreuer : Hedtstück (IBM)
<b>Patzner, Michael</b>	<i>Ein rechnerunterstütztes Lexikon für gehörlose Schüler</i> Prüfer : Gunzenhäuser Betreuer : Schweikhardt

---

<b>Paulusch, Oliver</b>	<i>Analyse und prototypische Implementierung tippfehlertolerierender Verfahren mit Berücksichtigung von Lemmatisierung</i> Prüfer : Gunzenhäuser Betreuer : Franz (Btx Datenbank Südwest)
<b>Pawlitschek, Michael</b>	<i>Ein Werkzeug zur Visualisierung von Smalltalk-80-Objekten</i> Prüfer : Gunzenhäuser Betreuer : Böcker
<b>Reichert, Günter</b>	<i>Entwurf und Implementierung eines WYSIWYG Editors zur Erstellung, Modifikation und Archivierung von Mixed Mode-Dokumenten</i> Prüfer : Barth Betreuer : Dengel
<b>Reichmann, Jürgen</b>	<i>Transputer-Peripherie-Subsystem</i> Prüfer : Lagally, Ebert (AEG-Ulm)
<b>Reinhardt, Klaus</b>	<i>Hierarchien mit alternierenden Kellerautomaten, alternierenden Grammatiken und finiten Transducern</i> Prüfer : Knödel Betreuer : Thierauf
<b>Rossmann, Thomas</b>	<i>Ein Editor zur interaktiven Erstellung von Benutzeroberflächen</i> Prüfer : Gunzenhäuser, Bullinger (IAO) Betreuer : Trefz (IAO)
<b>Reisinger-Voith, Maria</b>	<i>Entwurf und Implementierung eines interaktiven, einfach zu handhabenden Anwendungsgenerators für das relationale DBMS Ingres</i> Prüfer : Reuter (IPVR) Betreuer : Neugebauer (IPVR)
<b>Ryser, Sigrid</b>	<i>Ein Hypertextsystem mit teilautomatischer Strukturierung eines Begriffsnetzes</i> Prüfer : Gunzenhäuser Betreuer : Knopik
<b>Schaber, Wolfgang</b>	<i>Konzepte zur Modellierung von Büroprozeduren</i> Prüfer : Barth Betreuer : Niemeier (IAO)

---

<b>Schmid, Claudia</b>	<i>Grenzen sprachlicher Darstellungen in der Informationsverarbeitung</i> Prüfer : Gunzenhäuser Betreuer : Strothotte
<b>Schmid, Karl-Heinz</b>	<i>Spezifikation eines sprachunabhängigen generellen Struktur-editors</i> Prüfer : Lagally, Warnecke (IPA)
<b>Schmidbauer, Jörg</b>	<i>Eine Komponente zum Führen eines Beratungsdialoges mit einem natürlichsprachlichen Auskunftssystem auf Basis von Planverfahren</i> Prüfer : Gunzenhäuser Betreuer : Fehrle
<b>Schnekenburger, Thomas</b>	<i>Lineare Programme und Ellipsoid–Methoden</i> Prüfer : Knödel
<b>Schröter, Bernhard</b>	<i>Vergleich einer sequentiellen und einer asynchronen Realisierung des Debit–Credit–Benchmarks</i> Prüfer : Reuter (IPVR) Betreuer : Schiele (IPVR)
<b>Schroth, Albrecht</b>	<i>Konzipierung und Systementwurf einer Koppelnetzsteuerung für ein Multiprozessorsystem programmierbarer Logik</i> Prüfer : Ebert (AEG–Ulm)
<b>Schütze, Hinrich</b>	<i>Pluralbehandlung in natürlichsprachlichen Wissensverarbeitungssystemen</i> Prüfer : Rohrer, Studer (IBM) Betreuer : Link (IBM)
<b>Schwamberger, Hans</b>	<i>Unterstützung von NETVIEW in heterogenen SNA–Netzen</i> Prüfer : Ebert (AEG–Ulm)
<b>Schwarz, Edgar</b>	<i>Entw. u. Implementierung eines natürlichsprachlichen Demonstrationssystems für die Theorie der Kontraste zur Darstellung von Alltagswissen</i> Prüfer : Lehmann, Bullinger (IAO), Hoepelmann (Linguistik)

---

<b>Schwarzmaier, Doris</b>	<i>Rechnergestütztes Konfigurations-Management-System für heterogene DV-Netze</i> Prüfer : Ebert (AEG-Ulm), Lagally
<b>Schweizer, Eberhard</b>	<i>Erfassung, Justierung und Segmentierung von Dokumentstrukturen</i> Prüfer : Barth Betreuer : Dengel
<b>Semle, Heiner</b>	<i>Erweiterung einer abstrakten Maschine für ordnungssortiertes PROLOG um die Behandlung von polymorphen Sorten</i> Prüfer : Studer (IBM), Ludewig
<b>Siegle, Markus</b>	<i>Implementierung eines experimentellen Expertensystems für den Chip-Design</i> Prüfer : Burkhardt Betreuer : Rust
<b>Speer, Thomas</b>	<i>Interaktive Verarbeitung von Raster- und Vektorbildern</i> Prüfer : Gunzenhäuser Betreuer : Grieger
<b>Thaler, Helmuth</b>	<i>CADCAM — Betriebsüberwachungsprogramm</i> Prüfer : Lagally Betreuer : Seeland (Daimler Benz)
<b>Tischendorf, Manfred</b>	<i>Untersuchung verschiedener Strategien zur wissensbasierten Auswahl von Daten</i> Prüfer : Reuter (IPVR) Betreuer : Schmidt (Institut für Kernenergetik), Neugebauer (IPVR)
<b>Traub, Stefan</b>	<i>Datensicherungskonzepte in verteilten Rechensystemen</i> Prüfer : Ebert (AEG-Ulm)
<b>Wächter, Helmut</b>	<i>Transaktionsverwaltung und Recovery in Objektbanken</i> Prüfer : Reuter (IPVR)
<b>Weigel, Gerhard</b>	<i>Definition und Entwicklung einer Debug-Umgebung</i> Prüfer : Endres (IBM) Betreuer : Unckell (IBM)

---

---

<b>Wied, Klaus</b>	<i>Ein Laufzeitsystem und eine Beschreibungssprache für ein ereignisgesteuertes User Interface Management–System</i> Prüfer : Bullinger (IAO), Gunzenhäuser Trefz:
<b>Winkler, Klaus</b>	<i>Simulation einer Warren Abstract Maschine</i> Prüfer : Burkhardt Betreuer : Walter
<b>Wolfer, Klaus</b>	<i>Entwurf und Implementierung eines modularen Fenstersystems für alphanumerische Terminals</i> Prüfer : Bullinger (IAO), Gunzenhäuser Betreuer : Raether (IAO)
<b>Zendler, Hans-Peter</b>	<i>Implementierung einer Nachrichtenschnittstelle SUN–TANDEM über TCP/IP</i> Prüfer : Reuter (IPVR) Betreuer : Duppel (IPVR)
<b>Zeyer, Martin</b>	<i>Speichereffiziente Implementierung von “Packed Trees” für PCs</i> Prüfer : Eggenberger, Lagally Betreuer : Ziegler
<b>Zink, Roland</b>	<i>Implementierung einer prozeßorientierten parallelen deduktiven Datenbank (Regelcompilation)</i> Prüfer : Reuter (IPVR) Betreuer : Duppel (IPVR), Schiele (IPVR)
<b>Zipperer, Hans-Georg</b>	<i>Aufbau und Betrieb eines Verbindungsnetzwerkes zur dynamischen Erzeugung von Transputernetz–Topologien</i> Prüfer : Burkhardt Betreuer : Walter

---

Ist bei den Diplomarbeiten nur ein Prüfer benannt, so hat dieser auch die jeweiligen Arbeiten betreut

---

### 2.3.4.4 Studienarbeiten

**Ackermann, Joachim** *Optimierung paralleler Abarbeitungspläne für baumartige Ableitungsstrukturen anhand einiger Beispiele*

Prüfer : Reuter (IPVR)

Betreuer : Duppel (IPVR)

**Albicker, Klemens** *Ein Arbeitsplaneditor zur interaktiven Eingabe und Modifikation von Arbeitsplandaten auf objektorientierter Basis*

Prüfer : Gunzenhäuser

Betreuer : Kroneberg (IFF)

**Amman, Carina** *Spezifikation und Verifikation von Protokollen in verteilten Systemen*

Prüfer : Reuter (IPVR)

Betreuer : Wächter (IPVR)

**Athanassion, Akis** *Interaktion mit bildhaften Darstellungen bei einem wissensbasierten Lernsystem*

Prüfer : Gunzenhäuser

Betreuer : Strothotte

**Baum, Winfried** *Ein Transputer-Modul für den Einsatz in PCs*

Prüfer : Burkhardt

Betreuer : Bodenschatz

**Bayer, Rolf** *Entwicklung und Realisierung eines Simulatorkerns für konnektionistische Modelle*

Prüfer : Barth

Betreuer : Zell

**Beckmann, Frank** *Ein Labor zur Entwicklung und Evaluation von Planungsstrategien für ein Mehrpersonenspiel*

Prüfer : Gunzenhäuser

Betreuer : Böcker, Maier

**Beil, Thomas** *Objekterkennung im 3-D-Bereich*

Prüfer : Bullinger (IAO), Lagally

Betreuer : Nickel (IAO)

**Bell, Siegfried** *Evaluierung eines Programmes zum Erlernen von DCGs*

Prüfer : Lehmann

Betreuer : Wirth

---

---

<b>Blank, Karlheinz</b>	<i>Werkzeuge zur Gestaltung graphikorientierter Benutzereingaben</i> Prüfer : Gunzenhäuser Betreuer: Fehrle, Anglett (IBM)
<b>Böckmann, Phillip</b>	<i>Lernprogramm für Retrievalssprachen im Bereich des Bibliothekswesens mit Simulation der Wirkung einiger Kommandos und die Einführung in Suchstrategien</i> Prüfer : Gunzenhäuser Betreuer : Sailer (Fachhochschule für Bibliothekswesen, Stuttgart)
<b>Bohner, Markus</b>	<i>Entwicklung eines Browsers für die objektorientierte Sprache CTALK</i> Prüfer : Hanakata
<b>Both, Ralf</b>	<i>Ein Editor für eine hierarchisch organisierte Wissensbasis</i> Prüfer : Gunzenhäuser Betreuer: Knopik
<b>Briel, Edgar</b>	<i>Systementwurf und Systemkonfiguration eines EDV-Systems für eine Hochschulbibliothek</i> Prüfer : Ebert (AEG-Ulm)
<b>Brösamle, Klaus</b>	<i>KNOWTS — Eine objektorientierte Erweiterung von PROLOG</i> Prüfer : Barth Betreuer : Welsch
<b>Butz, Jochen</b>	<i>Untersuchung und Definition von Struktureigenschaften in Bürodokumenten sowie deren Bewertung mit Hilfe von kontinuierlichen Bewertungsmaßen</i> Prüfer : Barth Betreuer : Dengel
<b>Dolderer, Sven</b>	<i>Entwurf und Implementierung einer graphischen Ausgabe-Schnittstelle für das relationale DBMS Ingres</i> Prüfer : Reuter (IPVR) Betreuer : Neugebauer (IPVR)
<b>Dotzek, Karl</b>	<i>Spezifikation einer formalen Satzbeschreibung zum Generieren syntaktischer Strukturen im Rahmen der CUG</i> Prüfer : Studer (IBM) Betreuer : Novak (IBM)

---

<b>Eberhardt, Dieter</b>	<i>Aufbau und Funktionsweise von Computerviren auf dem IBM PC</i> Prüfer : Ebert (AEG-Ulm)
<b>Ebinger, Werner</b>	<i>Eine Implementierung des Resolutionsverfahrens für die Prädikatenlogik</i> Prüfer : Knödel Betreuer : Köbler
<b>Fach, Peter</b>	<i>Ein rechnergesteuertes System für eine standortunabhängige Erfassung und Auswertung analoger Meßwerte</i> Prüfer : Böhm Betreuer : Barzewsky (Institut für Wasserbau)
<b>Förderreuther, Johannes</b>	<i>Implementierung eines Compilers für die Hardware-Beschreibungssprache DACAPO unter Verwendung von LEX und YACC</i> Prüfer : Burkhardt Betreuer : Rust
<b>Gairing, Dieter</b>	<i>Entwurf und Implementierung eines Zeichenprogramms für Blinde</i> Prüfer : Gunzenhäuser Betreuer : Weber
<b>Gehring, Steffen</b>	<i>Futuristische Darstellung räumlicher Objekte mit Ray-Tracing Verfahren</i> Prüfer : Gunzenhäuser Betreuer : Grieger (Institut für Luft- und Raumfahrt)
<b>Geltz, Markus</b>	<i>Grafische Darstellung und Manipulation von Fischer-Technik Modellen</i> Prüfer : Gunzenhäuser Betreuer : Böcker
<b>Geschke, Thomas</b>	<i>Implementierung eines LSV/2-Protokolls zur Kopplung einer VAX-Station an eine Philipssteuerung</i> Prüfer : Burkhardt Betreuer : Maier (Fa. Unverzagt)

---

- Greitmann, Christine** *Entwurf und Realisierung einer graphischen Benutzeroberfläche I für einen Netzwerksimulator konnektionistischer Modelle*  
Prüfer : Barth  
Betreuer : Zell
- Günther, Volker** *Views als Benutzerschnittstelle für Datenbanken in einer Workstation-Umgebung*  
Prüfer : Reuter (IPVR)  
Betreuer : Klein (IPVR)
- Gutjahr, Bernd** *Entwurf und Implementierung der beiden Komponenten Wissensbasis und Inferenzmechanismus für ein Diagnose-Expertensystem*  
Prüfer : Lehmann, Bullinger (IAO)  
Betreuer : Kurz (IAO)
- Hanske, Wilhelm** *Entwurf und Implementierung von Anwenderdatenbanken auf dem relationalen DBS Ingres*  
Prüfer : Reuter (IPVR)  
Betreuer : Neugebauer (IPVR)
- Harm, Oliver** *Automatische Terminalbedienung*  
Prüfer : Lagally  
Betreuer : Wilhelm (IBM)
- Hermann, Michael** *Entwurf und Implementierung eines komfortablen Massenladers für das relationale DBMS Ingres*  
Prüfer : Reuter (IPVR)  
Betreuer : Neugebauer (IPVR)
- Hersmann, Wolfgang** *Testfunktionen zur Überprüfung von Konsistenz und Vollständigkeit der Wissensbasis eines Diagnosesystems*  
Prüfer : Lehmann  
Betreuer : Kurz (IFF)
- Hölzlein, Markus** *Die Verteilung der Abstände zufälliger Punkte im Quadrat*  
Prüfer : Knödel
- Hönes, Frank** *Möglichkeiten der visuellen Erkennung von Worten mit Hilfe von geometrischen Eigenschaften der enthaltenen Zusammenhangskomponenten*  
Prüfer : Barth  
Betreuer : Dengel

<b>Hoffmann, Marcus</b>	<i>Entwurf und Implementierung einer Betriebssystemumgebung für einen PC vom Typ AT ohne Externspeicher</i> Prüfer : Eggenberger
<b>Hohl, Hubertus</b>	<i>Adaptive Benutzerschnittstelle zur Präsentation von Lerninhalten</i> Prüfer : Gunzenhäuser Betreuer : Böcker, Schwab
<b>Huppenbauer, Helmut</b>	<i>Ein Expertensystem zur Schulungsberatung für IBM-Mitarbeiter</i> Prüfer : Barth Betreuer : Lenz (IBM)
<b>Huynh, van Loc</b>	<i>Implementierung des SOLO-Kerns</i> Prüfer : Lagally
<b>Jacobsen, Oliver</b>	<i>Entwicklung einer Benutzerschnittstelle für ein PC-gestütztes Optimierungsverfahren ganzzahliger Zuschnittsprobleme</i> Prüfer : Gunzenhäuser
<b>Janosch, Manfred</b>	<i>Umsetzung von in der Programmiersprache C beschriebenen Algorithmen in die Systembeschreibungssprache SIL</i> Prüfer : Burkhardt Betreuer : Rust
<b>Kaiser, Andreas</b>	<i>Assembler und Linker für Transputer</i> Prüfer : Böhm Betreuer : Homeister
<b>Kaja, Manfred</b>	<i>Entwicklung und Implementierung einer Anwenderdatenbank (Geohydrologie) auf dem relationalen DBS Ingres</i> Prüfer : Reuter (IPVR) Betreuer : Neugebauer (IPVR)
<b>Koczy, Oliver</b>	<i>Entwicklung eines Programmes zur Dienstplanerstellung</i> Prüfer : Böhm
<b>Korb, Thomas</b>	<i>Entwicklung und Implementierung einer Programmiersprache zur Definition konnektionistischer Netzwerke</i> Prüfer : Barth Betreuer : Zell

---

<b>Kortüm, Gerd</b>	<i>Systemgesteuerter Beratungsdialog als Zugangssystem zu wissensbasierten Informationssystemen</i> Prüfer : Gunzenhäuser Betreuer: Fehrle
<b>Kouroufexis, Andreas</b>	<i>Entwurf und Implementierung eines Visualisierungsverfahrens für einen im Octree-Modell dargestellten 3 D-Körper</i> Prüfer : Storr (Institut für Luft- und Raumfahrt) Betreuer : Joannides (Institut für Steuerungssysteme)
<b>Krauss, Rainer</b>	<i>Implementierung einer parallelen deduktiven Datenbank (Aktionen zur Anfrageevaluierung)</i> Prüfer : Reuter (IPVR) Betreuer : Duppel (IPVR), Schiele (IPVR)
<b>Kübler, Dietmar</b>	<i>Entwurf und Implementierung einer Anwenderdatenbank (Geohydrologie) auf dem relationalen DBS Ingres</i> Prüfer : Reuter (IPVR) Betreuer : Neugebauer (IPVR)
<b>Kutschker, Siegmar</b>	<i>Ein Verfahren zur Bestimmung der Primkompatiblen bei unvollständig spezifizierten Mealy-Automaten</i> Prüfer : Eggenberger Betreuer : Berger
<b>Liedtke, Thomas</b>	<i>Parser für Ausdrücke der Prädikatenlogik Programmiersprache PASCAL</i> Prüfer : Knödel
<b>Lutz, Rainer</b>	<i>Computergestützte Bewegungssimulation und Visualisierung graphischer Objekte im 3-dimensionalen Raum</i> Prüfer : Böhm (IPVR), Bullinger (IAO)
<b>Ly Hyung Dung</b>	<i>Implementierung Solointerpreter</i> Prüfer : Lagally
<b>Magino, Frank</b>	<i>Design und Erstellung einer Testversion einer Stückliste mit relationalem DBS</i> Prüfer : Ebert (AEG-Ulm) Betreuer : Ochs (Daimler Benz)

---

<b>Melchinger, Ernst</b>	<i>Eine Komponente zur Aufnahme von Dokumentstrukturwissen in ein Dokument-Architektur-Modell</i> Prüfer : Barth Betreuer : Dengel
<b>Merling, Dieter</b>	<i>Mode-Analyse logischer Programme</i> Prüfer : Barth Betreuer : Zell
<b>Müller, Klaus</b>	<i>Implementieren einer parallelen deduktiven Datenbank (Benutzerschnittstelle und Selektionsoptimierung)</i> Prüfer : Reuter (IPVR) Betreuer : Duppel (IPVR), Schiele (IPVR)
<b>Ocker, Matthias</b>	<i>Entwicklung eines interaktiven graphischen Editors für die Planung und Berechnung von Energienetzen</i> Prüfer : Böhm Betreuer : Schmieg (Ing. Beratungsfirma)
<b>Pauler, Gerold</b>	<i>Entwurf eines generischen Zeicheneinheitentreibers für MS-DOS als einfache Wälschnittstelle in der höheren Programmiersprache C</i> Prüfer : Ebert (AEG-Ulm)
<b>Pfisterer, Jörg</b>	<i>Transfer und benutzergerechte Manipulation von Roboterkommandos auf Basis einer graphisch-orientierten Rechneroberfläche</i> Prüfer : Ebert (AEG-Ulm) Betreuer : Rettich (IAO)
<b>Quecke, Hans</b>	<i>Anwendungen dynamischer Objekte in Dialoggeneratoren</i> Prüfer : Gunzenhäuser, Warnecke Betreuer : Raether (IAO)
<b>Rath, Ursula</b>	<i>Aufbau eines Expertensystems zur Einsatzplanung von Industrierobotern</i> Prüfer : Lehmann Betreuer : Jordan (Institut für Produktionstechnik und Automation)
<b>Reichert, Lars</b>	<i>Praktischer Vergleich verschiedener Unifikationsalgorithmen</i> Prüfer : Barth Betreuer : Zell

---

---

<b>Röhl, Roland</b>	<i>Implementierung der Hypothesenerzeugung eines wortorientierten Verfahrens zur Texterkennung</i> Prüfer : Barth Betreuer : Dengel
<b>Schadt, Christof</b>	<i>Realisierung eines Datenaustausches zwischen den Teilprojekten des PWAB-Verbundprojekts „Naturmeßfeld Horkheimer Insel“</i> Prüfer : Reuter (IPVR) Betreuer : Neugebauer (IPVR)
<b>Schäfer, Gert</b>	<i>Konzentrationsprogramm für Anwendungen im Bankenbereich</i> Prüfer : Ebert (AEG–Ulm)
<b>Schick, Karl-Wilhelm</b>	<i>Replikation in lokalen Netzwerken</i> Prüfer : Reuter (IPVR) Betreuer : Klein (IPVR)
<b>Schloß, Bernhard</b>	<i>Layout-Editor/Floorplanner für SYSEDIT</i> Prüfer : Burkhardt Betreuer : Rust
<b>Schneider, Mark-Tell</b>	<i>Entwurf, Aufbau und Erprobung eines Einplatinen-Rechners mit dem Mikroprozessor 8088 für portablen Einsatz</i> Prüfer : Eggenberger
<b>Schöbel, Thomas</b>	<i>Ein Modulkonzept für X PASCAL</i> Prüfer : Lagally
<b>Schwille, Jürgen</b>	<i>RAPID M-Erkennung von Parallelität in PROLOG-Programmen mittels Datenfluß und Modeanalyse</i> Prüfer : Barth Betreuer : Schwinn
<b>Siegle, Markus</b>	<i>CAD-Tool zur Plazierung von Zellen auf einer integrierten Schaltung unter Berücksichtigung vorgegebener Grundstrukturen</i> Prüfer : Burkhardt Betreuer : Rust

---

<b>Siehr, Hans-Jürgen</b>	<i>Erstellung eines Interpreters für die Übertragung graphischer Bildschirmschirmdarstellungen auf Laserdrucker am Beispiel von Postscript</i> Prüfer : Gunzenhäuser, Heimerl (IEV)
<b>Soberieray, Arndt</b>	<i>Überlegungen zu einem Diagnose- und Überwachungssystem für 3D-Koordinatenmeßgeräte als Komponenten flexibler Fertigungssysteme</i> Prüfer : Warnecke (IPA), Burkhardt Betreuer : Steger (IPA)
<b>Sommer, Tillman</b>	<i>Entwurf und Realisierung einer graphischen Benutzeroberfläche II</i> Prüfer : Barth Betreuer : Zell
<b>Sonntag, Christoph</b>	<i>Ein Hypercard-Informationsprogramm zur Struktur der Fakultät Informatik</i> Prüfer : Barth Betreuer : Zell
<b>Stirm, Dieter</b>	<i>Entwurf und Implementierung der Benutzeroberfläche für das Laufzeitsystem eines Diagnosesystems</i> Prüfer : Lehmann Betreuer : Kurz
<b>Thies, Markus</b>	<i>Gestaltung eines wissensbasierten Erklärungsdialogs</i> Prüfer : Gunzenhäuser Betreuer : Fehrle, Anglett (IBM)
<b>Tischer, Manfred</b>	<i>Entwurf und Implementierung einer graphischen Ausgabe-Schnittstelle für das relationale DBMS Ingres</i> Prüfer : Reuter (IPVR) Betreuer : Neugebauer (IPVR)
<b>Vojkovic, Vojko</b>	<i>Speicherungsstrukturen im Nonstop SQL</i> Prüfer : Reuter (IPVR) Betreuer : Duppel (IPVR)
<b>Wahl, Susanne</b>	<i>Entwurf und Implementierung eines interaktiven Programms zur rechnerunterstützten Vorgangsbearbeitung in Kindergarten</i> Prüfer : Gunzenhäuser, Eggenberger

---

---

<b>Weiß, Jürgen</b>	<i>Implementierung einer prozeßorientierten, parallelen und deduktiven Datenbank</i> Prüfer : Reuter (IPVR) Betreuer : Duppel (IPVR), Schiele (IPVR)
<b>Winkler, Klaus</b>	<i>Theoretischer Ansatz zur Erweiterung von PROLOG um objektorientierte Elemente</i> Prüfer : Barth Betreuer : Welsch
<b>Winnewisser, Roger</b>	<i>Kopplung-Koordinaten-Meßmaschine mit AIX-Workstation und CADCAM-System</i> Prüfer : Lagally Betreuer : Seeland (Daimler Benz)
<b>Woern, Ralpf</b>	<i>Entwicklung und Implementierung einer 3DM Datenbank von APOLLO für den Schaltungsentwurf</i> Prüfer : Burkhardt Betreuer : Rust
<b>Zeller, Martin</b>	<i>Manipulation geometrischer Objekte in PROLOG</i> Prüfer : Reuter (IPVR) Betreuer : Kremers (Geodätisches Institut), Zeller (IPVR)

---

Ist bei den Studienarbeiten nur ein Prüfer benannt, so hat dieser auch die jeweiligen Arbeiten betreut

# 3 Forschung

## 3.1 Forschungsvorhaben und Forschungsgruppen

### 3.1.1 Abteilung Betriebsssoftware

Leiter	<i>Lagally</i>
Professor	<i>Eggenberger</i>
Wiss. Mitarbeiter	<i>Armbruster, Kümmel, Schimpf (ab 1.2.), Ziegler</i>
Programmierer	<i>Merkel, Schlebbe</i>
Verwaltungsangestellte	<i>Groß (ab 17.5.)</i>

### Echtzeitdatenverarbeitung auf dem Personal Computer

( Otto Eggenberger, Uwe Berger )

Das Programmpaket 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. Neu ist die interaktive Bedienung bei der Installation des Systems und im laufenden Betrieb. Dazu gehören das Laden von Gerätetreibern und Prozessen sowie die momentane und kontinuierliche Anzeige von Prozeßzuständen. Auf Grund praktischer Erfahrungen wurde die Verträglichkeit mit existierenden Programmen verbessert.

Neben der Weiterentwicklung unseres eigenen Systems wurden acht Multitasking-Betriebssysteme hinsichtlich ihrer Eignung für Echtzeitanwendungen auf einem Personal Computer untersucht. Wesentliche Kriterien waren Prozeßverwaltung, Filesystem, Benutzeroberfläche, Entwicklungsumgebung und Kompatibilität zu Standardsoftware. Es zeigte sich, daß die Systeme sehr unterschiedlich im Leistungsumfang und in der Art der Programmierung sind. Eine Standardisierung im Bereich der Echtzeitdatenverarbeitung auf Personal Computern ist derzeit noch nicht absehbar.

---

## **Projekt: Pascal–Spracherweiterung für zahlentheoretische Anwendungen**

( *Klaus Lagally und Studenten* )

Das im Vorjahr bereits beschriebene XPASCAL–System wurde fertiggestellt und in mehrfacher Hinsicht weiterentwickelt:

- Die auf verschiedenen Kleinrechnern entwickelten Teile wurden auf ein UNIX–System (SUN3 unter SUN–OS) übertragen und dort integriert. Trotz der dadurch nötigen Umstellung auf andere Hochsprachen–Implementierungen (von TDI–Modula auf MOCKA, von ST–Pascal auf SUN–Pascal) sind dabei kaum Probleme aufgetreten; im wesentlichen waren nur einige Bibliotheks–Schnittstellen anzupassen.
- Das erstellte System ist ablauffähig und hat auch vernünftige Rechenzeiten, aber einen überraschend hohen Speicherbedarf. Daher wurde der für die Langzahlen zuständige Teil des Interpreters neu implementiert (Studienarbeit Robitschko). Die Integration ist noch nicht abgeschlossen, doch zeigen erste Tests bereits eine Reduktion des Speicherbedarfs fast um den Faktor 2.
- Die Sprache XPASCAL wurde um ein Modulkonzept erweitert, um Bibliotheks–routinen vorübersetzen und dazubinden zu können. (Studienarbeit Schöbel). Die gewählte Lösung ist voll aufwärtskompatibel und gestattet die textuelle Trennung von Schnittstellen–Spezifikation und Implementierung sowie Schnittstellenüberprüfung und Versionskontrolle zur Ladezeit. Die Implementierung ließ sich in das vorhandene Compiler– und Interpreter–System im wesentlichen problemlos nachrüsten, ohne die Schnittstelle zur Langzahl–Arithmetik zu ändern.

## **Projekt: WRG — ein Parsergenerator mit automatischer Fehlerbehandlung**

( *Klaus Lagally* )

An ein Compiler–Werkzeug, das im Lehr– und Übungsbetrieb eingesetzt werden soll, werden andere Anforderungen gestellt als an ein Werkzeug für Produktionszwecke:

- Das verwendete Parsing–Verfahren muß für Anfänger leicht verständlich sein, die Notation sollte nahe an der üblichen Schreibweise für kontextfreie Grammatiken liegen.
- Semantische Aktionen müssen in natürlicher Weise angeschlossen werden können und der gewohnten Denkweise und Programmietechnik der Benutzer entsprechen. Überraschungen sind unerwünscht.

- Der erzeugte Parser–Code muß leicht lesbar und bequem weiterzuverarbeiten sein.
- Die sinnvolle Behandlung syntaktischer Fehler ist wichtig, aber für den Anfänger zu schwierig; sie muß automatisiert werden.
- Das System sollte auf einem PC ablauffähig und mit einer der verbreiteten Hochsprachen–Implementierungen verträglich sein.
- Die automatische Erzeugung eines Scanners lohnt sich meist nicht; in der Regel ist ein leicht modifizierbarer Scanner für eine lexikalisch ähnliche Sprache vorhanden.
- Laufzeit– und Speichereffizienz sind von nachgeordneter Bedeutung.

Da die uns zugänglichen Systeme YACC und COCO diese Anforderungen nur man gelhaft erfüllen, haben wir ein neues Werkzeug entwickelt:

WRG (Werkzeug zur Rechnergestützten Generierung) erzeugt aus einer EBNF–Beschreibung einer kontextfreien LL(1)–Grammatik einen Parser nach der Methode des rekursiven Abstiegs, der wahlweise eine automatische Behandlung von Syntaxfehlern enthält. Die Zielsprache ist Turbo–Pascal. Die Grammatik kann eine L–Attributierung tragen, die auf das Parameterkonzept von Pascal abgebildet wird; semantische Aktionen sind Programmstücke in Pascal, die sich auch auf lokale und globale Hilfsvariablen beziehen können. Eine explizite Kellerung von Attributen und lokalen Größen ist durch die Abbildung auf das Prozedurkonzept von Pascal unnötig.

Das System wurde, ausgehend von einem von Hand implementierten Kern, durch Bootstrap auf den vollen Leistungsumfang erweitert. Es ist fertiggestellt und wird im Übungsbetrieb eingesetzt; derzeit sind keine Fehler bekannt. Nach den ersten vorliegenden Erfahrungen können Grammatiken mindestens im Umfang von Pascal verarbeitet werden. Die wahlweise erzeugte automatische Behandlung von Syntaxfehlern ist aufwendig und derzeit nicht optimal implementiert. Eine ausführliche Benutzungsbeschreibung ist in Arbeit, ebenso Effizienzverbesserungen zum Einsatz für Produktionszwecke.

## **Projekt: Lastverteilung und Prozessor–Kommunikation bei der parallelen Ausführung von Logikprogrammen**

( Stefan Schimpf )

Die Arbeit knüpft an das Forschungsprojekt „Parallelität und intelligentes Backtracking in logischen Programmen“ der Abteilung Programmiersprachen (siehe Festchrift zur Gründung der Fakultät Informatik 1988) an. Für das in diesem Projekt entstandene Datenflußmodel zur Realisierung von UND–Parallelität in Logikprogrammen wurde ein Konzept zur Implementierung auf einem Nonshared–Memory–Multiprocessor entwickelt. Geplant ist die Integration von ODER–Parallelität und

---

die Realisierung des Konzepts auf Transputern. Im weiteren Verlauf sollen Messungen zur Lastverteilung und Performance unter Verwendung verschiedener Scheduling–Strategien durchgeführt werden. Für Geschwindigkeitsvergleiche mit dem traditionellen (nicht–parallelen) Konzept der „Warren Abstract Machine“ wurde diese in Zusammenarbeit mit der Abteilung Computersysteme ebenfalls auf einem Transputer implementiert.

## **Projekt: Implementierung von Packed Tries**

( *Bernhard Ziegler, Martin Zeyer (Student)* )

Tries (von re TRIE val) sind m–äre Suchbäume, die sich besonders gut zum Speichern von großen Wortmengen eignen. Die Suche nach einem Wort im Trie hängt linear von der Wortlänge des gesuchten Wortes, nicht aber von der Größe des in ihm gespeicherten Wortschatzes ab.

In jedem Knoten des Tries ist für jedes Zeichen des unterliegenden Alphabets eine Komponente vorgesehen. Die Zahl der belegten Komponenten sinkt mit wachsendem Abstand von der Wurzel auf eins ab. Deshalb spart man bei Packed Tries Speicherplatz, indem man unterschiedliche Knoten einander überlagert. Dabei überdecken belegte Komponenten des einen Knoten leere Komponenten des anderen. Allerdings dürfen verschiedene Knoten nicht an der selben Speicherstelle beginnen, da sonst nicht mehr entscheidbar ist, welche Komponente zu welchem Knoten gehört. Mit dieser Methode ist es uns bisher gelungen, den Speicherbedarf bei etwa 140 000 Wortformen auf etwa 80 % des linearen Aufschreibs zu reduzieren. Wir hoffen ihn noch weiter auf etwa 60 % reduzieren zu können.

## **Forschungsvorhaben: Antwortgenerator**

( *Peter Kümmel* )

Durch die DFG anfänglich mit Sachmitteln gefördert, sind die Arbeiten zum Richtthema: „Speicherstrukturen für Antwortsucher“ 1989 weitergeführt worden. Ihre Ergebnisse dienen als Werkzeuge für die selbstdiagnosende Spracherkennung und gliedern sich:

1. in Erhebungen über die „Post–Shannon’sche, nichtklassische Informationstheorie“, basierend auf den sechs Deuter–Kriterien einer Bedeutung und
2. Neukonzeptionen „Nicht v. Neumann’scher Netzwerk– und Speicherstrukturen“

Der Antwortgenerator soll zur Simulation

- a) auditiv wahrnehmbarer, natursprachlicher Dialoge in Echtzeit und

- b) schöpferischer und origineller Assoziationsfunktionen des menschlichen Hirns ausgelegt sein.

Für a) dient der internationale Flugsicherungssprechfunk, begrenzt auf die Bodenkontrolle als Modell, für b) ein Minithesaurus mit Inhalten, die sich auf die Deuterkriterien beziehen. Es wurden weitere, dreidimensional abgeleitete, Kugelalgorithmen entwickelt. Dabei helfen die Deuterkugel-Algorithmen dem Vervollständigen der Informationstheorie und schnelleren Assoziationen sowie Findefunktionen. Alles dient zum Wahren des Echtzeitrahmens bei der Spracherkennung von 4–6 Silben/s Sprechgeschwindigkeit und 0,5–1,5 s Antwortintervall. Ergebnisse der nichtklassischen Informationstheorie ergänzen zunehmend Erkenntnisse über Anwendungen des abgelegten Inhalts für schnelle Such- und Assoziationsvorgänge in Speichern. Dies wird erforderlich bei der Spracherkennung mit selbsttätiger Inhaltzuordnung und automatischer Answersuche im Echtzeitrahmen.

## **Forschungskontakte**

Hewlett Packard, Böblingen (*Maderholz, Zeeb*)

Hewlett Packard, Böblingen (*Rochlitzer*)

IBM Forschungslabor, Heidelberg (*Schmutz*)

Kernforschungszentrum Karlsruhe (*Trauboth*)

Kernforschungszentrum Karlsruhe, IDT (*Jaeschke*)

MPI für Biologische Kybernetik, Tübingen (*Reichardt*)

### 3.1.2 Abteilung Computer–Systeme

Leiter	<i>Burkhardt</i>
Mitarbeiter	<i>Bodenschatz (bis 31.5.), Homeister, Krause, Moser, Rust, Walter, Weiss (ab 3.5.), Zipperer (ab 1.11.)</i>

## Übersicht

Trotz der weiterhin anhaltenden Überlastung im Studentenbetrieb konnten die Forschungsvorhaben in der Abteilung weitergeführt werden, wenn auch in relativ bescheidenem Rahmen.

In den Forschungsvorhaben über parallele Systeme wurde der Einfluß von Ein- und Ausgabe, bzw. Kommunikation, auf die Leistung eines Parallelrechnersystems analytisch und experimentell an verschiedenen Konfigurationen des Systems untersucht. Die Leistung zeigt sich proportional der Prozessorleistung und umgekehrt proportional dem Kommunikationsaufwand.

Das Projekt der Leistungsermittlung eines parallelen Transputersystems für die Abarbeitung von Programmen in einer Funktionsprache konnte erfolgreich abgeschlossen werden. Die übrigen Projekte erzielten z.T. bedeutende Fortschritte.

## Multi–Transputer–Reduktionsmaschine

( Wolfgang Bodenschatz )

Im Berichtszeitraum konnte die Hardware der Multi–Transputer–Reduktionsmaschine fertiggestellt werden. Ebenso konnten die Software–Module, die zum Betrieb der Maschine notwendig sind — dies ist ein Funktionsprachensystem zum Ablauf auf dem PC und ein Laufzeitsystem zum Ablauf auf den einzelnen Reduktionseinheiten — vervollständigt werden.

Experimentell wurde der Einfluß von Systemparametern und Parallelitätskörnung auf das Laufzeitverhalten untersucht. Verschiedene Strategien zur Aufgabenverteilung wurden analysiert und in Bezug auf erreichbare Lastverteilung beobachtet. Geeignete Netzwerktopologien wurden diskutiert und hinsichtlich ihres Leistungsverhaltens miteinander verglichen. Die Messungen wurden an Hand von Funktionsprachenprogrammen durchgeführt, die in der Notation der hierfür konzipierten Sprache FUN geschrieben wurden. Dabei zeigte sich, daß die mit 8 Reduktionseinheiten ausgestattete

Multi–Transputer–Reduktionsmaschine — abhängig von der gewählten Netzwerktopologie — eine bis zu 6,2 fache Geschwindigkeitsleistung der Maschine mit nur einer Reduktionseinheit erbringt. Bei der Zusammenschaltung von 16 Reduktionseinheiten zu einem Hypercube der Dimension 4 konnte bei den verwendeten Testprogrammen eine bis zu 9–fache Geschwindigkeitssteigerung erzielt werden.

## Paralleles Booten von Transputern

( Dieter Homeister )

Die meisten Transputernetzwerke werden seriell durch eine Kette oder einen Baum von Transputern geladen. Verwenden alle Transputer dasselbe Programm, wird dennoch meist für jeden Transputer der Ladevorgang wiederholt. Bei hoher Prozessorzahl und großen Programmen nimmt das Urladen der Programme merklich Zeit in Anspruch. Das hier entwickelte Verfahren ermöglicht es, viele Transputer gleichzeitig und direkt vom Host zu laden, ohne daß dies wesentlich länger dauert als das Laden eines einzelnen Transputers. Einzige Bedingung ist, daß alle Transputer dasselbe Programm erhalten sollen. Auf diese Art können auch große Datenmengen parallel an mehrere Transputer verteilt werden. Ein Beispiel soll dies verdeutlichen. 32 Transputer sollen mit 2 MByte identischen Daten versorgt werden. Die serielle Übertragung dauert bei 10 MBit/s im günstigsten Fall 70 Sekunden, die gleichzeitige Übertragung an alle Transputer dagegen nur 2,5 Sekunden.

Transputer kommunizieren über sogenannte Links, bidirektionale serielle Zweidraht–Verbindungen mit bis zu 20 MBit/s. Sie arbeiten mit einem Byteprotokoll mit Handschake für jedes Byte auf der entgegengesetzten Leitung. Zum parallelen Booten werden Transputerlinks verzweigt. Dies ist zwar vom Entwickler der Transputer nicht vorgesehen worden, kann aber dennoch unter Einhaltung aller Spezifikationen realisiert werden. Bei der einfachen Variante des parallelen Bootens fließt der Datenstrom vom Master gleichzeitig zu allen Slaves, das Handshake wird aber nur von einem Slave gesteuert. Alle weiteren Slaves laufen jetzt synchron zum ersten Slave „blind“ mit.

Bei der verbesserten Variante wartet eine Zusatzlogik bei jedem gesendeten Byte auf die Handshake–Signale der Slaves. Erst wenn von allen Slaves ein Handshake empfangen wurde, wird ein Handshake–Paket an den Master gesendet. Diese Anordnung kann als serieller Datenbus mit einem Sender und vielen Empfängern interpretiert werden. Die Zusatzlogik wurde für 6 Transputer in einem einzigen 24–poligen IC realisiert. Mit zwei ICs konnten 12 Transputer parallel gebootet werden. Hier fanden Programmierbare Logikbausteine (GALs) Verwendung. Umfangreiche Meßreihen ergaben eine völlige Fehlerfreiheit.

Die erste Variante (ohne Zusatzlogik) erwies sich als nicht vollständig zuverlässig, weshalb der in Hardware realisierten Variante der Vorzug zu geben ist. Bei den Messungen wurde als Nebeneffekt festgestellt, daß sich Transputer nicht vollständig deterministisch verhalten. Zeitmessungen können mit Fehlern von bis zu 8 % behaftet sein.

---

---

Dieses Verhalten wurde zwischenzeitlich von anderer Seite bestätigt. Für die Zeit dieser Datenübertragung wird also die zwischen Transputern übliche Punkt-zu-Punkt-Verbindungsstruktur zu einer bus-artigen Struktur mit einem Sender und mehreren Empfängern umkonfiguriert. Bei diesem Verfahren besteht keine Grenze für die Zahl der Slaves. Nach der Boot-Phase wird das Transputernetzwerk wieder konventionell betrieben.

## Automatisierung des Entwurfsprozesses für integrierte Schaltungen

( Stefan Rust )

Das Forschungsprojekt beschäftigt sich mit der Unterstützung der Systementwurfsphase von integrierten Schaltungen und Systemen. Dabei wird untersucht, wie der Entwurfsablauf grafisch unterstützt werden kann und welche Analyse-, Synthese- und Optimierungstools für die Automatisierung von einzelnen Entwurfsschritten benötigt werden. Seit 1987 wird an einer experimentellen Entwurfsumgebung SYSEDIT gearbeitet, an der die Konzepte erprobt werden können.

Die interaktive, grafische Entwurfsumgebung SYSEDIT enthält Editoren für die hierarchische Beschreibung der Systemspezifikation, der Architektur und des Floorplans/Layout. Verhaltensmodelle können in C oder DACAPO-II erstellt und in das System eingelesen werden. Die automatische Erzeugung von Simulationsmodellen der Systembeschreibung, der Architekturbeschreibung und der Logikbeschreibung dienen zur Verifikation des Entwurfs. Eine Toolbox und eine integrierte regelbasierte Entwurfssprache unterstützen die Optimierung der Verhaltensbeschreibung, die Generierung der Architekturbeschreibung und die Bewertung der Architektur.

Die Entwurfsumgebung SYSEDIT wurde im Berichtszeitraum soweit fertiggestellt, daß ein erster Chipentwurf durchgeführt werden konnte. Ausgehend von Anforderungen aus der Echtzeit-Bildverarbeitung wurde die Verhaltensbeschreibung eines Sortieralgorithmus erstellt und optimiert.

Mit SYSEDIT wurde dann der Architekturentwurf und der Logikentwurf eines Sortierchips durchgeführt und verifiziert. Dabei wurde bereits auf der Verhaltensebene auf einen testbaren Entwurf und eine hohe Regularität geachtet.

In Zusammenarbeit mit dem Institut für Mikroelektronik (IMS), Stuttgart, wurde eine Übertragungsprozedur erstellt, mit welcher der Logik-Entwurf von SYSEDIT automatisch an die IMS-Zellbibliothek angepasst und in die am IMS vorhandenen Entwurftools übertragen werden konnte. Damit besteht nun die Möglichkeit, Chipentwürfe für Forschung und Lehre am IFI zu erstellen und am IMS fertigen zu lassen.

Die weitere Arbeit an der SYSEDIT Entwurfsumgebung wird sich nun hauptsächlich auf die Erprobung und Erweiterung der Toolbox-Funktionen konzentrieren.

---

## Multi–Transputer–System

( Jörg Walter )

Topologien für Multi–Transputer–Systeme sind meist von vornherein durch die Kommunikationsverbindungen (Links) festgelegt. Wird eine dynamische Konfiguration des Netzwerks verlangt, kann ein Link–Vermittler–Baustein der Firma Inmos eingesetzt werden. Dieser Baustein besitzt 32 Link–Ein– und Ausgänge. Jeder Ausgang kann per Software mit einem beliebigen Eingang verbunden werden. Die Programmierung erfolgt über ein zusätzliches Configuration–Link.

Mit 4 dieser Bausteine wurde ein erweiterungsfähiges Vermittler–Board entwickelt, das die Verbindung von 16 Transputern erlaubt. Dabei sind alle Topologien (soweit es die Hardware des Transputers erlaubt) realisierbar. Die Karte ist maximal mit 8 Link–Switch Bausteinen bestückbar, so daß in dieser Ausbaustufe 32 Transputer angeschlossen werden können.

Programmiert werden die Link–Switches vom PC aus. Dies kann durch das Laden eines Konfigurationsfiles geschehen, oder durch „Handvermittlung“, wobei einzelne Verbindungen (ähnlich einer Telefonvermittlungsstelle) von Hand gesteckt werden. Die so entstandene Konfiguration kann als Konfigurationsdatei abgespeichert werden.

Von den Transputern aus ist eine Umkonfiguration nur auf dem Umweg über den PC möglich. Der Grund dafür liegt darin, daß bei der Verteilung von Programmen auf Transputernetzen unbekannter Topologie oft eine Art Wurmprogramm verwendet wird. Dieses enthält eine Codesequenz, die den Link–Switch programmiert. Dadurch würde die zuvor programmierte Konfiguration zerstört. Der Zeitverlust für den Umweg kann in Kauf genommen werden, da an den PC nur der Konfigurationsbefehl und ein Dateiname übergeben werden muß, der Konfigurations–Transputer ist dann frei für andere Aufgaben.

Das Wurmprogramm zur Erkundung beliebiger Transputernetze wurde in Zusammenarbeit mit dem Projekt Paralleles Booten entwickelt und in einer parallelen C–Version programmiert. Auf dem Root–Transputer gestartet, wird sukzessive jedes Link auf mögliche Nachbarn untersucht. Wird ein Nachbar gefunden, so überträgt sich der Wurm auf den Nachbarn und setzt dort die Untersuchung fort. Ist das ganze Netz erkundet, wird eine Tabelle mit den Link–Verbindungen auf dem PC ausgegeben.

---

### 3.1.3 Abteilung Dialogsysteme

Leiter	<i>Gunzenhäuser</i>
Wiss. Mitarbeiter	<i>Bauer (bis 31.5.), Böcker, Fehrle (bis 30.4.), Hanakata, von der Herberg (bis 31.3.), J.Herczeg (ab 1.7.), Holz (bis 31.3.), Knopik, Kunkel (bis 31.3.), Maier, M.Rathke (bis 30.6.), Schwab, Schweikhardt, Strothotte (bis 31.7.), Weber (beurlaubt bis 15.6.)</i>
Programmierer	<i>Krepplein, Werner</i>
Verwaltungsangestellte	<i>R.Martin, Wieland</i>
Hilfskräfte	<i>Bayer, Förch, Kochanek, Milan, Sander, Strobel, Ruhland, Wichert, Wihofsky</i>

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

### 1.1 Überblick

In den Forschungsvorhaben der Abteilung Dialogsysteme werden Methoden aus der praktischen und der theoretischen Informatik sowie der Künstlichen Intelligenz und der kognitiven Psychologie auf Aufgabenstellungen der Mensch–Computer–Kommunikation angewandt.

Im Berichtsjahr befaßten sich die einzelnen Vorhaben mit

- (1) prototypischen Methoden und Werkzeugen zur Gestaltung von wissensbasier-ten Benutzerschnittstellen insbesondere im Bürobereich (Projekt INFORM im BMFT–Verbundprojekt WISDOM),
- (2) der Verbesserung einer natürlichsprachlichen Benutzerschnittstelle für ein Expertensystem (Projekt KEYBIT, gefördert von IBM Deutschland),
- (3) rechnerunterstützten Hilfsmitteln für Blinde und Sehbehinderte (Forschungsgruppe Angewandte Informatik für Blinde),
- (4) der Integration von Text, Graphik und Sprache in interaktiven Systemen sowie
- (5) der Weiterentwicklung von objektorientierten Sprachen zur Wissensrepräsenta-tion.

Das Berichtsjahr war gekennzeichnet durch den Abschluß der unter (1), (2) und (4) genannten Vorhaben; die Abschlußarbeiten umfaßten die Fertigstellung von mehreren Dissertationen, einer Habilitation, einer Reihe von Publikationen und Buchbeiträgen, sowie Präsentationen der erstellten Prototypen.

Die Arbeit der Forschungsgruppe INFORM, die sich seit 1980 mit der wissensbasierten Mensch–Computer–Kommunikation befaßt, bildete bis April 1989 einen Teil des Verbundvorhabens WISDOM, in dem mit Partnern aus der Industrie — insbesondere der Firma TA Triumph Adler AG Nürnberg —, der Gesellschaft für Mathematik und Datenverarbeitung und dem Fraunhofer Institut für Arbeitswirtschaft und Organisation Stuttgart erfolgreich zusammengearbeitet wurde. Ein Teil der bis 1988 erzielten Ergebnisse wurde im Buch „Prototypen benutzergerechter Computersysteme“ (Verlag W.deGruyter, 1988) zusammengefaßt. Eine weitere Buchveröffentlichung über Ergebnisse des Projekts konnte 1989 als Manuskript abgeschlossen werden.

Die Arbeiten der Forschungsgruppe INFORM konnten in der 2. Jahreshälfte in Zusammenarbeit mit der Firma SEL, Pforzheim, im Rahmen eines ESPRIT–Projekts weitergeführt werden.

Die seit 1978 bestehende Arbeitsgruppe „Angewandte Informatik für Blinde“ wurde u.a. durch die Firma Hewlett–Packard, Böblingen, durch die Bereitstellung neuester Geräte gefördert. Sie tragen weiterhin intensiv zur Ausbildung blinder und sehbehinderter Informatikstudenten bei. Das German Chapter der ACM (Association for Computing Machinery) spendete in dankenswerter Weise eine moderne Blindenschriftausgabezeile für einen der blinden Studenten.

Die in einem früheren DFG–Projekt entwickelte Stiftplatte wurde zu einem interaktiven Werkzeug für Blinde weiterentwickelt. Von besonderem Interesse sind hier die im Rahmen einer abgeschlossenen Dissertation entwickelten neuen Dialogverfahren unter Verwendung von Gesten.

Die Arbeiten im Forschungsschwerpunkt „wissensbasierte, intelligente Mensch–Computer–Kommunikation“, der gemeinsam mit Prof. Bullinger und Prof. Rohrer eingerichtet und vom Land Baden–Württemberg bis Ende 1988 gefördert wurde, konnte im Berichtsjahr durch eine Habilitation erfolgreich abgeschlossen werden. In diesem Bereich zeichnet sich eine Kooperation mit einem externen Forschungszentrum ab.

Auch im Berichtsjahr wurden die Aktivitäten der Abteilung im Bereich der Informatikausbildung weitergeführt. Beispielsweise wurde die für die Informatik–Grundausbildung an der Universität Stuttgart entwickelte Konzeption erweitert: Mit Beginn des WS 1989/90 konnte die 1986 eingeführte Lehrveranstaltung „Grundlagen der Informatik“ dreigeteilt werden: (1) für den Studiengang „technisch orientierter Diplomkaufmann“ (Lehrauftrag Direktor Dr.Glatthaar), (2) für die Studiengänge Luft– und Raumfahrttechnik, Physik und für andere naturwissenschaftliche und technische Fachrichtungen (Professor Gunzenhäuser) und (3) für die Studiengänge der Maschinenbaufakultäten (Professor Baitinger, IPVR).

Gemeinsam mit der Abteilung „Intelligente Systeme“ wurde im WS 89/90 eine neue Lehrveranstaltung „Interaktive und intelligente Systeme“ (Vorlesung, Übungen, Softwarepraktikum) für Studierende im 5. Fachsemester eingeführt, die die bisherige Vorlesung „Dialogsysteme“ ablöst.

---

---

Das Interesse an der Informatikausbildung und der Computernutzung in Schulen und in der beruflichen Aus- und Weiterbildung ist unvermindert. Auch Methoden des rechnerunterstützten Lernens finden durch die zunehmende Anzahl von Rechnern an Arbeits- und Ausbildungsplätzen auffallend starke Verbreitung. Mitglieder der Abteilung befaßten sich auch im Berichtsjahr wieder mit einer Reihe von unterschiedlichen Aufgaben aus der Informatikausbildung, unter anderem im Rahmen der 1989 neu gegründeten Akademischen Software Kooperation (ASK) und des von ASK durchgeführten 1. bundesweiten Software-Wettbewerbs.

## **1.2 Forschungsgruppe INFORM: Projekt WISDOM (Wissensbasierte Systeme zur Bürokommunikation – Dokumentenverarbeitung, Organisation, Mensch–Computer–Kommunikation)**

*gefördert vom BMFT und von der Firma TA Triumph Adler AG*

*( Dorothea Bauer (bis 31.3.89), Dieter Böcker, Harald von der Herberg (bis 31.3.89),*

*Dieter Holz (bis 31.3.89), Dieter Maier, Martin Rathke (bis 30.5.89),  
Thomas Schwab )*

Im Verbundprojekt WISDOM, das zum 31.3.89 endete, wurden insbesondere folgende technischen Aufgaben abgeschlossen, die die Weiterverwendung der im Forschungsvorhaben entwickelten prototypischen unterstützenden Systeme auf unterschiedlichen Rechnersystemen ermöglichen:

An der Benutzeroberfläche (dem Bürodesktop) wurden Maßnahmen zur Verbesserung des Laufzeitverhaltens durchgeführt sowie einige noch bestehende Inkonsistenzen ausgeräumt. Eine geplante Evaluation mit „naiven“ Benutzern konnte jedoch aufgrund fehlender Mittel nicht im geplanten Umfang durchgeführt werden.

Um die Diskrepanz zu den in neueren Büroumgebungen vorkommenden Rechnertypen zu verringern wurde der innerhalb des Projektes unter Genera auf Symbolics entwickelte Bürodesktop für Standard-Rechner verfügbar gemacht. Grundlage dieser Arbeiten war die ILA-X-Windows Implementierung auf Symbolics. Die Anwendung läuft hierbei als X-Client auf einer Symbolics und ist über einen beliebigen X-Server im Netz zugänglich. Bildschirmgestaltung und Interaktion bleiben unverändert.

Zur Überwindung der Abhängigkeiten der erstellten Anwendungen von dem zugrundeliegenden, innerhalb des Projekts entwickelten Fenstersystem INFWIN wurde die Umstellung des INFORM Benutzerschnittstellen-Baukastens auf X-Windows untersucht. Das erarbeitete Konzept wurde in Form eines Arbeitsberichts veröffentlicht.

---

### **1.3 Forschungsgruppe INFORM: Projekt MCPS: Multimedia Communication, Processing and Representation**

*gefördert von der Firma SEL*

*( Jürgen Herczeg (seit 1.6.89), Dieter Maier (bis 30.6.89), Thomas Schwab )*

In diesem Vorhaben arbeitet seit 1.4.1989 die Forschungsgruppe INFORM mit einer Forschungsgruppe der Firma SEL, Pforzheim, innerhalb des europäischen Projektes RACE 1038.MCPR als Unterauftragnehmer zusammen.

Am Gesamtprojekt sind noch mehrere internationale Industriepartnern von SEL beteiligt. Es hat das Ziel, schnelle Breitbandkommunikationstechniken zu entwickeln und anhand von konkreten Anwendungssystemen zu demonstrieren. Die Breitbandkommunikation soll zum Austausch von Multimedia–Information genutzt werden. In Zusammenarbeit mit SEL wird ein Demonstrationssystem entwickelt, das es dem Benutzer erlaubt, interaktiv Multimedia–Information abzufragen und zu inspizieren. Als Anwendungsbereich wurden Reiseinformationen über Frankreich gewählt.

Die zugrundeliegenden Daten sind objektorientiert repräsentiert und bilden ein Netz aus heterogenen Informationsknoten, die ihrerseits Verweise auf Multimedia–Informationen wie z.B. Bilder, Videos und Sprache enthalten können. Das in Entwicklung befindliche System verbindet formale Anfragemechanismen mit der Vorgehensweise des Hypertext–Ansatzes. Bei SEL wird ein Hypermedia–Browser erstellt, der die Multimedia–Information geeignet präsentiert und es erlaubt, sich beliebig durch den Informationsraum zu bewegen. Im Rahmen des Unterauftrags wird dazu eine Komponente erstellt, die es erlaubt, interaktiv formale Anfragen zur Eingrenzung des Informationsraums zu erstellen und auszuwerten. Die Ergebnisse einer solchen Anfrage dienen als Einstiegspunkte für den Browser.

Im einzelnen wurden im Berichtsjahr von der Forschungsgruppe INFORM folgende Arbeiten durchgeführt:

Informationen aus einem Reiseführer über Frankreich wurden aufgearbeitet und objektorientiert als Klassen und Instanzen repräsentiert. Für die Benutzerschnittstelle wurde ein User Interface Toolkit entwickelt. Es stellt ein mächtiges Werkzeug für Common Lisp auf der Basis von X Windows dar. Dabei wurden die Ansätze der innerhalb des WISDOM–Projekts entwickelten Systeme WLisp und InfUims übernommen.

Mit Hilfe von Xit wurde der Prototyp HyperQuery für die Anfragekomponente implementiert. Dieser ermöglicht die inkrementelle Erstellung einer formalen Anfrage mittels direkter Manipulation. Durch menügesteuerte Interaktion werden syntaktisch fehlerhafte Anfragen ausgeschlossen. Zu jedem Zeitpunkt kann der Benutzer die Anfrage auswerten lassen, worauf die Menge der zutreffenden Instanzen dargeboten wird. Anschließend kann die Anfrage erneut modifiziert werden, so daß ein exploratives Vorgehen ermöglicht wird.

---

## 1.4 Forschungsgruppe „Angewandte Informatik für Blinde“

( Waltraud Schweikhardt (federführend), Gerhard Weber (ab 1.6.89),  
Alfred Werner )

Die Arbeit der Forschungsgruppe Angewandte Informatik für Blinde hatte zwei Schwerpunkte:

- (1) Neue Interaktionstechniken für Blinde: Es wurden neue Dialog- und Interaktionstechniken für Blinde entwickelt und prototypisch eingesetzt. Die lesenden Hände eines blinden Rechnerbenutzers erfassen die für Sehende auf einem Bildschirm dargestellten Informationen auf der Stuttgarter Stiftplatte. Sie wurde im Rahmen eines früheren DFG-Projekts entwickelt. Sie dient seit 1988 auch als berührungsempfindliches Eingabegerät für Gesten der Hände und der Finger. Als „Eingabesprache“ wurde die Beschreibungssprache FINGER definiert und auf einer Workstation vom Typ SUN 3/60 in der Programmiersprache C implementiert.
- (2) Tastbare Wiedergabe von Dokumenten, die mit Hilfe eines Scanners erfaßt wurden: Erfahrungen, die im DFG-Projekt „Bildschirmtext als rechnerunterstütztes Kommunikationsmittel für Blinde“ auf dem Gebiet der tastbaren Wiedergabe von Text und Graphik enthaltenden Bildschirmhalten gemacht wurden, werden bei dieser neuen Aufgabe eingesetzt.

Das optisch erfaßte Material wird in Text- und Graphikbereiche gegliedert, die dann gesondert weiterverarbeitet werden. Schließlich wird das Dokument in einer umstrukturierten Gesamtform, die dem voluminösen taktilen Material angepaßt ist, tastbar wiedergegeben.

Seit Dezember 1989 steht ein neuer Drucker zur Verfügung, mit dem tastbare Graphiken und Texte in ausgezeichneter Qualität erstellt werden können. Die dazu erforderlichen Rechnerprogramme wurden im Berichtsjahr fertiggestellt.

Herr Weber war von November 1988 bis Juni 1989 als Informatiker an der Overbrook School for the Blind in Philadelphia, PA, USA, wo er ein Gerätelabor mit rechnerunterstützten Blindenhilfsmittel einrichtete und damit die Informatikausbildung im International Program der Schule modernisierte.

Die Arbeiten an rechnerunterstützten Hilfen für Gehörlose wurden in Zusammenarbeit mit der Paulinenpflege Winnenden fortgeführt. Es wurde ein Bilderlexikon für Gehörlose erstellt. Die Dialogkomponente erlaubt, weitere Bilder aufzunehmen und Teillexika zusammenzustellen. Die Bilder können mit Hilfe einer „Maus“ erzeugt werden oder von einem Scanner als Schwarz–Weiß–Bilder erfaßt worden sein.

## **1.5 Projekt KEYBIT: Eine natürlichsprachliche Dialogkomponente für ein wissensbasiertes Beratungssystem**

( *Thomas Fehrle (bis 30.4.89), Thomas Knopik* )

In diesem Vorhaben wurden Benutzerschnittstellen für das wissensbasierte Beratungssystem KEYSTONE entwickelt und im Berichtsjahr erfolgreich abgeschlossen. Die Arbeiten im Berichtszeitraum befaßten sich im wesentlichen mit drei Komponenten:

- (1) Ergänzende Untersuchungen zu einem Parser mit Syntax–Semantik–Interaktion für natürlichsprachliche Eingabesätze.
- (2) Implementierung einer wissensbasierten Klärungskomponente, die den Benutzer in Dialogsituationen unterstützt, in denen korrekte Eingaben vom System nicht verarbeitet werden können. Die Klärungskomponente erläutert dem Benutzer die Ursachen, weshalb die Eingabe von KEYSTONE nicht verarbeitet werden kann und gibt zusätzlich gezielte Vorschläge zur Korrektur dieser Eingabe. Um die Möglichkeiten der Klärung zu erweitern, wurde ein zu (1) alternativer semantikorientierter Parser implementiert.
- (3) Erweiterung einer Wissenserwerbskomponente für den Sachgebietsexperten um Verfahren des „Machine Learning“ — wie z.B. Lernen aus Beispielen und Analogien — und um transaktionsorientierte Verfahren zur Behandlung von Ausnahmesituationen. Sie wird damit zu einem interaktiven Werkzeug für den Experten, das ihn bei der Aufgabe des Wissenserwerbs „intelligent“ unterstützt und dabei jederzeit die Konsistenz der Wissensbasis erhält.

## **1.6 Objektorientiertes Programmieren: Implementierung der objektorientierten Sprache CTALK**

( *Kenji Hanakata* )

Im Berichtsjahr wurden zwei verschiedene Benutzerschnittstellen für die Sprache CTALK prototypisch entwickelt. Eine Fensterschnittstelle wurde auf der Basis von MS–Windows, eine andere auf der Basis von X–Windows entworfen. Die erstgenannte wurde auf einem 386–PC unter UNIX V (Interactive System) implementiert, welche eine Schnittstellensoftware VP/ix anbietet, mit deren Hilfe man von einem UNIX–Prozeß aus mit einem MS–DOS–Prozeß kommunizieren kann. Dabei hat sich gezeigt, daß bei einem Einprozessorsystem ein Synchronisationsproblem in der pipe–Kommunikation auftritt, so daß der MS–Prozeß als Ein–/Ausgabeprozeß für UNIX nicht verwendbar ist.

---

Ferner wurde festgestellt, daß MS-Windows als Benutzerschnittstelle für die Mächtigkeit von CTALK zu beschränkt ist. Eine MS-Windows-Schnittstelle kann daher wohl als Endbenutzerschnittstelle verwendet werden, nicht aber als Entwicklungsumgebung.

Ein Prototyp einer X-Window-Schnittstelle für CTALK wurde erfolgreich auf einem Rechnersystem SUN (3/50, 60, 4 Sparc) implementiert. Sie muß noch hinsichtlich der Benutzerakzeptanz evaluiert und weiter verbessert werden.

## 2. Informatikausbildung

Auch im Berichtsjahr wirkte der Abteilungsleiter in mehreren Gremien mit, die sich mit der Informatikausbildung und dem Informatikunterricht befassen, so im Fachbereich „Ausbildung“ und im Fachausschuß „Informatikausbildung in Studiengängen an Hochschulen“ der Gesellschaft für Informatik e.V.

Auch als Vorsitzender des Fakultätentages Informatik an Hochschulen der Bundesrepublik und West-Berlins hatte er mannigfache Aufgabenstellungen der Informatikausbildung und –weiterbildung initiativ zu verfolgen und in unterschiedlichen Gremien (u.a. der Westdeutschen Rektorenkonferenz) zu vertreten.

Der Abteilungsleiter war im Berichtszeitraum auch Mitglied von entsprechenden Beratungsgremien an den Universitäten Hagen, Karlsruhe, Tübingen und Ulm. Es wurde als Vertreter der Informatikhochschulen in das Sachverständigungsgremium berufen, das die Akademische Software Kooperation (ASK) berät; innerhalb dieser Aktivität war er auch Juror des 1. Software Wettbewerbs der ASK für Arbeiten aus dem Bereich der Informatik.

Gemeinsam mit Professor Mandl, Tübingen, wurde im Berichtsjahr zum dritten Mal eine Fachtagung „Intelligente Lernsysteme“ an der Universität Tübingen durchgeführt, an der mehr als 50 Fachkollegen aus der Informatik, der Künstlichen Intelligenz, der Linguistik und der Kognitiven Psychologie aktiv teilnahmen. Die GI-Fachgruppe „Intelligente Lernsysteme“ bereitet im Rahmen der 20.Jahrestagung der Gesellschaft für Informatik 1990 ein Fachgespräch vor, das Herr Gunzenhäuser leiten wird.

## Forschungskontakte

Carleton University (Ottawa, Canada), School of Computer Science (*J.-R. Sack*)

Carnegie-Mellon-University, Dep. of Computer Science, Pittsburgh, PA, USA  
(*A. Newell, H.A. Simon, G. Steele, J. Carbonell, Ph. Hayes*)

Deutsche Blindenstudienanstalt, Marburg a.d.Lahn (*K. Britz, R.F.V. Witte*)

Deutsches Taubblindenvwerk, Hannover (*F. Zekel*)

Electrotechnical Laboratories, Tsukuba, Japan (*Yokoyama, Ishizaki*)

- Firma F.H. Papenmeier GmbH & Co.KG, Schwerte (*Bornschein*)  
Forchungsinstitut für Anwendungsorientierte Wissensverarbeitung an der  
Universität Ulm (*W. Radermacher, D. Rösner, W.-F. Riekert*)  
Fujitsu Laboratories, Kawasaki, Japan (*T. Hayashi, Uchida, Kobe*)  
Gesellschaft für Informatik e.V. Fachgruppen „Intelligente Lernsysteme“ und  
„Informatik in der Schule“  
Gesellschaft für Mathematik und Datenverarbeitung, St. Augustin  
(*Hoschka, Dzida, Wißkirchen*)  
Hewlett Packard Research Laboratory, Palo Alto, Calif. USA (*J. Laubsch*)  
Hewlett Packard Research Laboratory, Böblingen (*A. Sonntag, Ritter*)  
IBM Deutschland GmbH, Bereich Wissenschaft (*W. Glatthaar*)  
IBM Deutschland GmbH, Wissenschaftliches Zentrum Heidelberg  
(*A. Blaser, W. Wernecke, Th. Strothotte*)  
Institut für Arbeitswirtschaft und Organisation der Fraunhofer-Gesellschaft,  
Stuttgart (*H.-J. Bullinger, J. Ziegler*)  
Institut für Informatik der Universität Karlsruhe  
(*A. Schmitt, Blindenausbildungsprojekt*)  
Institut für Maschinelle Sprachverarbeitung der Universität Stuttgart  
(*Chr. Rohrer*)  
Institute of New Generation Computer Technology, ICOT, Tokyo, Japan  
(*Fuchi, Furukawa*)  
International Association of Pattern Recognition (*T. Pavlidis*)  
Massachusetts Institute of Technology, Artificial Intelligence and LOGO  
Laboratories, Cambridge, MA., USA (*H. Abelson, A. di Sessa*)  
METEC Ingenieur GmbH, Medizintechnik – Feinwerktechnik, Stuttgart  
Nikolauspflege Stuttgart, Private Bildungsstätte für Blinde und Sehbehinderte  
Overbrook School for the Blind, International Program, Philadelphia, PA, USA  
(*Annke Rimrott*)  
Siemens Forschungszentrum, Datenverarbeitung/Benutzerforschung, München  
(*K. Wimmer, M. Schneider-Hufschmidt*)  
Stanford Research Institute, Artificial Intelligence Center, Menlo Park, CA., USA  
(*D. Rosenschein*)  
TA Triumph Adler AG, Forschungszentrum, Nürnberg (*R. Lutze*)  
SEL Forschungszentrum Pforzheim (*A. Lesch*)  
The University of Western Ontario, Dep. of Comp.Science, London, Canada  
(*Helmut Jürgensen*)  
Tokyo Institute of Technology, Tokyo, Japan (*Tanaka*)  
University of Colorado at Boulder, Dep. of Computer Science  
(*G. Fischer, C. Kintsch, A. Lemke*)  
University of Kyoto, Japan (*M. Nagao, Nakamura, Tsujii*)
-

---

University of Linköping, Sweden, Computer Science Department  
(*E. Sandewall, R. Karlsson*)

University of Massachusetts, Amherst, MA., USA (*V. Lesser, E. Rissland*)

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

Uppsala Universitet. Institutionen för teknisk databehandling (*Sjöberg*)

### 3.1.4 Abteilung Intelligente Systeme

Leiter	<i>Lehmann</i>
Mitarbeiter	<i>Burkert, Castro (ab Juni), Emele (ab November), Forster (ab September), Heid (IMS), Kehl (bis Oktober), Langjahr (ab April), Momma (IMS), Ch. Rathke, Tausend, Wirth (bis September), Zajac (IMS, ab November)</i>
Hilfskräfte	<i>Bell, Christ (IMS), Gerbing, Herrmann, Heyn (IMS), König, Koppenhöfer, McKinnon (IMS), Mezger, Philipp, Resch, Stöhr (IMS)</i>

#### Abteilung Intelligente Systeme: Überblick

( *Egbert Lehmann* )

Die Forschungsarbeiten der Abteilung Intelligente Systeme sollen Beiträge zur Repräsentation und maschinellen Verarbeitung umfangreicher Wissensbestände liefern. Im Rahmen eines längerfristig angelegten Vorhabens wird vor allem an der Entwicklung von Verfahren zur inhaltsorientierten Verarbeitung natürlicher Sprache (in schriftlicher Form) und zur objektorientierten Repräsentation begrifflichen Wissens gearbeitet. Angestrebt wird die Entwicklung eines Programmsystems, das durch Einlesen von unbekannten deutschen Texten oder durch gezieltes Führen von natürlichsprachlichen Mensch–Maschine–Dialogen seine Wissensbasis inkrementell erweitern und damit seine globale Leistungsfähigkeit verbessern kann.

Die ersten hierzu realisierten Schritte beinhalten die Schaffung einiger notwendiger Grundkomponenten für maschinelle Sprachanalyse (Wörterbuch, Parser, Regelgrammatik, Einlesen von Texten) sowie Ansätze zur Einbeziehung von Lernverfahren zum Spracherwerb (z.B. Erlernen von Grammatikregeln oder von lexikalischem Wissen aus Satzbeispielen).

Schwerpunkt eines BMFT–geförderten Gemeinschaftsprojekts ist die Textgenerierung aus formalen Bedeutungsdarstellungen. Dabei finden Mehrsprachigkeit und Anforderungen der maschinellen Sprachübersetzung stärkere Beachtung.

Soweit es die begrenzten Kapazitäten zuließen, wurde außerdem auch an der Weiterentwicklung eines praxisnahen wissensbasierten Konfigurierungssystems gearbeitet.

Eine neue Lehrveranstaltung *Interaktive und Intelligente Systeme* wurde gemeinsam mit der Abteilung Dialogsysteme durchgeführt. Vorbereitende Arbeiten zur Gestaltung eines Software–Praktikums (SOPRA II) für diesen Studienschwerpunkt wurden begonnen.

---

## Analyse und Repräsentation von Begriffsdefinitionen

( *Gerrit Burkert* )

Ein Forschungsziel in der Künstlichen Intelligenz ist die Nachbildung intelligenten Verhaltens auf Computern. Hierbei spielt das Vorhandensein umfangreicher Bestände an (Welt-) Wissen eine wesentliche Rolle. Schwerpunkt der Arbeiten sind Untersuchungen zur Wissensakquisition durch syntaktisch-semantische Analyse natürlichsprachlicher Begriffsdefinitionen (wie sie beispielsweise in Lexika vorkommen) und deren interne Darstellung in einer operationalen Wissensbasis.

Wegen des hohen Grades struktureller Mehrdeutigkeiten bei der Analyse natürlicher Sprache wurde ein den spezifischen Anforderungen genügender Parser (Chart Parser) entworfen und in Common Lisp auf TI-Explorer und Sun-Rechnern implementiert. Beim Chart Parser handelt es sich um einen nichtdeterministischen Parser, der alternative Lesarten und Hypothesen zur weiteren Analyse in einer geeigneten Datenstruktur, der *Chart*, speichert.

In der aktuellen Version arbeitet der Parser mit einer kontextfreien Grammatik und einem Lexikon von ca. 2500 Einträgen. Eine zusätzlich implementierte Testumgebung mit Funktionen zur Zeitmessung, zur graphischen Ausgabe von Baumstrukturen und zum Zugriff auf die Chart erlaubt es, mit den verschiedenen Möglichkeiten des Parsers zu experimentieren (z.B. bottom-up, top-down oder bidirektionale Analyse). Der Parser wird derzeit in ein größeres System mit anderen Komponenten zur Sprachanalyse (Scanner, morphologische Analyse) integriert.

Das zu entwickelnde System soll ein großes Vokabular und ein gewisses Maß an Lernfähigkeit besitzen, so daß es in unterschiedlichen Bereichen eingesetzt werden kann. Eingegebene Sätze müssen auch dann noch inhaltlich verstanden werden, wenn dem System einzelne Worte und Wendungen der benutzten Sprache unbekannt sind. Im Laufe der Benutzung soll sich der Wortschatz und die Wissensbasis im Einklang mit den jeweiligen Anwendungsbedürfnissen erweitern.

## XCONFIG

( *Christian Rathke* )

Aufbauend auf einem Expertensystem, das im Rahmen einer Diplomarbeit auf PC-Basis in GCLisp implementiert worden war, wurde ein Redesign des Systems mit direkt-manipulativer Benutzerschnittstelle vorgenommen. Die Aufgabe des Expertensystems besteht darin, Software- und Hardware-Module für CNC-gesteuerte Maschinen zu konfigurieren. Dabei auftretende wirtschaftliche und technische Randbedingungen sind wesentliche Einflußgrößen auf diese Konfigurationsaufgabe.

Ziel ist ein *interaktives* System zur Lösung des vorliegenden Konfigurierungsproblems. Viele Facetten der KI werden dabei berührt: Expertensysteme, regelbasierte Systeme,

modellbasiertes Schließen, Constraints, Wissensrepräsentation und natürliche Kommunikation.

Im Rahmen von Diplom- und Studienarbeiten wurde der Konfigurierer unter Verwendung von LUCID CommonLisp und CLOS (Common Lisp Object System) auf SUNs übertragen (Diplomarbeit Rainer König). Außerdem wurde eine direkt-manipulative Benutzerschnittstelle (Studienarbeit Alexander Negele) entwickelt, die X-Windows verwendet, um die Interaktion mit Hilfe von Fenstern, Menüs und Maus zu ermöglichen.

Geplant ist die Realisierung der Konfigurationsaufgabe als interaktiver Problemlöseprozeß, wobei das System dem Benutzer mit Vorschlägen und Kritik hilft, das Konfigurationsproblem zu lösen. Darüberhinaus soll ein Metasystem zur Anpassung an Konfigurationsaufgaben entwickelt werden, das im Unterschied zu bestehenden Expertensystemumgebungen Wissen aus dem Anwendungsbereich enthält und in gleicher Weise wie der Konfigurierer sich durch kooperatives Verhalten auszeichnet.

## **FrameTalk**

( *Christian Rathke* )

FrameTalk ist eine Erweiterung von CLOS, die im Rahmen der Objtalk-Entwicklung eingeführte Konzepte enthält (bzw. enthalten soll.) Diese sind im einzelnen: Slotbeschreibungen, Perspektiven, Regeln und Constraints. Alle Ergänzungen sollen im Sinne objektorientierter Konzepte realisiert werden, d.h. sie sollen selbst die Form von Klassen, Instanzen und Methoden haben.

**Slotbeschreibungen:** Slots in FrameTalk sind zusätzlich mit Hilfe von Defaults, Initialisierungen, Triggers und Restriktionen beschrieben. Sie bestimmen dadurch einen weiteren Teil der Slotsemantik.

**Perspektiven:** Objekte in FrameTalk sind aus „Perspektiven“ zusammengesetzt. Perspektiven sind Inkarnationen ein und desselben Objektes in unterschiedlichen Kontexten.

**Regeln:** Regeln sind lokal zu Objekten. Sie beschreiben Zustandsübergänge (d.h. Slotbelegungen) von Objekten. Sie können zu verschiedenen Zeitpunkten angestoßen werden.

**Constraints:** Constraints sind Restriktionen zwischen den Werten verschiedener Slots eines Objektes. Constraints beinhalten eine „aktive“ Komponente, die bestimmt, was im Falle einer Constraintsverletzung zu tun ist, um einen konsistenten Zustand herzustellen.

---

## 1. Regeln in FrameTalk

Das Regelsystem für FrameTalk wurde nach dem Vorbild von LOOPS gestaltet. Im Unterschied zu herkömmlichen, im Bereich von Produktionssystemen bekannten Regelsystemen sind FrameTalk–Regeln *lokal* zu Objekten. Ihre Bedingungsteile referenzieren nur Slots des Objektes, innerhalb dessen „Scope“ sie definiert sind. Evtl. sind auch Slots von Objekten referenzierbar, die von diesem Objekt über Slotpfade erreichbar sind.

## 2. Subsumption in FrameTalk

Die *Subsumptionsrelation* ist (partiell) definiert zwischen zwei Konzeptbeschreibungen in einer terminologischen Wissensbasis. Für alle Konzepte beschreibt sie eine Taxonomie. Subsumption erlaubt die Klassifizierung von Konzepten gemäß ihrer Definitionen.

In FrameTalk werden Frames, d.h. Konzepte, mit Hilfe von Slotbeschreibungen, Methoden, Regeln und Constraints definiert. Die objektorientierte Basis von FrameTalk impliziert eine *Vererbungshierarchie*, entlang der Methoden und Slots von einer Klasse auf die andere vererbt werden. Diese ist nicht notwendigerweise auch eine Generalisationshierarchie gemäß der Subsumptionsrelation.

## 3. Frame–Matching in FrameTalk

Ein Vergleichsverfahren für Frames ist eine wichtige Komponente in einem System zur Repräsentation von Wissen. Wenn Frames als Beschreibungen verstanden werden, die Objekte oder Situationen aus unterschiedlichen Blickwinkeln charakterisieren, dann ergibt sich häufig die Notwendigkeit, Verträglichkeiten zwischen den Beschreibungen festzustellen, d.h. wie gut sie miteinander in Übereinstimmung gebracht werden können. In der frame–basierte Wissensrepräsentationssprache KRL bildet der Frame–Matcher die zentrale Komponente.

Der in KRL beschriebene Matcher wurde (in Teilen) für FrameTalk entworfen. Dabei wurde besonders auf die nutzbringende Verwendung der objektorientierten Konzepte von CLOS geachtet.

## Analoges Lernen für ein sprachverarbeitendes System

( Birgit Tausend )

Analoges Schließen erlaubt es, bereits erworbene Wissen zu verwenden, um neue Probleme zu lösen oder neue Konzepte zu erlernen. Dazu wird versucht neue Fakten über ein Konzept aus bekannten Fakten eines analogen Konzepts abzuleiten.

Im Bereich des Maschinellen Lernens wurden eine Reihe von Systemen für analoges Schließen implementiert. Probleme ergeben sich vor allem beim Suchen eines analogen

Konzepts in einer Menge von bekannten Konzepten aus der Größe des Suchraums und beim Finden der besten analogen Abbildung.

Eine mögliche Verbesserung stellt die Ausnutzung von Hinweisen auf eine Analogie dar, wie z.B. „A ist ähnlich zu B“ oder „A verhält sich wie B“. Darüberhinaus kann der Kontext, in dem ein analoger Schluß angewendet werden soll, auch zur Begrenzung des Suchraums herangezogen werden. Bereits existierende Verfahren leisten dies in beschränktem Umfang, doch die zugrundeliegenden Arbeitsweisen sind noch nicht befriedigend. Zudem arbeiten die meisten Systeme isoliert und sind nicht in ein größeres wissensbasiertes System eingebunden.

Angestrebt wird die Integration des Systems für analoges Lernen in ein natürlichsprachliches System, dessen sprachliche Fähigkeiten sich im Laufe des Zeit auf diese Weise ohne direkte Eingriffe des Benutzers verbessern sollen. So soll das System aufgrund bereits verstandener Sätze neue Textausschnitte verarbeiten können, auch wenn nicht alle Information dazu vorhanden ist, z.B. syntaktische oder semantische Information zu Wörtern fehlt. Durch die Verwendung analoger Hinweise kann das fehlende Wissen induziert werden und das Verstehen des Textes fortgesetzt werden, so daß das System insgesamt flexibler und benutzerfreundlich wird.

## **Lernverfahren zur Vervollständigung von Hornklauselmengen durch inverse Resolution**

( Rüdiger Wirth )

Ziel der Forschung war ein maschinelles Lernverfahren, das eine gegebene Menge von Hornklauseln interaktiv ergänzt. Das Verfahren beruht auf einer Umkehrung des Resolutionsprinzips, der sogenannten inversen Resolution, die einen einheitlichen Mechanismus für induktive Inferenzen darstellt.

Ausgehend von einer allgemeinen Formel der inversen Resolution werden zwei Operatoren definiert, die eine inverse Resolution durchführen. Es wird ein Lernsystem konzipiert, das aus zwei Komponenten besteht. Eine Komponente arbeitet Bottom-up und versucht durch Anwendung der Operatoren der inversen Resolution, allgemeinere und kompaktere Regeln zu erzeugen. Diese Komponente wird durch ein vorgeschaltetes Top-down-Verfahren zur Vervollständigung unvollständiger Beweise ergänzt, das eine bessere Ausnutzung vorhandenen Wissens und eine gezielte Erweiterung der bekannten Regelmenge ermöglicht.

In der Arbeit wurde auf eine formale Beschreibung des Verfahrens Wert gelegt, wobei gleichzeitig die Realisierbarkeit durch eine konkrete Implementation nachgewiesen wurde. Die hervorstechendsten Eigenschaften des Verfahrens sind die Fähigkeit, neue Prädikate automatisch zu erzeugen, einfache Wiederverwendbarkeit der gelernten Regeln und die Kombination eines Bottom-up- und eines Top-down-Lernverfahrens in

---

einem einheitlichen Rahmen. Durch die Verwendung von Hornklauseln als Repräsentationssprache ergeben sich vielfältige Möglichkeiten der Weiterentwicklung dieses Verfahrens, was durch die formale Beschreibung erleichtert wird.

Im Rahmen einer Studienarbeit (Bell, 1989) wurden mehrere Fallstudien durchgeführt, um das Verfahren und die konkrete Implementation zu testen. Ziel dieser Fallstudien war die Ergänzung von Grammatiken im Definite Clause Grammar–Formalismus.

## POLYGLOSS

( *Martin Emele, Ulrich Heid, Stefan Momma, Remí Zajac* )

Das Projekt POLYGLOSS wird in Zusammenarbeit zwischen der Abteilung Intelligente Systeme und dem Institut für Maschinelle Sprachverarbeitung (Abteilung Computerlinguistik, Prof. Dr. Christian Rohrer) durchgeführt. Die Förderung erfolgt für einen Zeitraum von drei Jahren durch den Bundesminister für Forschung und Technologie (über GMD — Projektträger Fachinformation).

Ziel des Projekts POLYGLOSS ist die Entwicklung eines universell einsetzbaren multilingualen Textgenerators. Teilweise aufbauend auf den Ergebnissen des Projekts SEM-SYN wird dabei versucht, ein modulares System zu entwickeln, das es erlaubt, aus einer abstrakten Repräsentation sowohl Einzelsätze als auch kleine Textstücke in Englisch, Deutsch und evtl. Französisch oder Japanisch zu erzeugen. Bausteine der abstrakten Repräsentation sind konzeptuelle Einheiten, die in einer Taxonomie hierarchisch angeordnet sind, sowie Relationen zwischen Konzepten. Bausteine der syntaktischen Repräsentation sind lexikalische Einheiten der einzelnen Sprachen und die syntaktischen Funktionen, die diese Einheiten in Sätzen einnehmen. Bei der Formulierung der linguistischen Beschreibungen wird zu einem großen Teil Gebrauch von vorhandenen Grammatikfragmenten, insbesondere für das Deutsche und das Französische gemacht, die am IMS im Zuge anderer Projekte, wie der Begleitforschung zu EUROTRA-D und dem ESPRIT Projekt ACORD entwickelt worden sind.

POLYGLOSS repräsentiert linguistische und konzeptuelle Objekte in TFS, einem logikbasierten Formalismus, dessen grundlegender Datentyp typed feature structures sind. Der Formalismus unterstützt multiple Vererbung und erlaubt es, Beschreibungen der Eigenschaften linguistischer Objekte auf verschiedenen linguistischen Beschreibungsebenen modular zu halten. Das zugrundeliegende TFS–System kontrolliert dabei die Kompatibilität von Einzelbeschreibungen und macht daher die explizite Bereitstellung spezieller Prozeduren zur Auflösung von Konflikten zwischen verschiedenen Teilbeschreibungen überflüssig. Das Generierungssystem kann infolgedessen völlig deklarativ konstruiert werden.

1989 wurde das TFS–System implementiert, sowie ein Vorschlag für die Gesamtsystemarchitektur von POLYGLOSS gemacht. Eine Reihe von linguistischen Teilproblemen wurden bearbeitet. Der Prototyp einer Wissensquelle für syntaktisches Wissen über

lexikalische Einheiten, der die Möglichkeiten von TFS ausnützen soll, ist derzeit in Arbeit.

## **Forschungskontakte**

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

Institut für Integrierte Informations- und Publikationssysteme (GMD-IPSI),  
Darmstadt (*Hoppe, Kracker*)

IBH, Schwieberdingen (*Hammer*)

Battelle Institut, Frankfurt (*Haug*)

Inst.f.Informatik der Universität Innsbruck (*Albrecht*)

ATR, Kyoto, Japan (*Kurematsu*)

Carnegie-Mellon-University, ICMT, Pittsburgh, USA  
(*Carbonell, Levine, Nirenburg, Nyberg*)

Université Cath. de Louvain (*Lerot*)

EUROTRA-D, Saarbrücken (*Caroli, Haller, Schütz*)

CSLI, Stanford University, USA (*Bresnan, Kay, Peters, Zaenen*)

GETA, Université de Grenoble, Frankreich (*Boitet, Guilbaud, Nedobejkine*)

USC-ISI, USA (*Bateman, Kasper*)

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

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

Fraunhofer-Institut für Arbeitswissenschaft und Organisation, Stuttgart  
(*Bullinger, Hoepelmann*)

IBM Deutschland, Institut für Wissensbasierte Systeme, Stuttgart  
(*Görz, Studer, v. Luck, Novak*)

IBM Deutschland, Wissenschaftliches Zentrum Heidelberg (*Schönenfeld*)

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

Institut für Industrielle Fertigung und Fabrikbetrieb der Universität Stuttgart  
(*E.Kurz, Raether*)

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

---

### 3.1.5 Abteilung Kombinatorik und Numerische Verfahren

Leiter	<i>Knödel</i>
Mitarbeiter	<i>Berger, Ebinger (ab Juli), Eusterbrock, Hockel (bis April), Hruschka, Köbler (bis August), Mijderwijk (ab Oktober)</i>
Hilfskräfte	<i>Bertol, Eberhardt, Ebinger, Fritz, Geuder, Hözllein, Reinhardt, Schnakenburger, Teodosiu, Weber</i>

#### Forschungsprojekt: Algorithmensynthese: Integration deduktiver und induktiver Komponenten

( *Jutta Eusterbrock* )

Algorithmensynthese sei die Konstruktion einer Funktion  $alg$ , die die Spezifikation  $\forall P : prop_{in}(P) \Rightarrow prop_{out}(P, alg(P))$  beweist. Als Nebenbedingung an zu generierende Algorithmen seien Komplexitätsbedingungen  $cost(alg(P)) \leq f(P)$  zulässig. Allgemeine Methoden zur Konstruktion von Algorithmen können nicht existieren. Es ist ein formales Schema entwickelt worden, das die Synthese von korrekten Algorithmen und deren Komplexitätsanalyse auf die Konstruktion geeigneter Generalisierungen und Dekompositionsfunktionen zurückführt und die Einbeziehung partieller Algorithmdefinitionen erlaubt. Dual hierzu ist ein formales Gerüst zum Lernen geeigneter Funktionen und Konzepte aus Beispielberechnungen entwickelt worden. Ein solcher integrativer Ansatz ist bisher nicht untersucht worden. Dieses Modell wird im Synthesesystem ASK implementiert und in einer Fallstudie experimentell untersucht. Interaktiv konnte ein Gegenbeispiel zu einem veröffentlichten Satz generiert werden.

#### Forschungsprojekt: Zufallsorakel

( *Werner Ebinger* )

Eine Orakelmaschine ist eine Turingmaschine, die an eine Orakelmenge Fragen stellen kann, die „kostenlos“, ohne Verbrauch von Speicherplatz oder Rechenzeit, beantwortet werden. Die Rechnung der Maschine ist vom Orakel abhängig, sie rechnet „relativ“ zum Orakel. Man schreibt zum Beispiel  $P^A$  für die bekannte Problemklasse  $P$  mit Orakel  $A$ . Viele offene Fragen der Komplexitätstheorie lassen sich nun im relativierten Fall mit bestimmten Orakeln lösen. Aber diese Lösung ist vom speziellen Orakel abhängig.

---

So wurde 1975 von Baker, Gill und Solovay gezeigt, daß es Orakel  $A$  und  $B$  gibt, so daß  $P^A = NP^A$  aber  $P^B \neq NP^B$ . Diese Orakel wurden speziell konstruiert, um die jeweilige Aussage zu erhalten.

Nimmt man jedoch ein zufälliges, ohne irgendwelche Absichten gewähltes Orakel, so scheint der Zusammenhang mit dem unrelativierten Fall eher gegeben. Gilt eine Aussage für ein Zufallsorakel mit großer Wahrscheinlichkeit, so liegt es nahe anzunehmen, daß sie auch für den unrelativierten Fall gilt. So gilt zum Beispiel mit Wahrscheinlichkeit 1 für zufällige Orakel  $C$ , daß  $P^C \neq NP^C$ . Ein weiteres Berechnungsmodell, bei dem der Zufall eine Rolle spielt, sind probabilistische Turingmaschinen. Es ergibt sich ein Zusammenhang zwischen  $P$  mit zufälligem Orakel und der unrelativierten probabilistischen Klasse BPP.

## Forschungsprojekt: Complexity Classes with Advice

( *Johannes Köbler, Thomas Thierauf* )

Es wird ein neues Konzept nichtuniformer Komplexitätsklassen eingeführt. Damit lassen sich in einheitlicher Weise sowohl neue als auch bekannte Komplexitätsklassen beschreiben. Wir modifizieren hierzu die von Karp und Lipton eingeführte Definition in zwei Richtungen: Als erstes betrachten wir nicht nur Advice-Funktionen, die durch eine reine Längenbeschränkung definiert sind, sondern auch solche, die durch strukturelle Komplexitätseigenschaften charakterisiert sind. Diese werden dann zu Advice-Funktionen erweitert, die von der Eingabe abhängen und nicht nur von der Eingabellänge.

Damit charakterisieren wir folgende Sprachklassen durch  $NP$  mit dem Advice von gewissen Optimierungsfunktionen aus  $OptP$  bzw. von  $\chi_n^{SAT}$ , der  $n$ -stelligen charakteristischen Funktion von  $SAT$

- $NP$  mit dünnen  $NP$  Orakeln, d.h.  $NP^{NP \cap SPARSE}$ ,
- $NP$  mit  $NP$  Orakeln, wobei nur eine polynomiale Anzahl von Orakelfragen zugelassen ist,
- die ungeraden Stufen der Booleschen Hierarchie.

Hieraus ergibt sich, daß jede im Sinne von Rich und Book, Tang auf  $SAT$  nichtdeterministisch truth table reduzierbare Sprache bereits deterministisch auf  $SAT$  truth table reduziert werden kann.

## Eigenwertseparationsproblem für Matrizenpolynome

( *Sven Mijderwijk* )

Seit Nov. 1989 befasse ich mich mit der Einarbeitung in ein Gebiet der Operatorentheorie, welches die Methoden der Symmetrischen und Hermiteschen Formen zur Eigenwertseparation für algebraische Gleichungen für die Lösung dieses Problems für Matrizenpolynome verwendet. Dabei werden geeignete Verallgemeinerungen des Bezoutiantenbegriffes vorgenommen und zum skalaren Fall analoge Sätze bewiesen. Ziel meiner Arbeit ist es, Algorithmen für obenstehendes Problem anzugeben, die in starkem Maße berücksichtigen, daß die betrachteten algebraischen Gleichungen Determinanten von Matrizenpolynomen sind. Die Arbeit wird durch Herrn Heinig an der Karl-Marx-Universität Leipzig betreut.

## Echtzeitdatenverarbeitung auf dem PC

( *Uwe Berger* )

Dieses Forschungsprojekt wird in Zusammenarbeit mit der Abteilung Betriebsssoftware durchgeführt und ist in Abschnitt 3.1.1, Seite 45, beschrieben.

## Forschungskontakte

Institut für Informatik der Universität Innsbruck (*Albrecht*)

Karl-Marx-Universität Leipzig (*G. Heinig*)

Universität Barcelona (*Jacobo Torán*)

Universität Ulm (*Johannes Köbler, Uwe Schöning*)

Universität Tokyo (*Seinosuke Toda*)

Universität Würzburg (*Klaus W. Wagner*)

TH Darmstadt, Fachgebiet Intellektik

### 3.1.6 Abteilung Programmiersprachen und ihre Übersetzer

Leiter	<i>Barth (beurlaubt bis 30.9.)</i> <i>Göttler (Lehrstuhlvertretung ab 3.4.)</i>
Mitarbeiter	<i>Günthör, Bräunl, Dengel (bis 31.3.), Kübler, Schusser, Schwinn, Welsch, Zell</i>
Hilfskräfte	<i>Bögeholz, Huber, Koch</i>

### Projekt: Sprachkonzepte für die Programmierung von massiv parallelen Rechnerarchitekturen

( Thomas Bräunl )

Konventionelle Programmiersprachen beziehen sich implizit auf ein von–Neumannsches Rechnermodell mit nur einem Prozessor. Um Programme für moderne Architekturen mit vielen Prozessoren erstellen zu können, benötigt man eine übersichtliche und sichere Programmierumgebung, die es erlaubt, parallele Verarbeitungsschritte explizit zu formulieren. Die Effizienz eines Programms ist jedoch weitgehend davon abhängig, wie gut das Rechnermodell einer Programmiersprache zu der tatsächlich vorhandenen Rechnerarchitektur paßt.

Ziel des Forschungsprojektes ist die Entwicklung einer flexiblen parallelen Programmiersprache, die unterschiedliche Programmierparadigmen integriert, und welche sowohl die verwendete Rechnerarchitektur (Hardware) als auch den Algorithmus zur Problemlösung (Software) spezifiziert.

### Projekt: Simulatoren für neuronale Netzwerke

( Andreas Zell )

Neuronale Netze sind Modelle für informationsverarbeitende Systeme, die sich aus vielen einfachen, parallel arbeitenden Einheiten zusammensetzen, welche Information in Form ihrer Aktivierung über ein Netz von Verbindungen austauschen. Sie sind interessant, weil sie die Hardware–Struktur des Gehirns in einem vereinfachten Modell nachzubilden versuchen.

Im Rahmen mehrerer studentischer Arbeiten wurde NetSim, ein Simulator für neuronale Netze, entwickelt. Dieser Simulator besteht aus 3 Komponenten:

---

- Simulatorkern,
- graphische Benutzeroberfläche und
- Netzwerk–Compiler.

Der Simulatorkern verwaltet die interne Repräsentation der neuronalen Netze und führt alle Operationen für die Simulation in der Arbeits- und der Lernphase durch. Die graphische Benutzerschnittstelle stellt die Topologie und den aktuellen Zustand des Netzwerks graphisch dar. Sie ermöglicht es auch, kleine und mittlere neuronale Netze zu konstruieren und interaktiv zu ändern. Eine mengenorientierte, deklarative Beschreibungssprache, NeSiLa (Network Simulation Language) erlaubt die Beschreibung komplexer neuronaler Netze auf eine einfache Weise. Der Netzwerk–Compiler für NeSiLa dient zur Generierung von größeren Netzen aus einer deklarativen Beschreibungssprache.

Abbildung 3.1: Struktur des Netzwerk–Simulators NetSim

Der Kern des Simulators führt die Propagierung der Zellaktivierungen durch und das Lernverfahren. Die Definition der Lernregeln und der Propagierungsfunktion ist Teil der Netzwerkbeschreibung. Netze können während der Simulation durch die graphische Benutzerschnittstelle modifiziert werden.

Die graphische Benutzeroberfläche wurde auf SUN 3 Workstations unter SunView in C implementiert. Sie wird im Rahmen weiterer Arbeiten derzeit noch erweitert und unter X-Windows reimplementiert.

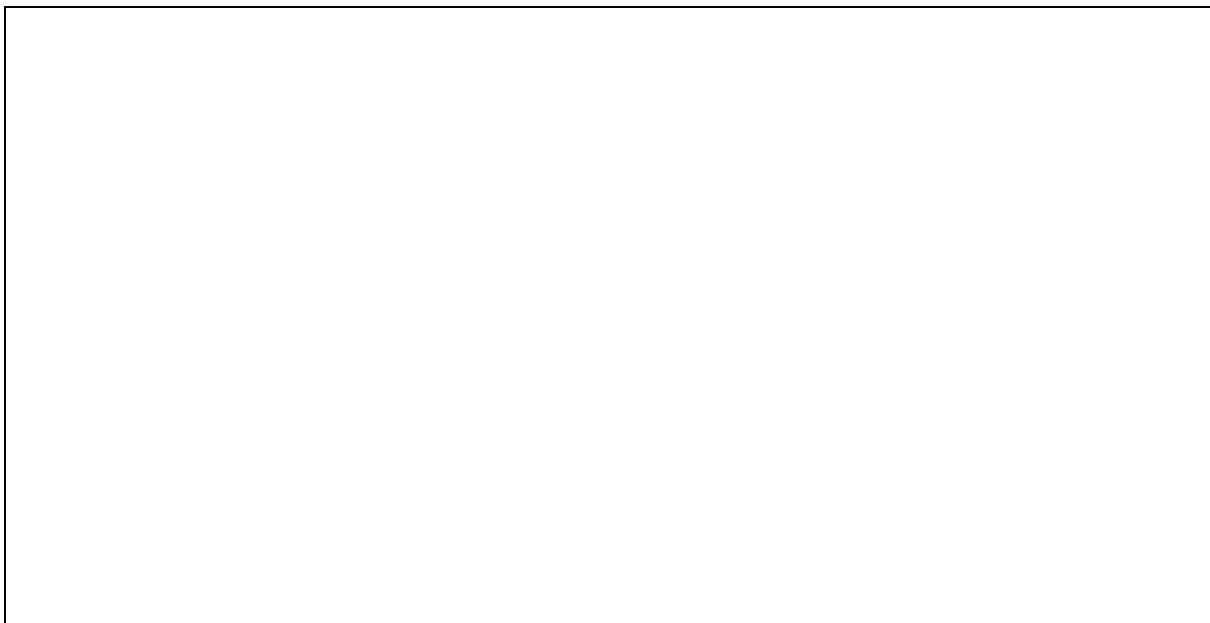


Abbildung 3.2: Graphische Benutzeroberfläche des Netzwerksimulators NetSim

## **Projekt: Metaebenen–Architekturen logischer Programme**

( *Andreas Zell* )

Ein Computerprogramm besitzt eine Metaebenen–Architektur, wenn es in zwei oder mehreren Ebenen strukturiert ist, wobei eine übergeordnete Ebene (Metaebene) den Programmablauf oder die Herleitungen der untergeordneten Ebene (Basisebene, Objektebene) steuert. Bei der Teilklasse der sogenannten introspektiven Architekturen besteht eine noch stärkere, bidirektionale Verbindung zwischen Metaebene und Objektebene. Introspektive Architekturen können damit ihr eigenes Problemlöseverhalten analysieren und modifizieren.

In diesem Projekt wurde zunächst die Notwendigkeit von Metaebenen–Architekturen anhand verschiedener Funktionen von Computersystemen und menschlicher introspektiver Aktivitäten motiviert, genauere Definitionen der Begriffe angegeben und die wichtigsten Konzepte dargestellt. Im folgenden wurden existierende Metaebenen–Architekturen beschrieben und nach der zugrundeliegenden Wissensrepräsentationsmethode klassifiziert.

---

---

Den eigentlichen Schwerpunkt des Projekts bildeten logische Metaebenen–Architekturen. Ausgehend von Konzepten und einfachen Metainterpretern wurden Iterative–Deepening Prolog und zeitbeschränktes logisches Programmieren theoretisch und praktisch untersucht. Auch Parallelität läßt sich in einer Metaebenen–Architektur einfach beschreiben, wie an einem parallelen Lisp–Interpreter in Concurrent Prolog gezeigt wurde.

Diese Fähigkeit der Reflexion über das eigene Schließen steht in engem Zusammenhang mit der Fähigkeit, sinnvolle Schlußfolgerungen über das Wissen und das daraus resultierende Verhalten anderer intelligenter Agenten (Personen oder Systeme) durchführen zu können. Dafür wurden zuerst die Konzepte auf mehrere Agenten erweitert, danach theoretische Ergebnisse des Schließens über das Wissen anderer hergeleitet und eine Implementierung angegeben.

Schließlich wurde ein Ansatz beschrieben, der Handlungen, Sensoren und Effektoren in das Gesamtkonzept integriert und als logische Basis für teilautonome Systeme dienen kann.

Die Arbeiten über Metaebenen–Architekturen sind in der gleichnamigen Dissertation über dieses Thema zusammengefaßt.

## Projekt: Das juristische Literaturarchiv JULIA

( *Herbert Göttler* )

JULIA ist ein Programm zur komfortablen computergestützten Literaturverwaltung. Bei der Konzeption stand eine weitverbreitete, „klassische“ Methode der Textarchivierung Pate: die Hängeordner–Registratur. Dieses einfache Modell ermöglicht die anschauliche Abbildung der Systemfunktionen von JULIA: Abzulegende Texte — entweder aus (beliebigen) Dateien übernommen oder mit dem integrierten Text-Editor erstellt — werden jeweils in einen „Hängeordner“ übernommen. Zur Charakterisierung des Inhalts dieser Hängeordner werden ihnen vom Benutzer frei beschriftbare „Reiter“ aufgesteckt. Bei der Suche im Testarchiv gibt der Benutzer dem System beliebige, als „Suchprofil“ kombinierbare Reiterbegriffe vor. JULIA ermittelt in Sekundenschnelle die zum jeweiligen Suchprofil passenden Texte und bietet sie dem Benutzer zur Weiterverarbeitung an.

Besonderer Wert wurde auf die sinnvolle Verwaltung der eingegebenen Reiterbegriffe gelegt. Diese werden zunächst in einem vom System automatisch fortgeschriebenen, alphabetisch geordneten Wörterbuch festgehalten. Darüber hinaus ist es möglich, die Reiterbegriffe hierarchisch in Über– und Unterbegriffe zu ordnen, was das Auffinden treffender Suchbegriffe sowohl bei der Textsuche als auch bei der Texteingabe erheblich vereinfacht. JULIA bietet eine leicht verständliche Bedienoberfläche, in der alle Systemfunktionen menügesteuert auf Tastdruck verfügbar sind. Die vorgestellte Version ist speziell entwickelt für den Einsatz in Rechtsanwaltskanzleien, beispiels-

---

weise zur Verwaltung umfangreicher Literaturarchive oder Leitsatzkarteien. Der modulare Systemaufbau und das universelle Konzept von JULIA bietet jedoch vielfältige Möglichkeiten in allen Bereichen, in denen es auf effiziente Archivierung von Texten aller Art unter einfachsten Bedingungen ankommt.

## **Projekt: Methodische Implementierung graphischer Software–Entwicklungs werkzeuge**

( *Herbert Göttler* )

In allen ingenieurwissenschaftlichen Disziplinen, insbesondere auch in der Informatik, werden Diagramme zur Darstellung von Sachverhalten verwendet. Um den oft beträchtlichen zeitlichen Aufwand zum Zeichnen zu reduzieren, braucht man Rechnerunterstützung. Dabei ist es nicht unbedingt wünschenswert, eines der vielen universellen Zeichenprogramme einzusetzen um Diagramme mit speziellen Eigenschaften zu produzieren. Da man „alles“ mit ihnen zeichnen kann, ermöglichen sie es auch, „falsche“ (in syntaktischem Sinn) Diagramme zu entwerfen. Es ist deshalb besser, für eine Diagrammkategorie ein ad hoc Werkzeug, einen syntaxgesteuerten Diagrammeditor, zur Verfügung zu haben, der Fehler verhindert. Man kann schnell solche Graphikeditoren erstellen, wenn man attributierte Graphgrammatiken einsetzt. Sie erlauben die syntaktischen und semantischen Eigenschaften von Diagrammen zu beschreiben. Der genannte Kalkül läßt sich auch bei Anwendungen im CAD–Bereich einsetzen.

## **Projekt: RAPiD: Ein Datenflußmodell zur Realisierung von Parallelität und intelligentes Backtracking in logischen Programmen**

( *Bernd Schwinn* )

Die logische Programmierung hat in den letzten Jahren vor allem im Bereich der Wissensverarbeitung stark an Bedeutung gewonnen. Diese Tendenz setzt sich fort, obwohl Prolog, die zur Zeit bekannteste logische Programmiersprache, auf einer relativ ineffizienten Herleitungsstrategie basiert. Die Hauptursache dafür ist, daß die bisherigen Implementierungen auf Architekturen erfolgen, die einige wichtige Eigenschaften der logischen Programmierung, wie Parallelisierbarkeit und Flexibilität, nicht geeignet ausnutzen können. In diesem Projekt wurden die Möglichkeiten für Effizienzsteigerungen durch besseres Ausnutzen der in logischen Programmen enthaltenen Nichtdeterminismen aufgezeigt. Dabei wurden die wichtigsten Realisierungsansätze für UND– und ODER–Parallelität sowie für intelligentes Backtracking miteinander verglichen. Als Ergebnis wurde ein Datenflußmodell (RAPiD) entwickelt, das die Realisierung von Parallelität und intelligentem Backtracking ermöglicht. Das Modell basiert hauptsächlich

---

---

auf einer Abhängigkeitsanalyse nach einem Ansatz von Doug DeGroot (Restricted AND–Parallelism). Dabei wird zunächst eine statische Klassifikation von möglichen Abhängigkeitssituationen unter den Teilzielen der Klauseln durchgeführt. Diese Klassifikation führt zur Darstellung einer jeden Klausel in Form eines eigenständigen Datenflußgraphen. Die Graphen enthalten einfache Test–Operationen, mit denen zur Ausführungszeit die Abhängigkeit von Teilzielen festgestellt werden kann. Diese Art der Abhängigkeitsanalyse wird auch zur Organisation des intelligenten Backtracking benutzt. Die ODER–Parallelität kann, wie beim AND/OR Process Model von Conery, so realisiert werden, daß parallel nach alternativen Lösungen für Teilziele gesucht wird, jedoch nur mit dem zuerst vorliegenden Ergebnis weitergearbeitet wird.

## **Projekt: Integration von logischer und objektorientierter Programmierung**

( *Christoph Welsch* )

Grundlage der objektorientierten Programmierung ist die Idee, einen vorliegenden Problembereich als Kollektion aktiver Einheiten, sogenannter Objekte zu sehen. Objekte repräsentieren Problemkomponenten. Sie können über Nachrichten miteinander kommunizieren und fordern sich dadurch gegenseitig zur Durchführung von Operationen auf.

Ein typisches Merkmal der objektorientierten Programmierung ist das Prinzip der Vererbung. Es gestattet, Gemeinsamkeiten von Objekten zentral zu beschreiben, und trägt so dazu bei, daß objektorientierte Systeme schnell implementiert und leicht gewartet werden können. Die objektorientierte Programmierung gilt heute als einer der größten Hoffnungsträger zur Überwindung der Software–Krise.

Ziel dieses Projektes ist es, die Vorteile der objektorientierten Programmierung (Modularität, Wartbarkeit, Abkapselung, Vererbung) auf die logische Programmierung zu übertragen. Die Punkte, durch die sich die logische Programmierung gegenüber anderen Programmiermethoden auszeichnet, sind zum einen ihr solides theoretisches Fundament, zum anderen die Abstraktion vom Kontrollfluß, die einen deklarativen Sprachstil fördert.

In der ersten Phase des Projekts wurde die logische Programmiersprache Prolog um ein Modulkonzept erweitert. Ein logisches Programm wird dabei als Kollektion kleinerer Wissenseinheiten (Objekte) aufgefaßt, die sich über Nachrichten wechselseitig konsultieren. Das Wissen eines Objekts ist durch eine Menge von Hornklauseln beschrieben.

In der zweiten Phase wurde ein Vererbungskonzept für derartige Objekte konzipiert. Spezielle Terme erlauben es, Klassen als Objektmengen darzustellen. Diese Terme sind hierarchisch in einem Vererbungsverband angeordnet. Dank der Möglichkeit, ihnen

---

Klauselmengen zuzuordnen, überträgt sich die Vererbungsstruktur der Terme auf die Wissensbasen von Objekten. Wissen, das in allgemeineren Klassen enthalten ist, vererbt sich automatisch auf speziellere Klassen und braucht nicht dupliziert zu werden. Ähnlich wie in rein objektorientierten Systemen unterstützt diese Form der Vererbung die Wiederverwendung früher beschriebenen Wissens.

Zur Zeit wird in der dritten und letzten Projektphase ein Modell zur Zustandsänderung von Objekten in das bestehende System integriert. Die Änderung von Objektzuständen wird dabei durch Modifikation der zugehörigen Wissensbasen realisiert. Die Regeln, denen diese Modifikationen zu gehorchen haben, sind innerhalb eines Objekts wiederum durch Hornklauseln beschrieben.

## **Forschungskontakte**

Stanford University (*M. R. Genesereth, N. Nilsson, M. L. Ginsberg*)

Deutsches Forschungszentrum für Künstliche Intelligenz

(*G. Barth, A. Dengel, R. Bleisinger, M. Sommer, J. Müller*)

IBM Deutschland GmbH, Bereich Wissenschaft, Abt. LILOG

(*U. Pletat, Ch. Beierle*)

Weizman Institute of Science, Rehovot, Israel (*E. Shapiro, B. Silverman*)

Swedish Institute of Computer Science (*M. Carlsson*)

GMD Gesellschaft für Mathematik und Datenverarbeitung (*J. Diederich, A. Beer*)

University of Rochester (*L. Buksy, P. Meeker*)

Siemens AG (*Th. Bareiß*)

BASF AG (*P. Herrmann, H. Fischer, G. Klett*)

Daimler Benz AG (*Th. Rapp*)

Österreichische Gesellschaft für Artificial Intelligence (*J. Retti*)

Tampere University of Technology (*H. Jaakola*)

Politecnico di Milano (*L. Mezzalira*)

California Institute of Technology (*Ch. L. Seitz*)

Free University Brussels, AI-Lab (*P. Maes*)

University of Southern California (*F. Arbab, R. Weinberg*)

### 3.1.7 Abteilung Software Engineering

Leiter	<i>J. Ludewig</i>
Wiss. Mitarbeiter	<i>H. Lichter, M. Deininger (ab 1.3.), K. Schneider (ab 1.8.)</i>
Programmierer	<i>M. Schneider (ab 1.10.)</i>
Verwaltungsangestellte	<i>Frau U. Günthör</i>

### Projekt: Objektorientierte Software–Entwicklung und Prototyping

Ziel dieser (schon früher begonnenen) 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).

### Projekt: GOSE

Im Rahmen dieses neuen Forschungsprojekts wird ein Simulationspaket in Form eines Computerspiels (GOSE = Game of Software Engineering) 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.

Mit den Vorarbeiten wurde begonnen, am Jahresende lief ein — extrem primitiver — Prototyp.

Wesentliche Merkmale und Schwerpunkte dieses Projekts sind

- die Sammlung und Zusammenstellung der heute bekannten oder vermuteten Gesetzmäßigkeiten im Software Engineering
- die Zusammenarbeit mit Leuten außerhalb des Projekts, die praktische Erfahrungen einbringen können
- Projekttraining für alle Mitarbeiter durch die Arbeit an GOSE
- Bereitstellung eines Mittels für die Ausbildung (die Arbeit mit GOSE)

## Forschungskontakte

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

ETH Zürich (*N. Wirth, H. Matheis*)

IBM Yorktown Heights (*D. Gangopadhyay u.a.*)

SIEMENS Corporate Research, Princeton, NJ (*T. Ostrand u.a.*)

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

### 3.1.8 Abteilung Theorie der Informatik

Leiter	<i>Knödel (kommissarisch bis 30.9.), Diekert (Lehrstuhlvertretung ab 1.10.)</i>
Mitarbeiter	<i>Münchow (ab 1.10.), Reinhardt (ab 1.10.), Reuß, Thierauf (bis 31.8.)</i>
Hilfskräfte	<i>Benzinger, Dörre, Ebinger, Faulstich, Hölzlein, Huppenbauer, Ketelhut, Nguyen, Räumschlüssel, Reinhardt, Schumacher, Schedel, Semle, Wahl, V.Zink</i>

#### Forschungsprojekt: Freie partiell kommutative Monoide

( Volker Diekert )

Freie partiell kommutative Monoide sind in den letzten Jahren zu einem wichtigen Forschungsgegenstand der theoretischen Informatik geworden. Sie dienen hier als ein abstraktes Modell zur Beschreibung nebenläufiger Prozesse. Untersucht man Transformationen nebenläufiger Prozesse, so führt dies auf das Studium von Ersetzungssystemen über diesen Monoiden. Diese Ersetzungssysteme verallgemeinern das klassische Konzept von Semi–Thue Systemen und Vektorersetzungssystemen (Petrinetzen). Es treten jedoch völlig neue Phänomene auf, die insbesondere die Konfluenz noetherscher Systeme und die Komplexität des Wortproblems betreffen. Unter Verallgemeinerung eines Ansatzes von Winkler–Buchberger für Termersetzungssysteme konnte ein neues entscheidbares Kriterium entwickelt werden, das hinreichend ist, die Konfluenz zu entscheiden. Es werden ebenfalls Kriterien entwickelt, die eine effiziente Lösung des Wortproblems erlauben. Die Lösung des Wortproblems basiert hierbei auf der Berechnung irreduzibler Nachfolger für noethersche Systeme. Hier konnte ein einheitlicher Algorithmus gefunden werden, der sich nach heutigem Wissen bis auf Konstante zeit-optimal verhält. Dieser Algorithmus basiert auf einem Konzept asynchroner Automaten, das von W. Zielonka entwickelt wurde. Eine Implementierung einiger der zugrunde liegenden Algorithmen, die in Vorarbeiten entwickelt wurden, fand im Rahmen von Diplomarbeiten an der TU–München bereits statt. Die oben erwähnten Forschungsergebnisse werden beim 7th Annual Symposium on Theoretical Aspects of Computer Science (STACS 90) in Rouen (Frankreich) vorgestellt.

#### Forschungsprojekt: Charakterisierung von Komplexitätsklassen mit formalsprachlichen Methoden

( Klaus Reinhardt )

Eine interessante Fragestellung bezüglich eines Automatentypen bzw. einer Zeit– oder Platzbeschränkung ist, ob ein nichtdeterministischer Automat dieser Art mehr leisten

kann als ein deterministischer Automat. Bei den Automaten für Typ 0– und Typ 3–Sprachen konnte gezeigt werden, daß dies nicht der Fall ist, bei Kellerautomaten konnte gezeigt werden, daß eine echte Inklusion vorliegt, bei Turingmaschinen mit Komplexitätsbeschränkung sind dies jedoch ungelöste Probleme. Aus diesem Grund ist es interessant, Komplexitätsklassen mit formalsprachlichen Methoden zu charakterisieren.

Für Sprachen  $L, L'$  gilt  $L \leq^{\log} L'$ , wenn es einen logarithmischen Transducer  $T$  mit  $x \in L$  gdw.  $f_T(x) \in L'$  gibt. Für eine Sprachklasse  $S$  ist die logarithmische Hülle  $LOG(S) := \{Y \mid \exists X \in S \text{ mit } Y \leq^{\log} X\}$ .

Seit längerem ist bekannt, daß  $NSPACE(\log(n)) = LOG(LIN)$  ( $LIN$  ist die Klasse der linearen kontextfreien Sprachen), daß  $LOG(CFL)$  gleich der Klasse der Sprachen ist, die von uniformen Schaltkreisen polynomieller Größe und logarithmischer Tiefe mit halbgebundenem Fan-In erkannt werden können, und daß  $LOG(DCFL)$  gleich der Klasse der Sprachen ist, die von einem  $CROW - PRAM$  mit polynomiell vielen Prozessoren in logarithmischer Zeit erkannt werden können.

Als neues Resultat konnte gezeigt werden, daß die logarithmische Hülle über jeder Stufe der Hierarchie mit alternierenden Einweg–Kellerautomaten die jeweilige Stufe der polynomiellen Hierarchie mit einer Alternierung weniger ergibt. Für die logarithmischen Hüllen über Stufen einer Orakelhierarchie über den kontextfreien Sprachen bzw. einer Hierarchie mit alternierenden kontextfreien Grammatiken gilt dies ebenfalls.

## Forschungsprojekt: Leistungsgebundene Restriktionsanalyse von Transputer–System–Topologien

( Joachim Münchow )

Die Leistungsfähigkeit eines Multiprozessor–Systems hängt stark vom verwendeten Kommunikationskonzept und von der realisierten Topologie ab. Während ersteres bei Transputern durch den Hersteller weitgehend vorgegeben ist, die Kommunikation erfolgt über ein integriertes Memory–Interface und vier nach dem Kanalkonzept arbeitende *Links*, kann die Topologie vom Anwender durch „Stecken“ von Verbindungen, was über sogenannte Konfigurationsbausteine auch dynamisch geschehen kann, frei gewählt werden.

Da die Leistungsfähigkeit einer Topologie stark von der gegebenen Problemstellung abhängt, zahlreiche Probleme führen auf Standardtopologien (Ring, Stern, Kubus ...), andere gar auf irreguläre Verbindungsstrukturen, ist es schwierig, ein Maß dafür anzugeben.

Untersucht wurden daher zunächst Topologien, die *allgemeinen* Anwendungen (identische Kommunikationswahrscheinlichkeiten) gerecht werden. Eine hier angesetzte leistungsgebundene Restriktionsanalyse bezüglich der zur Verfügung stehenden Hard-

---

ware–Resourcen führte auf eine Zerlegung in Topologie–Klassen. Es konnte gezeigt werden, daß bestimmte Repräsentanten dieser Klassen hinsichtlich ihrer Leistungsfähigkeit unter der gegebenen Restriktion als optimal angenommen werden dürfen. Be merkenswerterweise stellt sich heraus, daß dies bei geeigneter Prozeßzuordnung auch für beliebige spezielle Anwendungen gilt.

## **Forschungsprojekt: Complexity Classes with Advice**

( *Thomas Thierauf* )

Dieses Forschungsprojekt wird in Zusammenarbeit mit der Abteilung Kombinatorik und Numerische Verfahren durchgeführt und ist in Abschnitt 3.1.5, Seite 71, beschrieben.

## **Forschungkontakte**

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

Université de Bordeaux I, Frankreich (*R. Coré, Y. Metivier*)

Universität Erlangen, BRD (*R. König*)

Universität GH Kassel, BRD (*F. Otto*)

Polnische Akademie der Wissenschaften in Warschau, Polen (*E. Odman ski, W. Zielonka*)

Universität Ulm, BRD (*Th. Thierauf*)

Universität Würzburg, BRD (*G. Buntrock*)

Technische Universität München, BRD (*K.J. Lange*)

## 3.2 Veröffentlichungen

- Bauer, D.** 1) *Einsatz einer anwendungsneutralen Benutzerschnittstelle in einer Büroanwendung als Beispiel für wissensbasierte Mensch–Computer–Kommunikation*  
*Angewandte Informatik, S.294–302, 31(7), 1989*
- Böcker, H.-D.**
- Gunzenhäuser, R.**
- v.d. Herberg, H.**
- Maier, D.**
- Rathke, M.**
- Ressel, M.**
- Schwab, Th.** 2) *Unterstützende Komponenten für wissensbasierte Mensch–Computer–Kommunikation*  
*Angewandte Informatik, S.351–359, 31(8), 1989*
- Böcker, H.-D.**
- Fischer, G.**
- Nieper-Lemke, H.** *The Role of Visual Representations in Understanding Software*  
Patridge, D. (Hrsg.): *AI and Software Engineering*  
Ablex Publishing Corporation, Norwood, NJ,  
1989 (im Druck)
- Böcker, H.-D.**
- Herczeg, J.**
- Herczeg, M.** *ELAB — An Electronics Laboratory*  
*Proceedings of the Fourth Conference on Artificial Intelligence and Education,*  
IOS, Amsterdam, 1989
- Bodenschatz, W.** 1) *Multi–Transputer–Reduktionsmaschine*  
Tagungsband des gemeinsamen Workshops der Fachgruppen 2.1.4 und 3.1.2 der GI ‘Sprachen, Algorithmen und Architekturen für Parallelrechner’, S.150–181, Bad Honnef.
- 2) *Leistungsverhalten einer Multi–Transputer–Maschine*  
zur parallelen Reduktion von Funktionsprachenspro-grammen  
Tagungsband des GI/PARS Workshops ‘Konfiguration, Benutzung und Programmierung von Parallelrechnern’, Mitteilung-Nr. 6, München, April 1989
- Bräunl, T.** 1) *Parallaxis: A Flexible Parallel Programming Environment for AI Applications*  
*Applications of Artificial Intelligence VII, Orlando, Florida, März 1989, pp. 275 (11)*
-

- 
- 2) *A Specification Language for Parallel Architectures and Algorithms*  
*Fifth International Workshop Software Specification and Design,*  
*Pittsburgh, Pennsylvania, Mai 1989, pp. (3)*
- 3) *Structured SIMD Programming in Parallaxis*  
*Structured Programming, vol. 10, no. 3,*  
*Juli 1989, pp. (12)*
- Bräunl, Th.**                    1) *Iterative–Deepening Prolog*  
**Zell, A.**                    *Proceedings of the Second Scandinavian Conference on Artificial Intelligence,*  
                                  *Tampere, Finnland, Juni 1989, pp. 1041–1051*
- 2) *An Alternative Prolog Search Strategy*  
*Proceedings 2nd International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems,*  
*Tullahoma, Tennessee, USA, 6.–9. Juni 1989*
- Burkhardt, W. H.**            1) *Development of a PASCAL–System for Parallel Compilation*  
                                  *Proceedings, International Phoenix Conference on Computers and Communications,*  
                                  *Phoenix, Arizona, März 1989*
- 2) *Performance Degradation by Input/Output in Multiprocessor Systems*  
*Proceedings, 32nd Midwest Symposium on Circuits and Systems,*  
*Urbana, Illinois, August 1989*
- 3) *Locality Aspects and Cache Memory Utility in Microcomputers*  
*Microprocessing and Microprogramming,*  
*(North-Holland), Band 26 (1989), 51–62*
- Burkhardt, W. H.**            *User Protection in a Workstation Network*  
**Hoffnung, H.**                *Angewandte Informatik, (Vieweg), (5/1989), 197–202*
-

- Fehrle, Th.** *Eine menüorientierte Klärungskomponente für ein natürlichsprachliches Aufkunftssystem  
in B. Endres-Niggemeyer et.al.,  
GLDV-Jahrestagung 89, Ulm, März 89*
- Göttler, H.**
- 1) *Graphgrammars, a New Paradigm for Implementing Visual Languages*  
*Proceedings 3rd International Conference on „Rewriting Techniques and their Application“ (RTA-89), Chapel Hill, N.C., USA,*  
*Dershowitz (Ed.), Lecture Notes in Computer Science 355, Springer Verlag, New York, 1989, S.152–166*
  - 2) *Graphgrammars, a New Paradigm for Implementing Visual Languages*  
*Proceedings EUROGRAPHICS'89 (European Computer Graphics Conference and Exhibition) Hamaburg, W. Hansmann et al. (Eds.), North Holland, Amsterdam, 1989, S.505–516*
  - 3) *Graphgrammars, a New Paradigm for Implementing Visual Languages*  
*Proceedings 2nd European Software Engineering Conference (ESEC'89), Universsity of Warwick, Coventry, UK, Lecture Notes in Computer Science 387, Springer Verlag, New York, 1989, S.336–350*
  - 4) *Übersetzung von „Graph Grammars for Implementing Visual Languages“ ins Tschechische erscheint in MAA/Informational Systems, 1989, Prag*
- Göttler, H.**
- Günzel, U.**
- Himmelreich,B.**
- 1) *Das juristische Literaturarchiv JULIA*  
*Proceedings GI-Workshop „Informationstechnik am Arbeitsplatz von Juristen“, L. van Raden / M. Weihermüller (Eds.), Carl Heymann Verlag, Köln, 1989, S.99–105*
  - 2) *Das Juristische Literaturarchiv JULIA*  
*jur-PC, Heft 5–6/89, MediConsult Verlag, Wiesbaden, 1989, S.200–205*
-

- 3) *Das Juristische Literaturarchiv JULIA*  
*Proc. GI - 19. Jahrestagung (Computergestützter Arbeitsplatz),*  
*München, Oktober 1989, Informatik Fachberichte 223,*  
*Springer Verlag, Heidelberg, 1989, S.65–78*
- Gunzenhäuser, R.** *Tactile Representation of Scanned Documents*  
**Schweikhardt, W.** *Proceedings of the Workshop on Access to Visual Computer Information by Blind Persons*  
*Berlin, September 1989 (im Druck)*
- Heid, U.** *Collocations in Multilingual Generation*  
**Raab, S.** *Proceedings 4th Conference of the European Chapter of ACL,*  
*Manchester, U.K., April 1989*
- v.d. Herberg, H.** *Generating Help Information with a domain-independent Help System*  
**Rathke, M.** *Proceedings of the WOODMAN '89 Conference,*  
*Rennes, France, Mai 1989*
- Knopik, Th.** *AI-Methods for Structuring HYPERTEXT-Information*  
**Ryser, Sigrid** *Proceedings der HYPERTEXT II Konferenz,*  
*York, U.K., Juni 1989*
- Knopik, Th.** *Eine quantitative Methode zum Vergleich des Dialogverhaltens interaktiver Systeme*  
**Strothotte, Th.** *H.-J. Scheibl (Hrsg.)*  
*3. Kolloquium Proceedings des 3. Kolloquiums Software-Entwicklungs-Systeme und –Werkzeuge, S.13.4ff*  
*Technische Akademie Esslingen, September 1989*
- Köbler, J.** *On counting and approximation*  
**Schöning, U.** *In Acta Informatica, 26, S.363–379, 1989*  
**Torán, J.**
- Köbler, J.** *Turing machines with few accepting computations and low sets for PP*  
**Schöning, U.** *In Proceedings 4th Structure in Complexity Theory Conference, IEEE Computer Society Press, S.208–215,*  
**Toda, S.** *1989*  
**Torán, J.**

- Lehmann, E.** *Wissensrepräsentation  
Künstliche Intelligenz. (Luck, K.v., Hrsg.), Informatik  
Fachberichte 203, Springer Verlag, Berlin 1989,  
S.52–77*
- Ludewig, J.**
- 1) *Languages, methods, and tools for software specification  
in J. Zalewski, W. Ehrenberger: Hardware and Software  
for Real Time Process Control.  
North Holland, Amsterdam etc., 1989, 225–256*
  - 2) *CASE: Tools, Lies, and Video Games  
TECHNISCHE RUNDSCHAU 26/89, 50–55*
  - 3) *Modelle der Software-Entwicklung: Abbilder oder Vor-  
bilder?  
Softwaretechnik-Trends, 9, 3 (Okt. 1989), 1–12*
- Rathke, C.  
(mit Henninger, Ignatowsky, Redmiles)** *A Knowledge-Based Design Environment for Graphical  
Network Editors  
Proceedings 22nd Hawaii International Conference on  
System Sciences, Januar 1989*
- Rathke, C.** *Incremental Programming of Spreadsheet Models  
through Constraint Application  
Proceedings of the Al-Azhar Engineering First Conference,  
Vol. 17, pp. 91–104, Dezember 1989*
- Schwab, Th.** *Dialogmodellierung in interaktiven Systemen  
Maaß, S.; Oberquelle, H. (Hrsg.)  
Software Ergonomie '89, S.174–183  
Berichte des German Chapter of the ACM, Band 32,  
Teubner, 1989*
- Schwinn, B.** *Parallelität und intelligentes Backtracking in logischen  
Programmen  
Informationstechnik, Heft 4, August 89*
- Schwinn, B.  
Barth, G.  
Welsch, C.** *RAPiD — A Data Flow Model for Implementing Parallelism and Intelligent Backtracking in Logic Programs  
Proceedings PARLE'89, Lecture Notes in Computer  
Science, Nr. 366, S.115–132, 1989*

---

- Strothotte, Th.**      1) *Pictures Supporting Human–Computer Dialogs in Videotex Systems*  
*Proceedings of European Telecommunications Standards Institute Workshop on a Future Videotex System, Wien, Januar 1989*
- 2) *Pictures in Advice–Giving Dialog Systems: From Knowledge Representation to the User Interface*  
*Proceedings Graphics Interface '89, S.94–99, London, Ontario, Juni 1989*
- 3) *Aspects of Language and Pictorial Representations in Human–Computer Interaction*  
*Proceedings of the Workshop on Access to Visual Computer Information by Blind Persons, Berlin, September 1989 (im Druck)*
- Strothotte, Th.**      *Using Computational Geometry in Knowledge–Based Dialog Systems*  
**Nussbaum, D.**            Hübner, A.; Nagel, W.; Rippley, B.D.; Werner, G.  
**Sack, J.-R.**             (Hrsg.)  
*Mathematical Research, S.195–200, Akademie–Verlag, Berlin, 1989*
- Strothotte, Th.**      *Informationsvermittlung in interaktiven wissensbasierten Systemen durch bildhafte Darstellungen*  
**Böcke, D.**             Maaß, S.; Oberquelle, H. (Hrsg.)  
*Software Ergonomie '89, S.345–354*  
*Berichte des German Chapter of the ACM, Band 32, Teubner, 1989*
- Strothotte, Th.**      *A Note on the Construction of the Data Structure ‘Deap’*  
**Carlson, S.**            *Information Processing Letters 31(6), S.315–317, Juni 1989*
- Strothotte, Th.**      *A Note on the Construction of Min–Max Heaps*  
**Eriksson, P.**            *Scandinavian Journal on Numerical Analysis and Computing*  
**Vallner, S.**             *BIT 29, S.251–256, August 1989*

- Weber, G.** *Reading and pointing — modes of interaction for blind users*  
*Ritter, G.X. (Hrsg.)*  
*Information Processing 89*  
*Proceedings of 11th World Computer Congress,*  
*S.535–540, IFIP, 1989*
- Weber, G.** *Finger — formalizing interaction for gesture recognition*  
**Wetzel, P.** *SIGCHI Bulletin, S.96–97, ACM, Juli 1989*
- Welsch, C.** *A Model to Integrate Logic A Programming, Communicating Objects and State Change.*  
*Workshop Alternative Konzepte für Sprachen und Rechner, Fachberichte Informatik, 6 /89,*  
*EWH Koblenz, S.189–200, 1989*
- Welsch, Ch.** *Reasoning Objects with Dynamic Knowledge Bases*  
**Barth, G.** *Proceedings EPIA 89, Lecture Notes in Computer Science, Nr. 390, Springer–Verlag, S.257–268, 1989*
- Zell, A.** 1.) *Time Constrained Logic Programming*  
*Proceedings of the Fifth Israeli Symposium on Artificial Intelligence, Vision and Pattern Recognition,*  
*Dec. 27–28, 1988, Tel Aviv, Israel, pp. 301–314*
- 2.) *Zeitbeschränktes Logisches Programmieren*  
*Informatik Fachberichte 208, J. Retti, K. Leidlmair (Hrsgb.), 5. Österreichische Artificial Intelligence–Tagung, Igls, Tirol, März 1989,*  
*Springer–Verlag, S.96–105*
- Zell, A.** *A Declarative Neural Network Description Language*  
**Korb, Th.** *Microprocessing and Microprogramming, Vol. 27, No's 1–5, Proceedings EuroMicro 89, North Holland,*  
*Sept. 1989, pp. 181–185*
- Zell, A.** *NetSim: Ein Simulator für Neuronale Netze*  
**Korb, Th.** *Informatik Fachberichte 216, D. Metzing (Hrsg.)*  
**Sommer, T.** *GWAI–89, 13th German Workshop on Artificial Intelligence, Eringerfeld, Sept. 89, Springer–Verlag,*  
**Bayer, R.** *S.134–143*
-

### 3.3 Berichte

- Emele, M.** *Organizing Linguistic Knowledge for Multilingual Generation*  
**Heid, U.** *Internal paper: Stuttgart, IfI/AIS and IMS/CL:*  
**Kehl, W.** *Polygloss*  
**Momma, S.** *submitted to COLING-90*  
**Zajac, R.**
- Emele, M.** 1) *Typed Unification Grammars*  
**Zajac, R.** *Internal paper: Stuttgart, IfI/AIS and IMS/CL:*  
*Polygloss, November 1989*  
*submitted to COLING-90*
- 2) *Semantics for Feature Type Systems*  
*Internal paper: Stuttgart, IfI/AIS and IMS/CL:*  
*Polygloss, December 1989*  
*submitted to ACL-90*
- Heid, U.** *Syntactic Construction in (Machine) Translation Dictionaries*  
*Internal paper: Stuttgart, IfI/AIS and IMS/CL:*  
*Polygloss, December 1989*  
*submitted to the 5th International Symposium on Lexicography, 1990*
- Heid, U.** *Layered Lexicons for Generation*  
**Momma, S.** *Internal paper: Stuttgart, IfI/AIS und IMS/CL:*  
*Polygloss*
- Herczeg, J.** *The HyperQuery System*  
**Schwab, Th.** *Technical Report,*  
*Institut für Informatik, Universität Stuttgart, 1989*
- v.d. Herberg, H.** *Standardisierung des INFORM Benutzerschnittstellen-*  
**Herczeg, J.** *Baukastens*  
**Holz, D.** *WISOM-Arbeitsbericht AB-INF-89-01*  
*Forschungsgruppe INFORM, Universität Stuttgart*  
*März 1989*
- Schwinn, B.** 1) *Compilerbau*  
**Barth, G.** *Vorlesungsmanuskript, WS 88/89*
- 2) *Konzepte höherer Programmiersprachen*  
*Vorlesungsmanuskript, WS 89/90*

**Schwinn, B.**  
**Welsch, C.**  
**Barth, G.**

*Nichtprozedurale Programmierung,  
Vorlesungsmanuskript, SS 89*

**Zajac, R.**

*A Relational Approach to Translation  
Internal paper: Stuttgart, IfI/AIS and IMS/CL:  
Polygloss, December 1989  
submitted to Third International Conference on Theo-  
retical and Methodological Issues of Machine Translati-  
on of Languages (Austin 1990)*

## 3.4 Vorträge

**Berger, U.**  
**Eggenberger, O.**

*Multitasking–Betriebssysteme für Realzeitanwendungen  
auf AT-kompatiblen Personal Computern  
PEARL 89 — Workshop über Realzeitsysteme  
Boppard  
Dezember 1989*

**Böcker, H.-D.**

- 1) *Programmvisualisierung durch visuelle Programmierung  
Institut für integrierte Publikations- und Informationssysteme  
Darmstadt  
Dezember 1989*
- 2) *Werkzeuge zur Visualisierung dynamischer Aspekte von Programmen  
3. Workshop „Programmierwissen und seine Modellierung  
Trier  
September 1989*
- 3) *Wissensbasierte Designumgebungen als intelligente Lernsysteme  
3. Workshop der Fachgruppe Intelligente Lernsysteme  
Deutsches Institut für Fernstudien an der Universität Tübingen  
Juni 1989*
- 4) *ELAB — An Electronics Laboratory  
Fourth Conference on Artificial Intelligence and Education  
Amsterdam  
Mai 1989*
- 5) *Wissensbasierte Mensch–Computer Kommunikation —  
Mehr als ein Schlagwort?  
Institut für Informatik  
Universität Basel  
Mai 1989*

- 6) *Programmierumgebungen und Visualisierungstechniken*  
*Technische Akademie Elllingen*  
*Elllingen*  
*Mai 1989*
  
- 7) *On the Role of Knowledge Representation in Human-Machine Communication — Two Examples*  
*Department of Computer Science*  
*Boulder, Colorado*  
*März 1989*

**Bodenschatz, W.**      *Multi-Tansputersystem zur parallelen Reduktion von  
Funktionsprachenprogrammen*  
*6. Treffen der Deutschen OCCAM-Interessengemeinschaft der Transputeranwender (DOIT)*  
*Witten*  
*17./18. Oktober 1989*

**Bräunl, T.**      *Implementierung massiv paralleler Algorithmen mit  
Parallaxis*  
*IVS-Workshop „Parallele Algorithmen“*  
*Stuttgart*  
*8.12.1989*

**Burkhardt, W. H.**      *„Compilation or Interpretation“*  
*Computer Science Conference '89*  
*Louisville, Kentucky*  
*Februar 1989*

- Diekert, V.**
- 1) *Freie partiell kommutative Monoide: Ein algebraisches Modell zur Beschreibung von Nebenläufigkeit*  
*Institut für Informatik, Universität Stuttgart*
  
  - 2) *Rewriting on Traces*  
*Workshop, Free Partially Commutative Monoids*  
*Kochel am See*
  
  - 3) *Ersetzungssysteme über freien partiell kommutativen Monoiden*  
*Institut für Informatik, Universität Ulm*
-

- Eusterbrock, J.**
- 1) *Algorithmensynthese: Integration deduktiver und induktiver Komponenten*  
*Workshop: Verifikation, Konstruktion und Synthese von Programmen*  
*Karlsruhe*  
*6.–7. April 1989*
  
  - 2) *Lernen von optimalen Beweis– (= Spiel)strategien durch Konzeptformation*  
*Arbeitstreffen Maschinelles Lernen*  
*München*  
*13.–14. Juli 1989*
- Fehrle, Th.**
- Eine menüorientierte Klärungskomponente für ein natürlichsprachliches Aufkunftsystem*  
*Jahrestagung der Gesellschaft für linguistische Datenverarbeitung*  
*Ulm*  
*März 1989*
- Göttler, H.**
- 1) *Graphgrammars, a New Paradigm for Implementing Visula Lnguages*  
*RTA-89 (3rd International Conference on Rewriting Techniques and Application)*  
*Chapel Hill, North Carolina, USA*  
*03.04.1989*  
*(Wegen des Beginns meiner Lehrveranstaltung in Stuttgart und der Überschneidung mit einer Veranstaltung in Saarbrücken (s.u.) mußte ich einen Kollegen schicken, der meinen Vortrag hielt.)*
  
  - 2) *Das Juristische Literaturarchiv JULIA*  
*(Vortrag und Vorführung) „Informationstechnik am Arbeitsplatz von Juristen“*
  
  - 3) *Attributed Graph Grammars for Building Syntax-Directed Editors*  
*Informatik-Kolloquium der Universität Genf*  
*30.05.1989*
  
  - 4) *Diagramme sind attributierte Graphen*  
*Informatik-Kolloquium der Universität Kaiserslautern*  
*12.06.1989*

- Göttler, H. ff**
- 5) *Diagramme sind attributierte Graphen*  
*Informatik Kolloquium der Universität Braunschweig*  
*06.07.1989*
  - 6) *Graphgrammars, a New Paradigm for Implementing Visual Languages*  
*EUROGRAPHICS'89 (European Computer Graphics Conference and Exhibition)*  
*Hamburg*  
*08.09.1989*
  - 7) *Graphgrammars, a New Paradigm for Implementing Visual Languages*  
*(2nd European Software Engineering Conference)*  
*Warwick, Coventry, UK*  
*14.09.1989*
  - 8) *Graph-Based Modelling*  
*Informatik Kolloquium der Uniwersytet Jagiellonski Krakau*  
*27.09.1989*
  - 9) *Diagramme = Graphen + Attribute ? !!*  
*Informatik Kolloquium der Universität Oldenburg*  
*09.10.1989*
  - 10) *Das Juristische Literaturarchiv JULIA*  
*GI-Jahrestagung (Computergestützter Arbeitsplatz)*  
*München*  
*19.10.1989*
  - 11) *Diagramme = Graphen + Attribute ? !! (ausgewählte Beispiele)*  
*Informatik-Kolloquium der Universität Mainz*  
*03.11.1989*
  - 12) *Übergang vom funktionalen Entwurf zur Programmierung*  
*Workshop SEU-IS (Softwareentwicklungsumgebungen-Informationssysteme)*  
*Amt für Datenverarbeitung, Bundesakademie Mannheim*  
*20.11.1989*
-

- 
- 13) *Ein formales Modell zur Spezifikation und Implementierung von Software-Entwicklungsumgebungen*  
*Informatik-Kolloquium der Universität Tübingen*  
*06.12.1989*
- Gunzenhäuser, R.**
- 1) *Arbeitsmethoden der Informatik*  
*Historisches Kolloquium*  
*Universität Stuttgart*  
*Januar 1989*
  - 2) *Die Hochschulinformatik und ihr Einfluß auf den Informatikunterricht*  
*Universität Augsburg*  
*Januar 1989*
  - 3) *Seminar „Software Ergonomie“*  
*Lehrgangsleitung und Referate*  
*Technische Akademie Esslingen*  
*Juni 1989*
  - 4) *Algorithmen und Datenstrukturen*  
*Seminar IBM Deutschland*  
*Stuttgart*  
*Februar 1989*
  - 5) *Benutzerschnittstellen*  
*Seminar IBM Deutschland*  
*Stuttgart*  
*April 1989*
  - 6) *Tactile Representations of Scanned Documents*  
*European Communities Medical Programme Workshop*  
*on Access to Visual Computer Information by Blind Persons*  
*Berlin*  
*September 1989*
  - 7) *Intelligente Lernsysteme*  
*Universität Köln*  
*November 1989*
  - 8) *Wissensbasierte Mensch-Computer-Kommunikation*  
*Universität Bamberg*  
*Dezember 1989*
-

**Hanakata, K.**

- 1) *Object-oriented knowledge representation for language processing*  
*Laboratories of International Telephone and Telegraph*  
*Kamifukuoka, Japan*  
*April 1989*
  
- 2) *An object-oriented language COOL for knowledge engineering in practice*  
*University of Science Malaysia, Department of Computer Science*  
*Penang*  
*Juli 1989*
  
- 3) *Language generation from semantic networks for intercultural machine translation*  
*University of Science Malaysia, Department of Computer Science*  
*Penang*  
*August 1989*
  
- 4) *An object-oriented language COOL for knowledge engineering in practice*  
*University of Malaya, Computer Centre*  
*Kuala Lumpur, Malaysia*  
*Juli 1989*

**Heid, U.**

- 1) *A propos de la structuration d'un dictionnaire pour la génération automatique de textes techniques — peut-on réutiliser les hiérarchies terminologiques dans la génération de textes?*  
*Colloque: Terminologie et Industries de la langue*  
*Paris*  
*Januar 1989*
  
  - 2) *Analyse et génération morphosyntaxiques sur ordinateur: quelle place accorder au dictionnaire?*  
*Eingeladener Vortrag bei Forschungsgruppe SILEX des CNRS*  
*Lille*  
*Januar 1989*
-

- 3) *Collocations in multilingual generation*  
*EACL*  
*Manchester*  
*April 1989*
- 4) *Klassen im Übersetzungslexikon*  
*Sektion Computerlinguistik des Deutschen Romanistentags*  
*Aachen*  
*September 1989*

**Knödel, W.** *Numerische Untersuchung von Packungsalgorithmen*  
*Universität Innsbruck*  
*Oktober 1989*

- Knopik, Th.**
- 1) *AI-Methods for Structuring Hypertext-Information*  
*Konferenz HYPERTEXT II*  
*York, U.K.*  
*Juni 1989*
- 2) *Eine quantitative Methode zum Vergleich des Dialogverhaltens interaktiver Systeme*  
*3. Kolloquium Software-Entwicklungssysteme und -werkzeuge*  
*Technische Akademie Esslingen*  
*September 1989*
- 3) *Integration von Machine-Learning Verfahren in interaktive Wissensaquisitionskomponenten*  
*Workshop „Wissensaquisition“*  
*GMD Sankt Augustin*  
*September 1989*

- Lehmann, E.**
- 1) *KI und Expertensysteme*  
*8-stündiges Tutorium, veranstaltet vom Informatik-Verbund Stuttgart (IVS)*  
*Universität Stuttgart*  
*7. März 1989*
- 2) *Grundlagenkurs „Wissensrepräsentation“*  
*KIFS-89 (7. Frühjahrsschule Künstliche Intelligenz)*  
*Günne/Möhnesee*  
*11.-19.3.1989*

- Lichter, H.** *Software-Prototyping — eine Übersicht*  
*Sitzung des VDI/VDE-GMA 4.2*  
*Frankfurt*  
*21.4.1989*
- Ludewig, J.**
- 1) *Modelle der Software-Entwicklung: Abbilder oder Vorbilder?*  
*Antrittsvorlesung an der Universität Stuttgart*  
*28.6.1989*
  - 2) *Tutorium Software Engineering*  
*IVS*  
*20.2.1989*
  - 3) *Computer Aided Software Engineering (CASE): Wie es entstand, was es ist und was es sein sollte*  
*IBM Institut '89*  
*München, Beitrag Nr. 43*
  - 4) *CASE — eine kritische Übersicht*  
*Tagung der Technischen Akademie Esslingen*  
*September 1989*
  - 5) *Vom Atelier zur Werkstatt: Ansätze zur systematischen Software-Entwicklung*  
*Professorenkonferenz der Deutschen Bundespost*  
*Darmstadt*  
*November 1989 (Tagungsband in Vorbereitung)*
  - 6) *Seminare und Vorträge bei verschiedenen Firmen (Datensstrukturen und Algorithmen, Software Engineering, Prototyping, CASE)*
- Maier, D.** *ObjTalk an Object-Oriented Extension of LISP*  
*Workshop on Object-Oriented Methodology*  
*Baden-Baden*  
*April 1989*
- Rathke, C.**
- 1) *Beschreibungen als eine Erweiterung objektorientierter Programmierung*  
*Kolloquiumsvortrag bei GMD-IPSI*  
*Darmstadt*  
*August 1989*
-

- 
- 2) *Extending Object–Oriented Programming by Descriptions*

*IJCAI Workshop Object–Oriented Programming in AI  
Detroit, USA  
August 1989*

- 3) *Why do we need ObjTalk when we have CLOS?*

*Kolloquiumsvortrag, Department of Computer Science  
Boulder, USA  
1989*

- 4) *Incremental Programming of Spreadsheet Models*

*through Constraint Application  
Al–Azhar Engineering First Conference  
Kairo, Ägypten  
1989*

**Rust, S.**

- 1) *An Overview of the SYSEDIT Chip Design Environment*

*Euromicro 89  
Köln  
4.–8. September 1989*

- 2) *A Graphical Tool for the System and Architecture Design of Integrated Circuits*

*ITG/GI Workshop CAD–Umgebungen für den Entwurf  
integrierter Schaltungen und Systeme CAD–LAB  
Paderborn  
4.–5. Oktober 1989*

**Schwab, Th.**

*Dialogmodellierung in interaktiven Systemen*

*Software–Ergonomie '89  
Hamburg  
März 1989*

**Schweikhardt, W.**

*Die Stuttgarter Mathematischeschrift für Blinde*

*Tagung „Mathematischeschrift für Blinde“, Modellversuch  
„Informatik für Blinde“  
Universität Karlsruhe  
September 1989*

---

- Schwinn, B.**      *-RAPiD- A Data Flow Model for Implementierung  
Parallelism and Intelligent Backtracking in Logic Pro-  
grams  
PARLE89  
Eindhoven  
Juni 1989*
- Strothotte, Th.**      1) *Pictures Supporting Human–Computer Dialogs in Vi-  
deotex Systems  
European Telecommunications Standards Institute  
Workshop on a Future Videotex System  
Wien  
Januar 1989*
- 2) *Pictures in Advice–Giving Dialog Systems: From  
Knowledge Representation to the User Interface  
Kolloquiumsvortrag  
Universität Kopenhagen  
März 1989*
- 3) *Informationsvermittlung in interaktiven wissensbasier-  
ten Systemen durch bildhafte Darstellungen  
Software–Ergonomie '89  
Hamburg  
März 1989*
- 4) *Implizite Datenstrukturen  
Kolloquiumsvortrag an der Sektion Mathematik  
Universität Jena  
März 1989*
- 5) *Informationsvermittlung in interaktiven wissensbasier-  
ten Systemen durch bildhafte Darstellungen  
Kolloquiumsvortrag an der Sektion Mathematik  
Universität Jena  
März 1989*
- 6) *Using Computational Geometry in Knowledge–Based  
Dialog Systems  
Workshop über Geometrical Problems of Image Proces-  
sing, GEOBILD '89  
Georgenthal, DDR  
März 1989*
-

- Strothotte, Th. ff**
- 7) *Bildhafte Darstellungen in interaktiven wissensbasierten Systemen*  
*Kolloquiumsvortrag am Entwicklungslabor der IBM Deutschland GmbH*  
*Böblingen*  
*April 1989*
  - 8) *Bildhafte Darstellungen in interaktiven wissensbasierten Systemen*  
*Kolloquiumsvortrag*  
*Universität Basel*  
*Mai 1989*
  - 9) *Pictures in Advice-Giving Dialog Systems: From Knowledge Representation to the User Interface*  
*Graphics Interface '89*  
*London, Ontario*  
*Juni 1989*
  - 10) *Pictures in Advice-Giving Dialog Systems: From Knowledge Representation to the User Interface*  
*Kolloquiumsvortrag an den Universitäten*  
*Queen's University, Kingston, Ontario, Juni 1989*  
*University of Waterloo, Waterloo, Ontario, Juni 1989*  
*Simon Fraser University, Vancouver, B.C., Juli 1989*  
*Uppsala University, Schweden, Oktober 1989*
  - 11) *Aspects of Language and Pictorial Representations in Human-Computer Interaction*  
*European Communities Medical Programme Workshop on Access to Visual Computer Information by Blind Persons*  
*Berlin*  
*September 1989*
  - 12) *Bildhafte Darstellungen in interaktiven wissensbasierten Systemen*  
*Kolloquiumsvortrag*  
*Forschungsinstitut für Angewandte Wissensverarbeitung an der Universität Ulm*  
*Dezember 1989*

- 13) *Beyond Symbol Manipulation in Knowledge-Based Systems*  
*Kolloquiumsvortrag*  
*Universität Helsinki*  
*Dezember 1989*

**Weber, G.**

- 1) *FINGER — Formalizing Interaction for gesture Recognition*  
*Poster Session, Conference on Human Factors in Computing Systems, CHI 89*  
*Austin, Texas*  
*Mai 1989*
- 2) *Reading and Pointing— Modes of Interaction for Blind Users*  
*11th World Computer Congress*  
*San Francisco*  
*September 1989*
- 3) *Grundlagen der Datenverarbeitung*  
*Deutsche Bank*  
*Eschborn*  
*Dezember 1989*

**Welsch, C.**

- 1) *Programmiermethoden der 5. Generation.*  
*GNB*  
*Berlin*  
*März 1989*
- 2) *Ein Modell zur Integration kommunizierender, veränderlicher Objekte in die logische Programmierung*  
*Workshop Alternative Konzepte für Sprachen und Rechner*  
*Bad Honnef*  
*Mai 1989*
- 3) *Objektorientierte Programmierung.*  
*Hewlett-Packard*  
*Böblingen*  
*Juli 1989*
-

- 
- 4) *Reasoning Objects with Dynamic Knowledge Bases*  
*EPIA '89*  
*Lissabon*  
*September 1989*
- Zell, A.**
- 1) *Time Constrained Logic Programming*  
*Fifth Israeli Symposium on Artificial Intelligence, Vision and Pattern Recognition*  
*Tel Aviv, Israel*  
*Dec. 27–28, 1988*
- 2) *Iterative Deepening Prolog*  
*Second Scandinavian Conference on Artificial Intelligence*  
*Tampere, Finnland*  
*Juni 1989*
- 3) *Zeitbeschränktes Logisches Programmieren*  
*5. Österreichische Artificial Intelligence–Tagung*  
*Igls, Tirol*  
*März 1989*
- 4) *An Alternative Prolog Search Strategy*  
*2nd International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*  
*Tullahoma, Tennessee, USA*  
*Juni 1989*
- 5) *A Declarative Neural Network Description Language*  
*EuroMicro 89*  
*Köln*  
*September 1989*
- 6) *NetSim: Ein Simulator für Neuronale Netze*  
*GWAI-89, 13th German Workshop on Artificial Intelligence*  
*Eringerfeld*  
*September 1989*

### 3.5 Tagungen

- Burkhardt, W. H.**      *HICSS — 22 (Referent)  
Honolulu  
Januar 1989*
- Böcker, H.-D.**      *Mitglied im Programmkomitee der Tagung „Software  
Engineering '89“  
Universität Hamburg  
April 1989*
- Diekert, V.**      *Organisation der Tagung „Free Partially Commutative  
Monoids“ im Rahmen der EBRA-Working-Group No  
3166 „Algebraic and Syntactic Methods in Computer  
Science“ (ASMICS)  
Kochel am See  
2.-4.10.1989*
- Gunzenhäuser, R.**      1) *Vorsitzender des Programm-Ausschusses des 3. Work-  
shops „Intelligente Lernsysteme“  
Universität Tübingen  
Juni 1989*
- 2) *Mitglied des Programm-Komitees der 20. GI-Jahresta-  
gung 1990  
Universität Stuttgart*
- 3) *Mitglied des Programm-Ausschusses des Fachgesprächs  
„Software-Ergonomie“ 19. GI-Jahrestagung  
Universität München  
Oktober 1989*
- 4) *Mitglied des Programm-Ausschusses der 1. Internatio-  
nalen Tagung „Computers and Handicapped People“  
Wien  
August 1989*
- Knopik, Th.**      *Vorsitzender des Organisationskomitees der 20. Jah-  
restagung der Gesellschaft für Informatik  
Universität Stuttgart*
-

## 3.6 Herausgabe von Zeitschriften

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

## 3.7 Implementierungen

### Abteilung Betriebsssoftware

XPASCAL	<i>Spracherweiterung von Pascal für Arithmetik mit beliebiger Zahlenlänge und in Kreisteilungskörpern</i> <i>Sprache: Pascal, Modula-2</i> <i>K. Lagally</i>
WRG	<i>Top-Down-Parsergenerator mit automatischer Fehlerbehandlung</i> <i>Sprache: Pascal</i> <i>K. Lagally</i>
PPM	<i>PROLOG-2-WAM-Compiler</i> <i>übersetzt Prolog in optimierten Code der Warren Abstract Machine</i> <i>Sprache: C</i> <i>S. Schimpf, K. Brösamle</i>

### Abteilung Computersysteme

- SYSEDIT
- 1) *Flußdiagramm-Editor für Mikroprogramme*  
*Sprache: C*  
*S. Rust, U. Bachmann*
  - 2) *Ein grafisches Systementwurfsprogramm für ASICs*  
*Sprache: C*  
*S. Rust, J. Bosch*
  - 3) *Compiler von DACAPO-II nach SIL*  
*Sprache: C, lex, yacc*  
*S. Rust, J. Förderreuther*
  - 4) *Compiler von C nach SIL*  
*Sprache: C, lex, yacc*  
*S. Rust, M. Janosch*
  - 5) *Extraktor für das Gate-Array UA4*  
*Sprache: C*  
*S. Rust, U. Gerlach, G. Pauler*
-

- SYSEDIT ff                    6) *Plotter-Preview*  
*Sprache: C*  
*S. Rust, B. Hess, M. Spiegel*
- 7) *Full-Custom Layout-Editor,*  
*Sprache: C*  
*S. Rust, B. Schloß*
- 8) *Simulated Annealing Placement*  
*Sprache: C*  
*S. Rust, M. Siegle*
- 9) *Interpreter für eine regelbasierte Entwurfssprache*  
*Sprache: C*  
*S. Rust, M. Siegle*
- 10) *KARL-III Generatorprogramm*  
*Sprache: C*  
*S. Rust, B. Schloß*

WARREN                      *Eine Prolog Implementierung für Transputer-Systeme*  
*Sprache: C*  
*J. Walter, K. Winkler*

## Abteilung Dialogsysteme

- CTALK                        1) *Prototyp einer MS-Window-Benutzeroberfläche*  
*K. Hanakata*
- 2) *Prototyp einer X-Window-Benutzeroberfläche*  
*K. Hanakata*

HyperQuery                  *Ein interaktives Retrievalsystem für Reiseinformation*  
*J. Herczeg, Th. Schwab*

TRACK                        *Ein Werkzeug zur Realisierung eines graphisch visuellen Traces*  
*Sprache: Smalltalk-80*  
*J. Herczeg*

XIT                           *Ein User Interface Toolkit für X-Windows*  
*Sprache: Common LISP*  
*J. Herczeg*

KoFi	<i>Zeichenprogramm für Blinde auf der Basis von Gestenerkennung Sprache: C G. Weber</i>
SBT	<i>Anpassung des Stuttgarter Braille Terminals an einen PC, ermöglicht Zugang zu BTX Sprache: Pascal G. Weber</i>

## **Abteilung Intelligente Systeme**

FrameTalk	<i>Objektorientierte Sprache zur Repräsentation von Wissen Sprache: Common Lisp und Clos Ch. Rathke</i>
Analyse natürlicher Sprache	1) <i>Chart Parser Sprache: Common Lisp G. Burkert</i>
	2) <i>Fehlerkorrekturprogramm Sprache: Common Lisp P. Forster</i>

## **Abteilung Kombinatorik und Numerische Verfahren**

Entwurf und Analyse von Algorithmen und Datenstrukturen	<i>40 Beispiele zur Vorlesung Sprache: Pascal W. Knödel</i>
---	---

## **Abteilung Programmiersprachen und ihre Übersetzer**

Simulationssystem	1) <i>Ermittlung von Handlungshypothesen in einem Mehr-Agenten-Szenario Sprache: Arity Prolog (MS-DOS) P. Gerstl, A. Zell</i>
	2) <i>Erweiterung eines Prolog-Interpreters um Graphik-Primitive Sprache: Arity Prolog und Turbo C (MS-DOS) H. Bögeholz, A. Zell</i>

---

- 
- 3) *System zur Mode-Analyse logischer Programme*  
*Sprache: Pascal (SUN Workstations)*  
D. Merling, A. Zell
  - 4) *Meßsystem zum praktischen Vergleich verschiedener Unifikationsalgorithmen,*  
*Sprache: Pascal (SUN Workstations)*  
L. Reichert, A. Zell
  - 5) *Hypercard-Informationsprogramm zur Struktur des Instituts für Informatik*  
*Sprache: HyperCard (Apple Macintosh)*  
Ch. Sonntag, A. Zell
- NetSim
- 1) *Connectionist Network Simulator:*  
*Simulatorkern*  
*Sprache: C (SUN Workstations)*  
R. Bayer, A. Zell
  - 2) *Connectionist Network Simulator:*  
*Benutzeroberfläche*  
*Sprache: C, SunView (SUN Workstations)*  
T. Sommer, A. Zell
  - 3) *Connectionist Network Simulator:*  
*Netzwerk-Compiler*  
*Sprache: C, Yacc (SUN Workstations)*  
Th. Korb, A. Zell

# Abkürzungen

<b>ACM</b>	Association for Computing Machinery
<b>AIB</b>	Angewandte Informatik für Blinde (Forschungsgruppe)
<b>AK</b>	Arbeitskreis
<b>ALWR</b>	Arbeitskreis der Leiter wissenschaftlicher Rechenzentren
<b>BMBW</b>	Bundesministerium für Bildung und Wissenschaft
<b>DA</b>	Dienstauftrag
<b>BMFT</b>	Bundesministerium für Forschung und Technologie
<b>DFG</b>	Deutsche Forschungsgemeinschaft
<b>EATCS</b>	European Association for Theoretical Computer Science
<b>EBRA</b>	Esprit Basic Research Action
<b>FG</b>	Fachgruppe
<b>GI</b>	Gesellschaft für Informatik
<b>GID</b>	Gesellschaft für Information und Dokumentation
<b>GMD</b>	Gesellschaft für Mathematik und Datenverarbeitung
<b>HBFG</b>	Hochschulbauförderungsgesetz
<b>IAO</b>	Institut für Arbeitswissenschaft und Organisation
<b>IEEE</b>	Institute of Electrical and Electronics Engineers
<b>IEV</b>	Institut für Eisenbahn- und Verkehrswesen
<b>IFF</b>	Institut für Industrielle Fertigung und Fabrikbetrieb
<b>IFIP</b>	International Federation of Information Processing
<b>IGDD</b>	Interest Group of Distributed Data
<b>IMS</b>	Institut für Maschinelle Sprachverarbeitung
<b>IPA</b>	Institut für Produktionstechnik und Automatisierung
<b>IPVR</b>	Institut für Parallelle und Verteilte Höchstleistungsrechner
<b>ISO</b>	International Organization for Standardization
<b>IVS</b>	Informatik-Verbund Stuttgart
<b>LA</b>	Lehrauftrag
<b>MWK</b>	Ministerium für Wissenschaft und Kunst Baden Würtemberg
<b>NTG</b>	Nachrichtentechnische Gesellschaft
<b>SC</b>	Subcommittee

---

---

<b>SFB</b>	Sonderforschungsbereich
<b>SI</b>	Schweizerische Informatikergesellschaft
<b>SIG</b>	Special Interest Group
<b>SIGCHI</b>	Special Interest Group Computer Human Interaction
<b>TC</b>	Technical Committee
<b>WG</b>	Working Group