



JAHRESBERICHT

1990

Universität Stuttgart
Institut für Informatik

Azenbergstraße 12 7000 Stuttgart 1

Herausgeber: Prof. Dr. Egbert Lehmann
Geschäftsführender Direktor
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

Das Jahr 1990	1
1 Selbstverwaltung und Organisation	3
1.1 Leitung des Instituts für Informatik	3
1.2 Mitarbeit und Mitgliedschaft in Gremien	5
1.2.1 Universität Stuttgart	5
1.2.2 Wissenschaftliche Institutionen	8
1.2.2.1 Mitarbeit	8
1.2.2.2 Mitgliedschaft	9
2 Forschung	12
2.1 Forschungsvorhaben und Forschungsgruppen	12
2.1.1 Abteilung Betriebsssoftware	12
2.1.2 Abteilung Computer-Systeme	17
2.1.3 Abteilung Dialogsysteme	21
2.1.4 Abteilung Intelligente Systeme	31
2.1.5 Abteilung Kombinatorik und Numerische Verfahren	38
2.1.6 Abteilung Programmiersprachen und ihre Übersetzer	40
2.1.7 Abteilung Software Engineering	44
2.1.8 Abteilung Theorie der Informatik	46
2.2 Veröffentlichungen	49
2.3 Berichte	55
2.4 Vorträge	56
2.5 Tagungen	67
2.6 Herausgabe von Zeitschriften	68
2.7 Implementierungen	69

3 Fakultätsbezogene Aufgaben	72
3.1 Dekanat der Fakultät Informatik	72
3.2 Zentrale Fakultätseinrichtungen	72
3.2.1 Bibliothek	72
3.2.2 Rechnernetz	73
3.2.3 Workstations	74
3.2.4 Elektrotechnik	74
3.2.5 Mehrbenutzersysteme	77
3.2.6 PC-Pools	77
3.3 Lehre	78
3.3.1 Aufbau des Informatikstudiums	78
3.3.1.1 Diplomstudiengang Informatik	78
3.3.1.2 Nebenfachstudium Informatik	79
3.3.1.3 Weitere informatikorientierte Studiengänge	79
3.3.2 Lehrveranstaltungen	80
3.3.2.1 Lehrangebot im Sommersemester 1990	80
3.3.2.2 Lehrangebot im Wintersemester 1990/91	84
3.3.3 Informatik-Kolloquium	88
3.3.4 Habilitationen und Examensarbeiten	91
3.3.4.1 Dissertationen	91
3.3.4.2 Diplomarbeiten	92
3.3.4.3 Studienarbeiten	104
Abkürzungen	115

Das Jahr 1990

Nach Gründung der Fakultät Informatik 1988 und der 1989 erfolgten Trennung des neu gegründeten Instituts für Parallelle und Verteilte Höchstleistungsrechner (IPVR) vom Institut für Informatik (IfI) ging es im Jahr 1990 darum, einerseits die neu entstandenen Strukturen zu konsolidieren und andererseits die weiterhin drückende Überlast auf dem Lehrkörper durch Neuberufungen und Verbesserung der Arbeitsbedingungen zu vermindern. Letzteres soll vor allem durch den Umzug in ein gemeinsames Gebäude für alle Abteilungen der Fakultät erreicht werden. Beiden Zielen sind wir bis zum Jahresende 1990 durch gemeinsame Anstrengungen wesentlich näher gekommen.

Der durch das anhaltende Wachstum der Stuttgarter Informatik notwendig gewordene Umzug in ein Gebäude im Industriegebiet Stuttgart–Vaihingen soll nunmehr nach baulicher Erweiterung um ein zweites Obergeschoß im April 1991 erfolgen. Es ist zu hoffen, daß der Vorteil der gemeinsamen Unterbringung der Stuttgarter Informatik bei Verfügbarkeit einer größeren Anzahl studentischer Arbeitsplätze größer ist als die Nachteile einer wenig stimulierenden örtlichen Umgebung und der räumlichen Isolation von anderen Einrichtungen unserer Universität.

Die Durchführung des Lehrbetriebs für den Diplomstudiengang Informatik, für das Nebenfach Informatik und für die Grundlagenausbildung in Informatik — eine nach Bedeutung und Umfang ständig zunehmende Dienstleistung — blieb bei der vorhandenen erheblichen Überlast auch im Berichtsjahr 1990 das beherrschende Thema. Im Wintersemester 1990/91 hatten wir 250 Studienanfänger, obwohl nur eine Kapazität von etwa 160 Anfängerplätzen vorhanden ist.

Diese kräftezehrenden Aufgaben der Lehre wirkten sich leider in erheblichem Maße negativ auf die laufenden Forschungsarbeiten aus, die im wesentlichen nur im Rahmen von Drittmittelvorhaben durchgeführt werden können. Daher waren im Berichtsjahr unsere Bemühungen auf die Besetzung mehrerer vakanter C4-Professuren gerichtet. Für die im Institut für Informatik angesiedelten Stellen „Theorie der Informatik“ (Nachfolge Prof. Schwabhäuser), „Programmiersprachen und ihre Compiler“ (Nachfolge Prof. Barth) sowie die neu eingerichtete Stelle „Grundlagen der Informatik“ wurden Berufungsvorschläge erarbeitet, die die Zustimmung des Senats fanden.

Der Ausbau der Infrastruktur des Instituts bzw. der Fakultät Informatik hat im zurückliegenden Jahr weitere Fortschritte gemacht. Die gerätetechnische Ausstattung für die Lehre im Diplomstudiengang Informatik wurde vor allem durch die Beschaffung von zwei leistungsfähigen Rechnersystemen HP 845 ergänzt.

Zur immer dringlicher werdenden Verkürzung der mittleren Studienzeiten unserer Studenten wurde ein ganzes Bündel von Maßnahmen wie strikte Einhaltung der sechsmonatigen Bearbeitungsdauer von Studien- und Diplomarbeiten, transparentere

Gestaltung des Studienplans, Referenzstudienpläne für die einzelnen Studienschwerpunkte, bessere Abstimmung der Lehrveranstaltungen aufeinander und Bewertung der Lehrveranstaltungen durch die Studenten beschlossen.

Das Jahr der Wiederherstellung der deutschen Einheit brachte uns intensivierte Kontakte mit Kollegen und Studenten aus der ehemaligen DDR. So weilten bereits im Sommersemester 1990 insgesamt 20 DDR–Studenten der Informatik für ein Semester an unserer Universität. Mehrere gebrauchte VAX–Rechenanlagen wurden als Spende an Universitäten der neuen Bundesländer abgegeben. Im Wintersemester 90/91 lernten wir Prof. Posthoff (TU Chemnitz) als Gastprofessor kennen und schätzen.

Die unter Vorsitz von Prof. Reuter, dem Dekan unserer Fakultät, und unter der dankenswerten aktiven Beteiligung zahlreicher Angehöriger unseres Instituts (besonders erwähnt sei hier Herr Knopik als Vorsitzender des Organisationskomitees) an der Universität Stuttgart abgehaltene 20. Jahrestagung der Gesellschaft für Informatik kann aufgrund repräsentativer Umfragen bei externen Teilnehmern als voller Erfolg bezeichnet werden.

Februar 1991

Egbert Lehmann

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

Betriebssoftware	Prof. Lagally
Computer Systeme	Prof. Burkhardt
Dialogsysteme	Prof. Gunzenhäuser
Intelligente Systeme	Prof. Lehmann
Kombinatorik und Numerische Verfahren	Prof. Knödel
Programmiersprachen und ihre Übersetzer	Dr. Göttler (<i>Lehrstuhlvertr. bis 31.3.</i>)
Software Engineering	Prof. Ludewig
Theorie der Informatik	Prof. Knödel (<i>kommissar.</i>) Dr. Diekert (<i>Lehrstuhlvertr. bis 31.3.</i>) Prof. Schmidt (<i>Lehrstuhlvertr. 23.4.–31.8.</i>)

1.2 Mitarbeit und Mitgliedschaft in Gremien

1.2.1 Universität Stuttgart

Böcker, H.D.	Mitglied im Prüfungsausschuß der Fakultät Informatik Mitglied im Fakultätsrat der Fakultät Informatik
Burkhardt, W.H.	Mitglied im Fakultätsrat der Fakultät Informatik Vorsitzeder der Studienkommission (bis Okt.90) Mitglied der Berufungskommission <i>Praktische Informatik: Verteilte Systeme</i> Mitglied der Berufungskommission <i>Grundlagen der Informatik</i>
Diekert, V.	Mitglied des erweiterten Fakultätsrats Informatik (bis 31.3.)
Gunzenhäuser, R.	Mitglied im Fakultätsrat der Fakultät Informatik Stellv. Vorsitzender der Studienkommission <i>Informatik</i> Vorsitzender der Studienkommission <i>Informatik</i> (ab 18.06.) Mitglied des Großen Senats (bis 30.09.) Mitglied im Prüfungsausschuß <i>Techn.-orientierter Diplomkaufmann</i> Mitglied der Berufungskommission <i>Praktische Informatik: Bildverstehen</i> Mitglied der Berufungskommission <i>Theoretische Linguistik</i> Mitglied der Senatskommission <i>Koordinierungsausschuß Informatik</i>
Knödel, W.	Prodekan der Fakultät Informatik (bis 31.3.) Mitglied des Großen Senats (bis 31.9.) Vorsitzender der Berufungskommission <i>Theoretische Informatik</i> (Nachfolge Schwabhäuser) Mitglied der Berufungskommission <i>Programmiersprachen und ihre Übersetzer</i> (Nachfolge Barth)

	Mitglied der Senatskommission Informatik (bis 31.9.)
	Kooptiertes Mitglied des Fakultätsrats Mathematik
	Mitglied der Berufungskommission für die C3-Professur <i>Mathematische Stochastik</i>
	Mitglied der Berufungskommission für die C3-Professur <i>Topologie und Algebra</i>
Krause, K.	Mitglied im Fakultätsrat der Fakultät Informatik
Lagally, K.	Mitglied des Fakultätsrats Informatik Vorsitzender des Prüfungsausschusses Informatik 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>Bildverstehen</i> Mitglied der Berufungskommission <i>Formale Konzepte der Informatik</i>
Lehmann, E.	Mitglied des Fakultätsrats Informatik Mitglied des Prüfungsausschusses Informatik Vorsitzender der Berufungskommission <i>Bildverstehen</i> Mitglied der Berufungskommission <i>Programmiersprachen und ihre Übersetzer</i> Mitglied der Berufungskommission <i>Phonetik</i> (Fak. 11) Senatsberichterstatter in der Berufungskommission <i>Textil- und Faserchemie</i> (Fak. 3)
Lichter, H.	Mitglied der Berufungskommission <i>Programmiersprachen und ihre Übersetzer</i>
Ludewig, J.	Stellvertretender geschäftsführender Direktor des IfI Sprecher des IVS (Informatik-Verbund Stuttgart) Mitglied des Fakultätsrats der Fakultät Informatik Leiter der Software-Kommission (Kommission der Fakultät Informatik) Vorsitzender der Berufungskommission <i>Programmiersprachen und ihre Übersetzer</i> Mitglied der Berufungskommission <i>Verteilte Systeme</i>

	Mitglied der Berufungskommission <i>Formale Konzepte der Informatik</i>
	Mitglied der Senatskommission <i>Koordinierungsausschuß Informatik</i>
Rathke, Ch.	Mitglied der Berufungskommission <i>Bildverständen</i>
Reuß, W.	Mitglied des Fakultätsrats Informatik (bis 30.9.) Mitglied der Berufungskommission <i>Theoretische Informatik</i>
Tausend, B.	Mitglied der Studienkommission Mitglied im erweiterten Fakultätsrat

1.2.2 Wissenschaftliche Institutionen

1.2.2.1 Mitarbeit

Diekert, V.	Partner der EBRA-Working Group No 3166 <i>Algebraic and Syntactic Methods in Computer Science</i> (ASMICS)
Gunzenhäuser, R.	Stellvertretender Sprecher der Fachgruppe 7.0.1 <i>Intelligente Lernsysteme</i> der Gesellschaft für Informatik Vorsitzender des Fakultätentages Informatik Vorsitzender der Kommission <i>Rahmenprüfungsordnung Informatik</i> des Fakultätentages und der Hochschulrektorenkonferenz Mitglied von wissenschaftlichen Projektberatungsgremien an den Universitäten Hagen und Karlsruhe Mitglied des wiss. Beirats der Akademischen Software Kooperation (Karlsruhe / Bonn) Komm. Sprecher der ASK-Softwaregruppe Informatik
Knödel, W.	titl.a.o.Prof. der TU Wien Gründungsmitglied der Gesellschaft für Informatik
Ludewig, J.	Sprecher der GI-Fachgruppe 4.3.1 <i>Requirements Engineering in der industriellen Anwendung</i>
Mahling, A.	Redaktionelles Mitglied der <i>Studio-Blätter</i> (herausgegeben von der Staatlichen Hochschule für Musik und Darstellende Kunst Stuttgart)

1.2.2.2 Mitgliedschaft

American Association for Artificial Intelligence	Lehmann, Rathke
APL-Club Germany	Schweikhardt
Association for Computing Machinery (ACM)	Burkhardt, Berger, Lagally, Lehmann, Rathke, Ziegler
ACM-SIG Computer for the Physically Handicapped	Weber
Association for Computational Linguistics	Lehmann
Deutsche Mathematiker-Vereinigung	Knödel
DIN-Ausschuß ATHBeh-5 (Ausschuß Technische Hilfen für Behinderte) <i>Kommunikationshilfen für sensorisch Behinderte</i>	Schweikhardt
European Association for Theoretical Computer Science (EATCS)	Diekert, Ebinger, Knödel, Reinhardt
Gesellschaft für Angewandte Mathematik und Mechanik	Gunzenhäuser, Knödel
Gesellschaft für Didaktik der Mathematik	Gunzenhäuser
Gesellschaft für Informatik	Berger, Bräunl, Burkert, Burkhardt, Deininger, Eggenberger, Eusterbrock, Forster, Gunzenhäuser, Hanakata, Knödel, Knopik, Kümmel, Lagally, Lehmann, Licher, Ludewig, Maier, Merkel, Nitsche, Rathke, Ressel, Reuß, K. Schneider, Schweikhardt, Schwille, Tausend, Welsch, Zell, Ziegler, Zipperer
GI-Arbeitskreis 3.1.2 <i>Architektur von Betriebssystemen</i>	Lagally

GI–Arbeitskreis 4.3.2 <i>Wissensbasierte Systeme für das Prototyping</i>	Lichter
GI–Arbeitskreis 7.1.0 <i>Weiterbildung in Informatik</i>	Gunzenhäuser
GI–Arbeitskreis <i>Informatik und Behinderte</i> der GI–Fachgruppe 2.3.1 <i>Software–Ergonomie</i>	Schweikhardt
GI–Fachausschuß 3.2 <i>Systemarchitektur</i>	Lagally
GI–Fachausschuß 7.1 <i>Informatik an Hochschulen</i>	Gunzenhäuser
GI–Fachbereich 7 <i>Ausbildung und Beruf</i>	Gunzenhäuser
GI–Fachgruppe 1.1.1 <i>Theoretische Informatik</i>	Knödel, Kümmel
GI–Fachgruppe 1.1.4 <i>Wissensrepräsentation</i>	Lehmann
GI–Fachgruppe 1.2.5 <i>Sprachanalyse</i>	Kümmel
GI–Fachgruppe 1.3.1 <i>Natürlichsprachliche Systeme</i>	Lehmann
GI–Fachgruppe 1.5.1 <i>Knowledge Engineering</i>	Knopik
GI–Fachgruppe 2.1.1 <i>Software Engineering</i>	Lichter, Ludewig
GI–Fachgruppe 3.3.1 <i>Rechnernetz</i>	Schweikhardt
GI–Fachgruppe 3.5.1 <i>Methoden des Entwurfs und der Verifikation digitaler Schaltungen</i>	Rust
GI–Fachgruppe 3.5.4 <i>CAD–Umgebungen für den Entwurf integrierter Schaltungen und Systeme</i>	Rust

GI-Fachgruppe 3.5.5 <i>Architekturen für hochintegrierte Schaltungen</i>	Kümmel
GI-Fachgruppe 4.3.1 <i>Requirements Engineering in der industriellen Anwendung</i>	Ludewig
GI-Fachgruppe 7.0.1 <i>Intelligente Lernsysteme</i>	Gunzenhäuser
GI/NTG Fachgruppe <i>Fehlertolerante Systeme</i>	Burkhardt
GI/NTG Fachgruppe <i>Rechnernetze</i>	Burkhardt
Gesellschaft für Mathematik, Oekonomie und Operations Research	Knödel
Gesellschaft für Mathematisch-naturwissenschaftlichen Unterricht	Gunzenhäuser
IEEE	Burkhardt
IEEE Computer Society	Burkhardt, Ludewig, Rathke (Affiliate)
IEEE Design Automation Standards Subcommittee	Burkhardt
ISO/TC 173/SC 4 <i>Aids and Adaptations for Communication</i>	Schweikhardt
Linguistische Grundlagen für Sprachverarbeitung (Graduiertenkolleg an der Universität Stuttgart)	Lehmann
Österreichische Mathematische Gesellschaft	Knödel
Österreichische Statistische Gesellschaft	Knödel
Schweizerische Informatikergesellschaft	Ludewig
Verein Deutscher Ingenieure (VDI)	Schöbel

2 Forschung

2.1 Forschungsvorhaben und Forschungsgruppen

2.1.1 Abteilung Betriebsssoftware

Leiter	<i>Lagally</i>
Professor	<i>Eggenberger</i>
Wiss. Mitarbeiter	<i>Armbruster (bis 30.4.), Kümmel, Schimpf, Schöbel (ab 19.10.), Ziegler</i>
Programmierer	<i>Merkel, Schlebbe</i>
Verwaltungsangestellte	<i>Groß (bis 30.11.)</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. Die Benutzerschnittstelle zur interaktiven Bedienung des Systems wurde um ein Hilfesystem ergänzt. Dieses basiert auf Techniken, wie sie von Hypertext-Systemen bekannt sind. Als weiteres Hilfsmittel zur Kommunikation paralleler Prozesse wurden Funktionen zum Austausch von Botschaften implementiert.

Durch die Entwicklung einer Platine mit einem AD- und einem DA-Wandler entstand eine preisgünstige Möglichkeit zur Ein-/Ausgabe analoger Signale. Für Testzwecke wurden Programme zur Erkennung der Hardware-Konfiguration eines PCs und zur Beobachtung von internen Daten des Standard-Betriebssystems erstellt.

Platzsparende Parsingverfahren für allgemein kontextfreie Grammatiken

(*Thomas Schöbel*)

Allgemein kontextfreie Grammatiken kommen in vielen Anwendungen vor, bei denen mehrdeutige Sachverhalte dargestellt werden müssen. Beispiele dafür sind die Verarbeitung von natürlichen Sprachen, das Erkennen von Mustern in Chromosomen oder spezielle Programmiersprachen. Aus der Theorie der formalen Sprachen ist bekannt, daß es kontextfreie Grammatiken gibt, für die es eine exponentiell wachsende Zahl von möglichen Ableitungsbäumen gibt. Bereits bekannte Verfahren zum Parsen dieser Grammatiken (Earley, CYK) können diese Ableitungsbäume durch eine Datenstruktur darstellen, die $O(n^3)$ Platz beansprucht, wobei n die Länge des Eingabewortes darstellt.

Es wurde ein neues Parsingverfahren entwickelt, das zur Darstellung aller möglichen Syntaxbäume nur noch $O(n^2)$ Platz benötigt. Dabei ist der Zeitbedarf mit $O(n^3)$ asymptotisch genau so gut wie bei allen anderen praktikablen Algorithmen. Zur Zeit wird untersucht, wie gut die konstanten Faktoren des Verfahrens im Vergleich zu den bereits bekannten Verfahren liegen. Es konnte bereits gezeigt werden, daß das neue Verfahren asymptotisch gleich schnell wie das Earley–Verfahren ist, auch im Falle eindeutiger bzw. $LR(k)$ –Grammatiken.

Da der Platzbedarf von Algorithmen eine weit kritischere Größe darstellt als der Zeitbedarf, ist zu hoffen, daß das neue Verfahren das bisher meistverwendete Earley–Verfahren ersetzen kann. Dazu werden im nächsten Jahr ausführliche Vergleiche ange stellt werden.

Projekt: Parsergenerator mit automatischer Fehlerbehandlung

(*Klaus Lagally*)

Der im Vorjahr entwickelte Parser–Generator WRG wurde weiterentwickelt und um weitere Betriebsmodi ergänzt. Neben einer vollen Syntaxfehlerbehandlung in den erzeugten Parsern, die nach Erkennen eines Syntaxfehlers versucht, wieder aufzusetzen, gibt es auch die Möglichkeit, nach dem ersten Fehler abzubrechen, sowie eine Version, die korrekte Eingabe voraussetzt. Die beiden letzten Fälle sind sinnvoll, wenn die Eingabe automatisch generiert ist, wie etwa bei späteren Pässen eines Compilers, und können wesentlich effizienter behandelt werden. Auch für den inhärent aufwendigen Fall eines wieder aufsetzenden Parsers lassen sich durch geänderte interne Datenstrukturen wesentliche Laufzeitgewinne erreichen; diese Entwicklung ist noch nicht abgeschlossen.

Projekt: Verarbeitung arabischer Texte

(*Klaus Lagally, Mohsen Mardassi (Student)*)

Die automatische Aufbereitung von mehrsprachigen Texten bereitet dann erhebliche Schwierigkeiten, wenn eine der beteiligten Sprachen einen speziellen Zeichensatz benötigt und/oder linksläufig ist. Dies betrifft speziell die hebräische und die arabische Schrift, wobei die letztere als verbundene Schreibschrift zusätzliche Probleme bringt.

Für die reine Textverarbeitung liegen bereits eine Reihe von Industrie-Entwicklungen vor, die jedoch alle spezielle Hardware-Einrichtungen benötigen, deren Beschaffung sich für einen gelegentlichen Einsatz nicht lohnt. Außerdem ist für deren Bedienung eine spezielle Schulung erforderlich.

Für das am Institut verbreitet eingesetzte System L^AT_EX wurde daher eine Erweiterung entwickelt, die es gestattet, arabische Texte und die gewohnten rechtsläufigen Sprachen zu kombinieren. Sie besteht aus einem T_EX-Makropaket und einem arabischen NASH-Zeichensatz. Die Eingabekonventionen lehnen sich an die übliche wissenschaftliche Transkription an, spezielle Hardware ist nicht erforderlich.

Lastverteilung und Speicherverwaltung bei der parallelen Ausführung von Logikprogrammen

(*Stefan Schimpf*)

Für die sequentielle Ausführung von Logikprogrammen wurden in den letzten zehn Jahren sehr effiziente Verfahren entwickelt. Diese Verfahren basieren jedoch auf der Verfügbarkeit eines globalen Speichers, auf den schneller Zugriff möglich ist, sowie auf einer Stapelverwaltung, die auf eine festgelegte sequentielle Ausführungsreihenfolge zugeschnitten ist.

Das Ziel der Forschung ist die Entwicklung eines Ausführungsmodells für ein Multiprozessorsystem, das einerseits die Einschränkung der Ausführungsreihenfolge der Teilaufgaben lockert und auf einen gemeinsamen Speicher verzichtet, andererseits aber die Vorteile der sequentiellen Ausführungsmodelle soweit als möglich beibehält, um in einer Situation gleichmäßiger Lastverteilung mit jedem Prozessor etwa so schnell wie ein sequentielles Verfahren zu sein.

Das parallele Modell soll auf einem kleinen Testfeld von vier Transputern implementiert werden. Messungen über den Kommunikationsaufwand bei verschiedenen Strategien zur Aufgabenverteilung sollen zunächst auf diesem kleinen Testfeld entwickelt und später eventuell auf einem größeren Testfeld durchgeführt werden.

Ein sehr schneller Algorithmus für Mustererkennung in Texten

(*Udo Merkel, Bernhard Ziegler*)

Daniel M. Sunday hat in [1] drei Suchalgorithmen vorgestellt, die nach seinen Untersuchungen schneller als das Verfahren von Boyer-Moore sein sollen (schnellstes bislang bekannte Verfahren). Wir konnten jedoch zeigen, daß Sundays Messungen inkorrekt sind, weil sie sich lediglich auf idealisierte Darstellungen des Boyer-Moore-Algorithmus beziehen, in denen der Vergleich Muster-Text explizit durchgeführt wird.

In Boyer und Moores Originalversion werden zwei Fälle getrennt behandelt dergestalt, daß der schnellere von beiden ausschließlich über eine Verschiebefunktion fortgeschaltet wird. Für Muster der Länge 1 kommt es zu überhaupt keinem, für Muster der Länge > 1 nur in einer „Buchstabe paßt“-Situation zum Umschalten in den langsameren Fall des expliziten Vergleichs zwischen Text und Muster. Die Zahl solcher Vergleiche kann also nicht, wie von Sunday unterstellt, als Maß der Schnelligkeit eines Suchalgorithmus dienen.

Aber auch wenn man die Zahl der Zugriffe auf den Text als Maß heranzieht, stellt sich heraus, daß die von Sunday vorgeschlagenen Verbesserungen sich nicht lohnen. Sundays Algorithmen benötigen bei jeder Verschiebung einen zusätzlichen Textzugriff; da nun bei natürlichen Sprachen die schnellere „Paßt nicht“-Situation zugleich die häufigere ist, muß die Zahl der Verschiebungen je Zugriff sich im Mittel verringern — unseren Untersuchungen zufolge bis auf die Hälfte [2].

Sundays Algorithmen lassen sich durch nachträgliches Implantieren der „Paßt/Paßt nicht“-Fallunterscheidung verbessern; der Effizienzgewinn reicht jedoch nicht, um Boyer und Moores ingeniosen Algorithmus auszustechen.

- [1] D.M. Sunday, Comm. ACM 33, 8, pp.132-142 (1990)
- [2] B. Ziegler, Bericht 2/90, Fakultät Informatik, Univ. Stuttgart

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“
-

Zu vorstehend 1. und 2. stehen Manuskripte für Buchmonographien zur Publikation an.

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 Deuterokriterien beziehen. Es wurden weitere, dreidimensional abgeleitete, Kugelalgorithmen entwickelt. Dabei helfen die Deuterokugel-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 (*Rochlitzer*)

Kernforschungszentrum Karlsruhe (*Trauboth*)

Kernforschungszentrum Karlsruhe, IDT (*Jaeschke*)

MPI für Biologische Kybernetik, Tübingen (*Reichardt*)

GMD-Forschungsstelle Karlsruhe (*Grosch*)

Universität Erlangen–Nürnberg (*Fischer*)

2.1.2 Abteilung Computer–Systeme

Leiter	<i>Burkhardt</i>
Mitarbeiter	<i>Krause, Homeister, Moser, Rust (bis 30.6.), Sticcotti (ab 5.11.), Weiss, Walter, Zimmermann (ab 1.9.), Zipperer</i>

Übersicht

Der Berichtszeitraum war gekennzeichnet durch große Personalfluktuation. Darunter haben die Forschungsvorhaben sehr gelitten. Dennoch konnten beachtliche Ergebnisse erzielt werden.

Im Forschungsvorhaben über das Verhalten paralleler Systeme wurden die Arbeiten über den Einfluß von Ein- und Ausgabe, sowie Kommunikation darin, auf das Systemverhalten weiter fortgeführt.

Ein massiv paralleles Multicomputersystem zur Berechnung entkoppelter Probleme

(*Dieter Homeister*)

Für die Problemklasse der vollständig entkoppelten Probleme wird eine geeignete Parallelrechner–Architektur, Topologie und Lastbalancierungs–Software entwickelt. Bisherige statische Scheduling–Verfahren erwiesen sich dabei als ungeeignet. Durch die Rekonfigurierbarkeit des Systems kann der Rechner außer für entkoppelte Berechnungen auch als Universalrechner genutzt werden. Dies wurde durch den Betrieb mit dem verteilten Transputerbetriebssystem Helios gezeigt.

Das System besteht aus 25 Transputern ohne globalen Speicher. Es wird Wert auf eine sehr hohe Effizienz des Parallelrechners gelegt, um die Lösung auch auf Systeme mit sehr vielen Prozessoren übertragen zu können. Dazu wurden systematische alle Verlustquellen untersucht und weitgehend ausgeschaltet. Mit 25 Prozessoren ist noch ein nahezu linearer Speedup möglich.

Eine dieser Verlustquellen ist durch die Programmladezeit bedingt. Daher wurde ein Verfahren entwickelt, um Transputersysteme parallel booten zu können. Transputer–netzwerke werden gewöhnlich sequentiell gebootet. Das hier entwickelte Verfahren ermöglicht es, beliebig viele Transputer gleichzeitig mit demselben Code zu laden, ohne daß dies wesentlich länger dauert als das Laden eines einzelnen Transputers. Auf diese Art können auch große Datenmengen sehr schnell in großen Transputernetzen verteilt werden.

Transputer kommunizieren über sogenannte Links, bidirektionale serielle Zweidraht-Verbindungen mit bis zu 20 MBit/s. Sie arbeiten mit einem Byteprotokoll mit Handshake für jedes Byte auf der entgegengesetzten Leitung. Zum parallelen Booten werden diese Transputerlinks verzweigt. Dies ist zwar vom Entwickler der Transputer nicht vorgesehen worden, kann aber dennoch unter Einhaltung aller Spezifikationen realisiert werden. Eine Zusatzlogik, die in 2 GALs Platz findet, sorgt für das korrekte Zeitverhalten. Diese Anordnung kann auch als serieller Datenbus mit einem Sender und vielen Empfängern interpretiert werden. Umfangreiche Meßreihen ergaben eine völlige Fehlerfreiheit.

Automatisierung des Entwurfsprozesses für integrierte Schaltungen

(*Stefan Rust (bis 30.6.90)*)

Im Rahmen des Forschungsprojektes wurde eine interaktive, graphische Entwurfsumgebung (SYSEDIT) entwickelt, die den explorativen Systementwurf von integrierten Schaltungen unterstützt.

SYSEDIT stellt eine durchgehende Unterstützung des Entwurfsprozesses von der Entwicklung der Verhaltensbeschreibung, der Erstellung der Architekturbeschreibung auf Register-Transfer Ebene bis zur Chip-Implementierung zur Verfügung.

Das Entwurfssystem beinhaltet eine umfangreiche Toolbox für die Phasen des System- und Architekturentwurfs und die Bewertung des Entwurfsergebnisses.

Diese Funktionen erlauben ein „Fast Prototyping“ unterschiedlicher Entwurfsalternativen. Einzelne Entwurfsschritte lassen sich unter Verwendung einer regelbasierten Entwurfssprache anwendungsspezifisch automatisieren.

Die Entwurfssystem umfaßt die graphische Unterstützung für die Erstellung und Optimierung der Verhaltensbeschreibung und ihre Simulation und Bewertung. Die Bewertung erlaubt die Verteilung einzelner Systemfunktionen auf die Software- oder Hardwareebene einer spezialisierten integrierten Schaltung.

Es stehen Synthesefunktionen für die Umsetzung der Verhaltensbeschreibung in reihbare Prozessorstrukturen zur Verfügung. Die generierten Architekturbeschreibungen lassen sich sofort graphisch darstellen und in Simulationsmodelle in der Hardwarebeschreibungssprache KARL-III umsetzen. Die Simulation erlaubt eine Verifikation des Systems auf Hardware-, Mikroprogramm- und Softwareebene.

Ein Floorplanner für die Bewertung der Chip-Implementierung erlaubt die frühzeitige Abschätzung der Realisierbarkeit des Entwurfs in einem Semi-Custom ASIC.

Die noch fehlenden Komponenten für diese durchgehende Entwurfsunterstützung konnten fertiggestellt und die Funktionsweise des Entwurfssystems SYSEDIT an hand von Entwurfsbeispielen am Tag der offenen Tür des Instituts demonstriert werden.

Das im Rahmen des Forschungsprojektes mit SYSEDIT entworfene Sortierchip für die Bildverarbeitung konnte durch die Unterstützung des Instituts für Mikroelektronik (IMS), Stuttgart (Herr Roos, Herr Dr. Schwederski, Prof. Dr. Höflinger) hergestellt werden.

Tests am IMS unter Verwendung der speziellen Festfunktionen des Sortierchips zeigten die korrekte Arbeitsweise des Sortierchips "on first silicon".

Die Daten des Chips sind:

Technologie	ACMOS 2um Gate Forest
Chip	9.6 mm x 8.7 mm
Core	7.0 mm x 5.7 mm benutzt
Anzahl Zeilen	1879
Anzahl Transistoren	18 000 (aktiv)
I/O Pads	80 Signal + 40 Power
Leistung	0.9 W (Worst case)
Speed	min. 5 Mhz Pixel Takt

Dabei konnte bei der Verlagerung des Sortieralgorithmus in eine integrierte Schaltung gegenüber einer Software-Lösung unter Verwendung eines 68020 Mikroprozessors (16 Mhz) eine Leistungssteigerung von über 700x in der Verarbeitungsgeschwindigkeit erreicht werden.

Im Rahmen dieser Fallstudie wurde gezeigt, wie durch die umfassende Entwurfsunterstützung für den Systementwurf eine Verkürzung der Entwicklungszeit des Sortierchips um fast die Hälfte erreicht werden kann.

SPARC Koprozessor Board für PC

(Jörg Walter)

Ähnlich den in der Abteilung vorhandenen Transputersystemen sollte mit einem Cypress CY7C601SPARC Prozessor eine Einschubkarte für PCs entwickelt werden. Diese Technik hat den Vorteil, daß Filesystem und Bildschirmsteuerung sowie die Tastatureingabe vom PC abgehandelt werden. Dadurch vereinfacht sich der Entwurf und der Benutzer trifft auf eine bekannte Umgebung. Die SPARC Karte ist für maximal 16Byte Hauptspeicher ausgelegt und besitzt zusätzlich 64Byte schnelles statisches RAM. Da zum Zeitpunkt der Entwicklung noch kein Cache-Controller verfügbar war, bildet der statische Speicher einen Teil des Hauptspeichers.

Über eine serielle Schnittstelle kann auch von einem Terminal aus auf die Karte zugegriffen werden, die Taktfrequenz des Systems beträgt 22MHz.

Durch die Komplexität der erforderlichen Logik war eine Realisierung mit den verfügbaren Bestückungstechniken (einseitig, keine SMD-Technik) auf einer PC Einsteckkarte nicht möglich. Das Layout beanspruchte 2 Doppel-Europakarten mit je 6 Lagen, deren Fertigung aus finanziellen Gründen nicht realisiert werden konnte.

Um Projekte dieser Art durchführen zu können, wurde mit der Firma Wandel & Goltermann in Eningen eine Absprache getroffen, nach der Layouts der Abteilung bei WG gefertigt werden.

Eine Bedingung dafür war, daß die Layouts nahtlos in den Fertigungsprozeß bei WG übernommen werden können. Für die Schaltplan–Eingabe und Leiterplattenentwicklung wird CADSTAR verwendet, das auf PCs läuft und im interaktiven Betrieb brauchbare Layouts liefert. Schwierigkeiten gab es vor allem bei der Anpassung der Schnittstelle zu den NC–gesteuerten Bohr– und Fräsmaschinen, die jedoch mit Hilfe der Mitarbeiter von Wandel & Goltermann bewältigt wurden.

Das erste Layout, eine 4–lagige Platine für ein Transputersystem wurde kurz vor Jahresende fertiggestellt. Mit der nun im Hardware–Labor vorhandenen Ausrüstung, vom elektronischen Leiterplattenentwurf über die Simulation bis hin zu Meßgeräten wie Logikanalyzer und Digitaloszilloskop können nun auch größere Projekte bis zur Hardware–Ebene realisiert werden.

SYSEdit–Weiterentwicklung

(*Hans-Georg Zipperer*)

Die Arbeiten an der integrierten Entwurfsumgebung SYSEdit, die von Herrn Rust begonnen wurden, wurden fortgeführt. Es wurde ein Programmpaket zur Synthese von mikroprogrammierten und hartverdrahteten Steuerwerken implementiert. Weitere Arbeiten sind derzeit im Gange.

Zur Unterstützung des Entwurfs von Schaltungen mit Xilinx–LCAs wurden 2 Compiler für die Synthese endlicher Automaten (Moore– und Mealy–Maschinen) auf MS–DOS portiert und an die Xilinx–Software angepaßt. Die Entwicklung weiterer Werkzeuge für dieses System ist geplant.

Um die Programmierung der Sun–Workstations unter der graphischen Oberfläche Sun–View zu vereinfachen, wurde ein Programmgenerator implementiert, der die interaktiv–graphische Entwicklung der Benutzeroberfläche einer Applikation ermöglicht und sodann den Quellcode zur Implementierung dieser Benutzeroberfläche erzeugt. Derartige sog. Interface Builder werden für die Programmierung graphischer Benutzeroberflächen zunehmend eingesetzt.

Die Kontakte zum Institut für Mikroelektronik (IMS) wurden vertieft und eine enge Kooperation zwischen den Instituten vereinbart. Der von Herrn Rust entwickelte Sortierchip zur Echtzeit–Bildverarbeitung wurde vom IMS gefertigt und erwies sich als funktionstüchtig. Der gemeinsame Entwurf und die Fertigung weiterer Chips sind geplant.

2.1.3 Abteilung Dialogsysteme

Leiter	Gunzenhäuser
Wiss. Mitarbeiter	<i>Böcker (bis 30.06), Hanakata, Herczeg, Hohl (ab 1.11.), Knopik, Mahling (ab 1.03.), Maier, Nitsche (ab 1.10.), Ressel (ab 1.09.), Schwab (bis 30.11.), Schweikhardt, Weber (bis 30.06.)</i>
Programmierer	<i>Krepplein, Werner</i>
Verwaltungsangestellte	<i>R. Martin, Wieland</i>
Hilfskräfte	<i>Bayer, Förch, Lorch, Kochanek, Recker, Milan, Nitsche, Ruhland, Sander, Strobel, Wihofsky, Wichert</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 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 wissensbasierten Benutzerschnittstellen insbesondere für Anfragesysteme (Projekt DRUID im Rahmen zweier europäischer RACE–Projekte),
- (2) der Erstellung eines „intelligenten“ Tutors für den Kompositionunterricht (DFG–Forschungsvorhaben),
- (3) rechnerunterstützten Hilfsmitteln für Blinde und Sehbehinderte,
- (4) der Entwicklung und Erprobung moderner Benutzeroberflächen für Blinde und Sehgeschädigte (gemeinsam mit Fa. Papenmeier, Schwerte) sowie
- (5) der Weiterentwicklung von objektorientierten Sprachen zur Wissensrepräsentation.

Das Berichtsjahr war gekennzeichnet durch den Beginn neuer, unter (1), (2) und (4) genannter Vorhaben. Eine Reihe von Publikationen und Präsentationen bezog sich aber noch auf die 1989 erfolgreich abgeschlossenen Vorhaben. So konnten wesentliche Ergebnisse aus der Arbeit der früheren Forschungsgruppe INFORM in neue Vorhaben übernommen und gemeinsam mit der Firma SEL weiterentwickelt werden.

Ein neues von der DFG gefördertes Projekt befaßt sich mit rechnerunterstütztem Lernens im Kompositionssunterricht und wird gemeinsam mit der Musikhochschule Stuttgart durchgeführt.

Wesentliche Ergebnisse der seit 1978 bestehenden Forschungsgruppe „Angewandte Informatik für Blinde“ konnten auch in das mit der Firma Papenmeier begonnene Kooperationsprojekt übernommen werden; hier konnten 1990 insbesondere die Grundlagen für ein europäisches Projekt gelegt werden, das 1991 beginnen kann.

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

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

Auch im Berichtsjahr wurden von der Abteilung Dialogsysteme neue Aktivitäten im Bereich der Informatikausbildung initiiert. Die im WS 89/90 begonnene Lehrveranstaltung „Grundlagen der Informatik“ für die Studiengänge Luft- und Raumfahrttechnik, Physik, Vermessungswesen und andere technische Studiengänge wurde im SS 1990 fortgeführt und im WS 1990/91 erneut abgehalten. Gemeinsam mit der Abteilung „Intelligente Systeme“ wurde im SS 1990 ein Fachpraktikum zur Lehrveranstaltung „Interaktive und intelligente Systeme“ für Studierende im 6. Fachsemester eingeführt.

1.2 Forschungsgruppe DRUID: Projekte MCPR und GUIDANCE

gefördert von der Firma SEL

*(Jürgen Herczeg, Hubertus Hohl (seit 1.11.), Matthias Ressel (seit 1.9.),
Thomas Schwab (bis 30.11.) und studentische Mitarbeiter)*

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

Am Gesamtprojekt sind verschiedene internationale Industriepartner beteiligt. Ziel ist es, schnelle Breitbandkommunikationstechniken zu entwickeln und anhand von konkreten Anwendungssystemen zu demonstrieren. Die Breitbandkommunikation soll zum

Austausch von Multimedia–Information genutzt werden in Form von Texten, Festbildern, Bewegtbildern (Videos) sowie akustische Information (Sprache und Musik). Darüberhinaus sind in das System auch Kommunikationstechniken wie beispielsweise Videokonferenz zu integrieren.

Als Anwendung wird ein Informationssystem zum Anfordern und Inspizieren von Reiseinformation entwickelt. Die zugrundeliegenden Daten sind objektorientiert repräsentiert und bilden ein Netz aus Informationsknoten, die Verweise auf verschiedenartig gespeicherte Multimedia–Informationen enthalten. Beim Abrufen der gewünschten Informationen werden der Ansatz einer formalen Anfrage nach dem Prinzip relationaler Datenbankanfragesprachen und Browsingtechniken, wie sie in Hypertextsystemen verwendet werden, miteinander verknüpft. Die formalen Anfragen dienen dabei im wesentlichen dazu, geeignete Einstiegspunkte für das Browsing zu finden.

Folgende Arbeiten wurden im Berichtsjahr durchgeführt:

Das interaktive Anfragesystem HyperQuery wurde implementiert und in ein vom Projektpartner SEL entwickeltes Hypermedia–Browsingsystem integriert. Die Anfrage wird menüorientiert erstellt und ist inkrementell modifizierbar. Die Implementierungen basieren auf Arbeiten, die im Vorjahr innerhalb des MCPR–Projektes durchgeführt wurden.

Innerhalb des GUIDANCE–Projektes wurde die Benutzerschnittstelle des Anfragesystems ergänzt um eine direkt manipulative, graphikorientierte Komponente, die als Alternative zur textorientierten, formalen Anfrage verwendet werden kann. Die Implementierung dieser Komponente wurde anhand einer von SEL erstellten formalen Benutzerschnittstellenspezifikation durchgeführt.

Die Implementierung der Benutzerschnittstellen erfolgte mit Hilfe des parallel dazu entwickelten objektorientierten Benutzerschnittstellen–Baukastens XIT für das X Window System. Dieser stellt bereits vordefinierte Benutzerschnittstellenobjekte sowie Mechanismen zum Definition neuer Objekte zur Verfügung.

1.3 Forschungsprojekt: Ein intelligenter Tutor für den Kompositionunterricht

gefördert von der Deutschen Forschungsgemeinschaft

*(Heinz-Dieter Böcker (bis 30.06.), Andreas Mahling, Rainer Wehinger
von der Staatl. Hochschule für Musik und Darstellende Kunst, Stuttgart
und studentische Mitarbeiter)*

Die Arbeiten bauen auf einem ebenfalls von der DFG geförderten Kleinprojekt zur Repräsentation musikalischen Wissens aus dem Jahre 1987 auf. In der ersten Projektphase liegt der Schwerpunkt auf den Gebieten der musikalischen Wissensrepräsentation.

tation, der Gestaltung intelligenter computergestützter Kompositionswerkzeuge und deren Einsatz in der Lehre.

Im Rahmen des Musikprojekts wurden 1990 sowohl Systemkomponenten weiterentwickelt, die zum Zeitpunkt des Projektbeginns im März 1990 bereits zur Verfügung standen, als auch neue, das Gesamtsystem ergänzende Prototypen implementiert.

Das gegenwärtige System besteht aus einem Sequenzer, einem Noteneditor, einem Phrasenstruktur-Editor, einer mit Hypertext vergleichbaren Dokumentationskomponente, einer Musikwissensbasis (Objekte des musiktheoretischen Grundwissens und ausgewählte speziellere Kapitel z.B. aus der Harmonielehre) sowie einem Werkzeug für den Entwurf graphischer Editoren für Objektnetze.

Die durchgeführten Arbeiten waren insbesondere darauf ausgerichtet, die verwendete Entwicklungsumgebung (Smalltalk-80 auf Sun 3/60, Atari ST4) möglichst schnell zu einer „Musik–Arbeits– und Lernumgebung“ auszubauen. Hierzu wurde die Implementierung eines MIDI–Sequenzers fortgesetzt, der es erlaubt, den Rechner wie eine Mehrspur–Tonbandmaschine zu benutzen, wobei die tatsächlichen Möglichkeiten dieses Prototyps die Funktionalität und Benutzerfreundlichkeit einer Tonbandmaschine bei weitem übertreffen.

Da für die Workstation eine MIDI–Schnittstelle¹ nicht verfügbar war, erfolgte der Anschluß des aus Projektmitteln beschafften Yamaha Sample–Players über einen Personal Computer (Atari ST4), der mit den erforderlichen Schnittstellen (RS232 von und zur Workstation, MIDI von und zum Synthesizer) ausgerüstet war.

Die Notwendigkeit zumindest einer graphischen Notation zur Unterstützung der musikalisch–kompositorischen Arbeit führte zur Entwicklung eines Noteneditors, zu dessen Besonderheiten es gehört, nicht nur die Bezüge zwischen Notationssymbolen explizit zu repräsentieren, sondern zum Teil auch die musikalische *Bedeutung* von Symbolen bzw. Symbolgruppen (siehe Abb. 2.1).

Da es zu den erklärten Zielen des Projekts gehört, zu zeigen, daß mit *wissensbasierten* Musiksystemen eine qualitativ neue Ebene in der computergestützten musikalischen Arbeit erreicht werden kann, wurde mit dem Entwurf und der Implementierung eines Systems begonnen, das ein (über den Sequenzer) eingespieltes Musikstück selbständig notiert, also eine Partitur erstellt.

Den musikalisch–kompositorischen Aspekten des Projekts wurde durch die Entwicklung mehrerer Prototypen eines Phrasenstruktur–Editors Rechnung getragen. Sowohl das musikalische Material (Daten) als auch die Verarbeitungsvorschriften (Programm) werden in diesem Editor graphisch repräsentiert.

Durch den praktischen Einsatz des Systems in Vorlesungen und Seminaren an der Staatlichen Hochschule für Musik und Darstellende Kunst Stuttgart, konnte die Relevanz der im Projekt entwickelten Ansätze überprüft werden.

¹ Musical Instrument Digital Interface

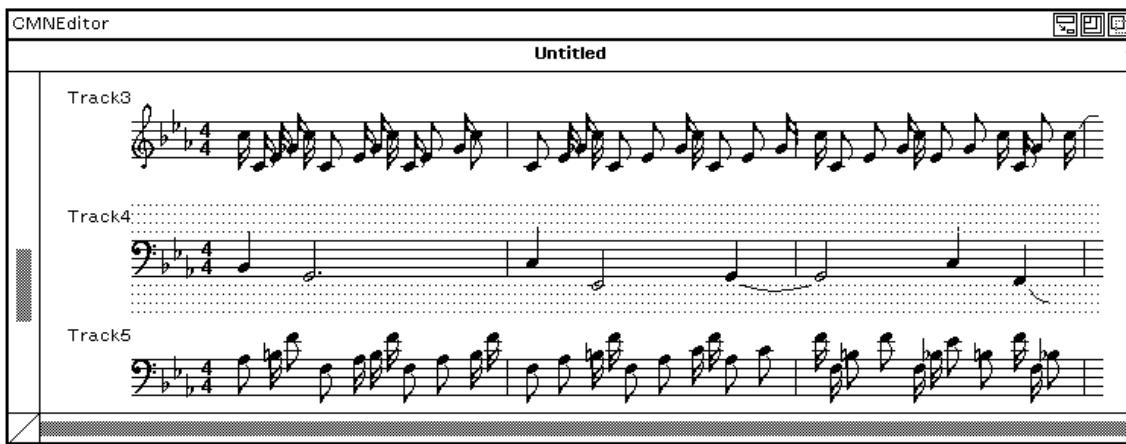


Abbildung 2.1: Der Noteneditor.

1.4 Forschungsgruppe: „Angewandte Informatik für Blinde“

(Waltraud Schweikhardt, Gerhard Weber (bis 30.6.), Alfred Werner und studentische Mitarbeiter, Gastwissenschaftler Endré Pintér aus Ungarn (bis 15.09.))

Im Berichtsjahr hatte die Arbeit der Forschungsgruppe drei Schwerpunkte:

(1) Tastbare Wiedergabe gedruckter Dokumente:

Es wurde ein Verfahren entwickelt und implementiert, mit dem zu einem mit Hilfe eines Scanners erfaßten gedruckten Dokument eine tastbare Orientierungsseite geprägt wird. Homogene Blöcke (Text, Graphik, ...) werden gefunden und mit entsprechenden Punktmustern versehen ausgegeben. Eine solche „Orientierungsseite“ ermöglicht es blinden Lesern, den Inhalt schneller und gezielt zu erfassen.

Schwarz-weiß-Graphiken werden nach einem selbst entwickelten Algorithmus in tastbare Punkt muster abgebildet. Der Algorithmus ist in der Programmiersprache APL2 auf einem Rechner vom Typ IBM PS/2 implementiert. Der Zeichensatz wird in Anlehnung an unsere Punktschrift-Codierung des APL-Zeichensatzes aus dem Jahr 1980 in 8-Punktschrift codiert, so daß APL2 auch von Blinden benutzt werden kann.

(2) Rechnerunterstützte Hilfen für Gehörlose:

Die Arbeiten am Bildlexikon für gehörlose Schüler wurden vorläufig abgeschlossen. Das Programm enthält insbesondere eine Dialogkomponente, die es ermöglicht, Teillexika für spezielle Unterrichtszwecke zu bilden. Das Lexikon wurde an der Paulinenpflege in Winnenden, einem Berufsbildungswerk und Schulen für Gehörlose und Schwerhörige, vorgeführt und zur Erprobung zur Verfügung gestellt.

(3) Verbindung von Zeigebewegungen und Sprachausgabe:

Es wurde eine Interaktionsmethode entwickelt, die es ermöglicht, Wörter und Buchstaben mit einem Zeigeinstrument (hier mit einer Maus) auszuwählen und vorlesen zu lassen.

1.5 Forschungsprojekt: Benutzeroberflächen für Blinde und Sehgeschädigte

gefördert von der Fa. Papenmeier GmbH & Co, KG, Schwerte

(Gerhard Weber (seit 01.09.) und studentische Mitarbeiter)

Textverarbeitungs- und Tabellenkalkulationsprogramme verwenden immer häufiger graphische Benutzeroberflächen. Bisher ist es jedoch nicht möglich, derartige Bildschirminhalte in Braille wiederzugeben, da die Schriftzeichen nur durch ihr Bild und nicht anhand ihrer Codierung wiedergegeben werden.

Das Projekt hat zum Ziel, Interaktionsmethoden zur Textbearbeitung auf der Basis einer Punktschriftanzeige Blinden zugänglich zu machen. Blindenschrift kennt kaum Attribute für einzelne Zeichen, die normalerweise beim Setzen von Texten verwendet werden. (Es ist z.B. nicht möglich, die Zeichen kursiv zu stellen, sie wären unleserlich.) Es werden deshalb Konzepte entwickelt, wie verschiedene Zeichenattribute (z.B. Fettchrift, Farbe) beim interaktiven Erkunden des Bildschirms durch die Gestaltung des Dialogs zugänglich werden.

Diese eher statische Analyse des Bildschirminhalts bildet die Grundlage zur dynamischen Wiedergabe von flächenhaft angeordnetem Text, wie z.B. in Fenstersystemen oder in Textverarbeitungsprogrammen, die gleichzeitig Graphiken und Text darstellen.

Im Berichtsjahr konnten die bisherigen Forschungsergebnisse zur Umsetzung von direkten Interaktionsformen (wie sie in Fenstersystemen verwendet werden) auf mehreren Arbeitstreffen der *Concerted Action on Technology and Blindness* der Europ. Gemeinschaft vorgestellt werden.

1.6 Objektorientiertes Programmieren: Implementierung der objektorientierten Sprache CTALK

(Kenji Hanakata und studentische Mitarbeiter)

Im Berichtsjahr wurde eine Benutzerschnittstelle für die objektorientierte Sprache CTALK auf der Basis von X-Windows entwickelt. Im Gegensatz zu Smalltalk, bei dem die graphischen Komponenten konsequent als Objekte bzw. deren Methoden definiert sind, werden sie in der entworfenen Schnittstelle nur über eine definierte Ebene als „Schnittstellenobjekte“ bzw. „primitive methods“ in CTALK integriert. Die Methoden rufen jeweils direkt eine in C geschriebene Funktion auf, die anderseits auf der

C–Ebene verschiedene Widgets (höhere Funktionseinheiten in X–Windows) bzw. Xlib (Bibliotheksfunktionen) aufrufen. Auf diese Weise werden die Schnittstellen zwischen CTALK und X–Windows klar definiert.

Im wesentlichen bestehen die Schnittstellen aus Browser–Teil und Applikationsteil. Der Browser–Teil erleichtert den Benutzern die sogenannte *Navigation* in der Wissensrepräsentation, während der Applikationsteil ihnen die Werkzeuge für den Entwurf einer anwendungspezifischen visuellen Schnittstelle, wie „Buttons“, „PullDownMenues“, Text– und Graphik–Windows zur Verfügung stellt. Der Browser hat bei einem praktischen Einsatz die gewünschte Mächtigkeit und Leistungsfähigkeit gezeigt.

1.7 Hypertext– und Hypermedien–Systeme

(*Thomas Knopik, Doris Nitsche (seit 1.10.) und studentische Mitarbeiter*)

Im Berichtsjahr wurde ein Informationssystem für größere wissenschaftliche Veranstaltungen entwickelt. Es strukturiert die Gesamtveranstaltung hierarchisch in deren Teilbereiche, Sitzungen in den Teilbereichen sowie einzelne Vorträge. Das System beinhaltet Informationen über Zeit und Ort des Ereignisses sowie die dazu in Relation stehenden Räumlichkeiten, Vortragenden und Themen. Als Implementationsumgebung wurde HyperCard auf einem Macintosh gewählt. Dadurch stand für den Benutzer eine leicht zu erlernende und bedienende Benutzerschnittstelle zur Verfügung. Sie ermöglicht es dem Benutzer, mit direkter Manipulation das System entlang der Hierarchiepfade, der Relationen, alphabetischer Listen und eines Zeitplans zu durchwandern. Außerdem hat er die Möglichkeit, nach bestimmten Personen und Themen zu suchen.

Parallel dazu wurde ein Autorensystem entwickelt, mit dem die Informationssysteme über verschiedene Veranstaltungen generiert werden können. Dabei liegt die Grundstruktur und das Funktionalitätsspektrum fest. Der Autor baut mit Hilfe des Autorensystems die Hierarchie der Ereignisse auf. Dabei wird er vom System dahingehend unterstützt, daß es die Raum–, Personen– und Themenkarten automatisch erzeugt und die bereits bekannten Informationen darauf überträgt. Zusätzlich generiert das System alphabetisch geordnete Listen der Veranstaltungen, Personen, Themen und Räume sowie einen chronologischen Zeitplan. Die Konsistenz der Information auf den verschiedenen Karten und Listen wird bei Änderungen automatisch aufrechterhalten.

Das Informations– und Autorensystem wurde anhand der 20. Jahrestagung der Gesellschaft für Informatik in Stuttgart erprobt.

1.8 Studie: Entwicklung und Einsatz von qualitativ hochwertiger Lernsoftware auf dem Gebiet der Informatik

gefördert von der Firma IBM

(Rul Gunzenhäuser, Dieter Maier)

Ziel der im Rahmen eines Kooperationsvertrag mit IBM durchgeführten Arbeiten ist die Förderung der Entwicklung und des Einsatzes von qualitativ hochwertiger Lernsoftware auf dem Gebiet der Informatik. Durch die Studie soll die Arbeit der Softwaregruppe „Informatik“ der Akademischen Softwarekooperation (ASK) unterstützt werden.

Im einzelnen werden in der Studie der Einsatzzweck von Lernsoftware für die Ausbildung in Informatik geklärt, Kriterien für die Bewertung erarbeitet, sowie eine Bestandsaufnahme und Bewertung bestehender Lernsoftware im Fachbereich Informatik durchgeführt.

Weiterführende Konzepte zur Entwicklung von Lernsoftware und Initiativen zur Förderung der Erstellung von fachspezifischer Software für die Ausbildung sollen erarbeitet werden.

Weitere Aktivitäten im Rahmen der Studie waren die Mitwirkung bei der Begutachtung der Lernsoftware im Rahmen der ASK Hochschulsoftwarepreise und Empfehlungen für das Einstellen von referierter Informatik–Lernsoftware in die ASK–Datenbank SISY.

2. Informatikausbildung

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

Als Vorsitzender des Fakultätentages Informatik an Hochschulen der Bundesrepublik hatte er mannigfache Aufgabenstellungen der Informatikausbildung und –weiterbildung initiativ zu verfolgen und in unterschiedlichen Gremien — insbesondere der Hochschulrektorenkonferenz (HRK) — zu vertreten. Im Berichtsjahr konzentrierten sich die Arbeiten insbesondere im Entwurf einer neuen Rahmenprüfungsordnung für den Diplomstudiengang Informatik an wissenschaftlichen Hochschulen sowie auf die Integration von wissenschaftlichen Hochschulen im Gebiet der früheren DDR in die Bundesrepublik. Bei mehrerer Vortrags- und Studienreise des Abteilungsleiters zu den Hochschulen in den jetzigen neuen Bundesländern wurden Kontakte geknüpft sowie unterschiedliche Förderungsmaßnahmen für die Informatikausbildung initiiert und begutachtet.

Im Berichtsjahr war Herr Gunzenhäuser Mitglied mehrerer Beratungsgremien an den Universitäten Hagen (Thema: Forschungsvorhaben im Bereich des rechnerunterstützten Lernens), Karlsruhe (Themen: Ausbildung blinder Informatikstudenten und Akademische Software Kooperation) und Tübingen (Lehrerfortbildung in Informatik).

Gemeinsam mit Prof. Mandl (Tübingen und München) wurde zum vierten Mal eine Fachtagung über „Intelligente Lernsysteme“ vorbereitet und durchgeführt. Diese Tagung wurde im Rahmen der 20. Jahrestagung der Gesellschaft für Informatik e.V. abgehalten; an ihr nahmen mehr als 80 Fachkollegen aus der Informatik, der Linguistik und der Kognitiven Psychologie teil.

Forschungskontakte

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

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

Deutsches Taubblindenvwerk Hannover (*F. Zekel*)

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

Firma F. H. Papenmeier GmbH & Co.KG Schwerte (*J. Bornschein*)

Forschungsinstitut für Anwendungsorientierte Wissensverarbeitung an der
Universität Ulm (FAW)
(*Prof. W. Radermacher, Dr. D. Rösner, Dr. W.-F. Riekert*)

Freie Universität Berlin, Institut für Informatik (*Prof. Strothotte*)

Fujitsu Laboratories Kawasaki, Japan (*Prof. T. Hayashi, Prof. Uchida*)

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

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

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

IBM Deutschland GmbH, Wissenschaftliches Zentrum Heidelberg
(*Dr. A. Blaser, W. Wernecke, Dr. Th. Strothotte*)

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

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

Institut für Informatik der Universität Karlsruhe
(*Prof. A. Schmitt, Modellversuch Informatik für Blinde*)

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

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

-
- International Association of Pattern Recognition (*Prof. T. Pavlidis*)
METEC Ingenieur GmbH, Medizintechnik - Feinwerktechnik, Stuttgart
Overbrook School for the Blind, International Program, Philadelphia, Pa. USA
(*Frau A. Rimrott*)
SEL Forschungszentrum Pforzheim
(*A. Lesch, R. Weiß und wissenschaftliche Mitarbeiter*)
TA Triumph Adler AG, Forschungszentrum Nürnberg
(*Dr. R. Lutze und wiss. Mitarbeiter*)
University of Colorado at Boulder, Dep. of Computer Science
(*Prof. G. Fischer, Prof. C. Kintsch, Dr. A. Lemke*)
University of Kyoto, Japan (*Prof. N. Nagao, Prof. Nakamura, Prof. Tsujii*)
University of Western Ontario, Dep. of Computer Science, London, Kanada
(*Prof. H. Jürgensen*)
Uppsala Universitet, Matematiska Institutionen (*Prof. S. Christofferson*)
Uppsala Universitet, Institutionen för Teknisk Databehandling (*Prof. Sjöberg*)
-

2.1.4 Abteilung Intelligente Systeme

Leiter	<i>Lehmann</i>
Mitarbeiter	<i>Burkert, Castro, Emele, Forster, Heid (IMS), Langjahr, Momma (IMS), Rathke, Schullerer (ab 13.9.), Tausend, Yoshimoto (IMS), Zajac (IMS)</i>

Übersicht

(*Egbert Lehmann*)

Das Forschungsinteresse der Abteilung liegt im Überlappungsbereich von inhaltsorientierter Sprachverarbeitung, Repräsentation von Wissen und maschineller Lernfähigkeit. Angestrebgt wird langfristig die Entwicklung eines prototypischen Systems zum Einlesen und Analysieren von deutschsprachigen Texten unterschiedlichen Inhalts, das wesentliche Elemente des Inhalts (Wortbedeutungen und Sachverhalte) zu erkennen und in einer generelle Wissensbasis zu speichern gestattet. Die nach generellen Prinzipien repräsentierten Sachverhalte sollen in einem natürlichsprachlichen Mensch–Maschine–Dialog abfragbar sein. Die sprachliche und sachliche Kompetenz soll durch Einbeziehung von Methoden zum maschinellen Lernen im Laufe der Zeit erweitert werden (Das System soll Texte lesen und daraus lernen können.).

Als erste Komponenten dieses Systems wurden ein Textscanner, ein maschinelles Wörterbuch, eine Regelgrammatik sowie Programmpakete für die morphologische und syntaktische Analyse in Common–Lisp entwickelt (Lehmann, Burkert). Das Wörterbuch berücksichtigt explizit einen Grundwortschatz von z.Zt. etwa 5000 besonders wichtigen deutschen Wörtern, im Zusammenwirken mit einem System für die morphologische Analyse deutscher Wortformen (bei Berücksichtigung von Flexion, Deviation und Komposition) kann damit eine hohe Textüberdeckung erreicht werden. Die syntaktische Analyse erfolgt mittels eines Chart–Parsers und einer Konstituentengrammatik und wird durch Feature–Unifikation ergänzt.

Induktive, analoge oder erklärbungsbaerte Verfahren des maschinellen Lernens werden auf ihre Brauchbarkeit in diesem Bereich untersucht (z.B. zum Erlernen unbekannter Worte, Grammatikregeln, Bedeutungsdefinitionen aus Textbeispielen) und weiterentwickelt (Tausend, Lehmann).

Gleichzeitig wurde an der Entwicklung von Formalismen und Implementationsprinzipien für die Repräsentation von Wissen gearbeitet (Rathke, Forster). Das hier entstehende System Frame–Talk soll auf der Basis von CLOS Möglichkeiten zur Repräsentation hierarchisch angeordneter Frames, von Slotbeschreibungen, Regeln, constraints und Perspektiven enthalten. Aus dem Studium von Begriffsdefinitionen in einem Bedeutungswörterbuch wurden Anforderungen an einen terminologischen Repräsentationsformalismus mit klar definierter modelltheoretischer Semantik abgeleitet und erste Implementierungsversuche unternommen (Forster, Burkert).

Im Rahmen des vom BMFT geförderten, gemeinsam mit dem Institut für Maschinelle Sprachverarbeitung bearbeiteten Projektes POLYGLOSS (Mitarbeiter: Emele, Momma, Zajac, Heid, Yoshimoto) zur Entwicklung eines multilingualen Sprachgenerators wurde ein Formalismus zur Beschreibung von typisierten Merkmalsstrukturen entwickelt, der über eine klare Semantik verfügt und die Beschreibung linguistischer Fakten in modularer Weise unterstützt. Da Grammatikbeschreibungen in der Form von Wohlgeformtheits-Constraints ausgedrückt werden, ist dieser Formalismus inhaltungsfrei, d.h. er kann für Sprachgenerierung ebenso wie für Sprachanalyse eingesetzt werden.

Ein Verarbeitungssystem für diesen Formalismus wurde in Common Lisp implementiert und mit verschiedenen Grammatikbeschreibungen (unter besonderer Beachtung von HPSG) sowie deutschen, englischen und französischen Sprachdaten erfolgreich getestet. Auf der Internationalen Konferenz für Computerlinguistik COLING-90 sowie auf einem vom Projektteam veranstalteten Workshop im November 1990 wurden die erzielten Forschungsergebnisse in mehreren Fachvorträgen und Programmdemonstrationen der internationalen Fachöffentlichkeit erfolgreich vorgestellt.

Als anwendungsnahe Ergebnis in Richtung Expertensysteme entstand ein grafikunterstütztes wissensbasiertes Konfigurierungssystem für CNC-Maschinen, das zu einem interaktiven Problemlöser für unterschiedliche Konfigurierungsaufgaben verallgemeinert werden soll (Rathke).

Extraktion und Repräsentation von Wortbedeutungen

(Gerrit Burkert, Peter Forster)

Für die Sprachverarbeitung in realen Anwendungsbereichen wird ein umfangreiches Lexikon (zehn- bis hunderttausend Einträge) benötigt. Neben Aspekten der Organisation des Lexikons spielt die Frage der Repräsentation des lexikalischen Wissens eine zentrale Rolle. Ein weitgehend ungelöstes Problem ist die Repräsentation von semantischem Wissen. Da ein Lexikon mit dem genannten Umfang kaum von Hand erstellbar ist, werden Möglichkeiten zur automatischen Extraktion — beispielweise aus maschinenlesbaren Wörterbüchern — untersucht.

Ansätze zur Repräsentation semantischen Wissens basieren i.a. auf Begriffshierarchien in Form von semantischen Netzen bzw. Frames. Bei terminologischen Repräsentationsformalismen liegt der Schwerpunkt auf der Repräsentation von definitorischem Wissen. Terminologische Repräsentationsformalismen zeichnen sich durch eine klar definierte modelltheoretische Semantik aus. Durch Inferenzmechanismen kann Wissen, das in Definitionen nur implizit enthalten ist, abgeleitet werden. Ausgehend von Definitionen im Duden Bedeutungswörterbuch wurde die Eignung terminologischer Repräsentationsformalismen zur Repräsentation semantischen Wissens untersucht. Es ergab sich die Notwendigkeit zur Erweiterung des terminologischen Formalismus, was allerdings zur Nichtentscheidbarkeit terminologischer Inferenzen (Klassifikation) führt.

Zunächst wurde ein terminologisches „Kernsystem“ mit wohldefinierter Semantik und

effizienten Inferenzverfahren entworfen und ein erster Prototyp in CLOS (Common Lisp Object System) implementiert. Erweiterungen dieser terminologischen Komponente sind geplant. Im Rahmen einer Studienarbeit wird zur Zeit eine Komponente zur Repräsentation assertionalen Wissens entwickelt. Diese Komponente stellt Routinen zur Verfügung, die auf der Wissensbasis operieren und Inferenzen der terminologischen Komponente verwenden. Da inzwischen maschinenlesbare Wörterbücher (z.B. Oxford Advanced Learner's Dictionary, OALD) verfügbar sind, erscheinen Versuche sinnvoll, einen Teil des benötigten lexikalischen Wissens aus solchen Wörterbüchern zu extrahieren. Die Extraktion semantischen Wissens aus Definitionstexten ist bisher nur in Ansätzen möglich. Zur Analyse der Definitionstexte wurden inzwischen ein Textscanner, eine morphologische Komponente sowie ein Chart Parser entwickelt. Im nächsten Schritt wird versucht, für eine Reihe einfacher aufgebauter Definitionen die Zielrepräsentation automatisch zu erzeugen.

Konex+

(Christian Rathke)

Aufbauend auf einem Expertensystem, das im Rahmen einer Diplomarbeit auf der Basis von CLOS und CLX entwickelt worden war, wurden wesentliche Erweiterungen durchgeführt. 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.

Der Konfigurierer wurde unter Verwendung von dem in der Abteilung Dialogsysteme entwickelten Benutzerschnittstellenbaukasten XIT zu einer wissensbasierten Designumgebung erweitert. Die Konfigurationsaufgabe wird dabei als interaktiver Problemlöseprozeß aufgefaßt, wobei das System dem Benutzer mit Vorschlägen und Kritik hilft, das Konfigurationsproblem zu lösen.

In zukünftige Arbeiten 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.

Slots in FrameTalk sind zusätzlich mit Hilfe von Defaults, Initialisierungen, Triggers und Restriktionen beschrieben. Sie bestimmen dadurch einen weiteren Teil der Slotsemantik. In FrameTalk werden Slotbeschreibungen in CLOS–Methoden umgesetzt. Der Umsetzungsprozeß ist als Eigenschaft und Verhalten von Meta–Objekten realisiert. Damit können neue oder modifizierte Slotbeschreibungen objektorientiert, d.h. als Erweiterung bestehender Funktionalität, implementiert werden.

Slotbeschreibungen erlauben die Klassifikation von Frames gemäß der Subsumptionsrelation. Die *Subsumptionsrelation* ist (partiell) definiert zwischen zwei Konzeptbeschreibungen in einer terminologischen Wissensbasis. Für alle Konzepte beschreibt sie eine Taxonomie. Das Subsumptionsprotokoll ist gemäß der objektorientierten Ausrichtung als eine Reihe von generischen Funktionen realisiert, die die Subsumptionsbeziehung für Konzepte, Slotbeschreibungen und einzelne Elemente der Slotbeschreibung feststellen.

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

Constraints sind Restriktionen zwischen den Werten verschiedener Slots eines Objekts. Constraints beinhalten eine „aktive“ Komponente, die bestimmt, was im Falle einer Constraintsverletzung zu tun ist, um einen konsistenten Zustand herzustellen. Ein Constraintssystem wurde auf CLOS übertragen und mit Hilfe von XIT an ein prototypisches Formularsystem angebunden.

Analogen Lernen

(Birgit Tausend)

Beim analogen Schließen wird versucht, bereits vorhandenes Wissen, z.B. Problemlösungen oder Konzepte, zu nutzen, um ähnliche Probleme zu lösen oder ähnliche Konzepte zu definieren. Der Vorgang des analogen Schließens ist sehr komplex. Besonders bei der Suche nach einem bekannten, dem zu lösenden ähnlichen Problem im Raum aller bekannten Probleme oder bei der Übertragung von Teilen des bekannten Problems auf das neue Problem sind die Suchräume sehr groß. Diese großen Suchräume können beispielsweise dadurch beschränkt werden, daß analoge Hinweise der Form „A ist ähnlich zu B“ oder spezielle, auf die Anwendung zugeschnittene Ähnlichkeitsmaße angewendet werden.

Als Prototyp eines Systems, das seine Wissensbasis durch Analogiebildung erweitert, wurde ein System entwickelt, das die Definition logischer Prädikate lernt. Ausgangspunkt für das entwickelte Lernverfahren ist eine Wissensbasis mit bekannten logischen

Prädikaten. Mit Hilfe dieser Definitionen und einem analogen Hinweis ist es möglich, neue logische Prädikate zu definieren.

Die Verwendung von analogen Hinweisen erleichtert zwar die Suche nach ähnlichen bekannten Prädikaten, ein eingehender Vergleich der beiden Prädikate ist aber dennoch notwendig, will man eine korrekte Übertragung des Wissens gewährleisten. Hier ist die Entwicklung geeigneter Ähnlichkeitmaße notwendig. Zur Lösung dieses Problems wurde mit einer Untersuchung über den Einsatz von Ähnlichkeitsmaßen beim Analogiernen begonnen.

Darüberhinaus soll das entwickelte Verfahren auch in anderen Anwendungsbereichen eingesetzt werden. Angestrebt wird die Integration des entwickelten Verfahrens in ein natürlichsprachliches System, dessen sprachliche Fähigkeiten sich im Laufe des Zeit auf diese Weise verbessern sollen. Mit geringen Modifikationen könnte das Verfahren z.B. auf das Lernen ähnlicher Grammatikregeln bei DCG's, angewendet werden.

Unabhängig von diesem Lernverfahren wird auch untersucht, wie die Grammatik eines Chart-Parsers durch Analogiebildung erweitert werden kann, wenn sie nicht mächtig genug ist, um alle eingegebenen Sätze zu parsen. Eine wichtige Rolle spielt dabei die Information über erfolgreich geparsete Teile des Satzes, die die Chart auch dann enthält, wenn Fehler beim Parsen des gesamten Satzes aufgetreten sind. Abhängig von diesem Kontext sollen bekannte Regeln so modifiziert werden, daß der eingegebene Text vollständig verarbeitet werden kann, die Regeln aber nicht übergeneralisiert werden.

Polygloss: Entwicklung eines universell einsetzbaren multilingualen Textgenerators

*(Martin C. Emele, Ulrich Heid, Stefan Momma, Kei Yoshimoto,
Rémi Zajac)*

Im Rahmen der Forschungstätigkeiten des Projekts POLYGLOSS wurden im Berichtszeitraum die folgenden Ergebnisse erzielt:

- es wurde ein Formalismus zur Beschreibung von typisierten Merkmalsstrukturen entwickelt, der über eine klare Semantik verfügt und die Beschreibung linguistischer Fakten in modularer Weise unterstützt; da Grammatikbeschreibungen in Form von Wohlgeformtheits-Constraints ausgedrückt werden, ist der Formalismus inhärent richtungsfrei und kann für Analyse und Generierung gleichermaßen eingesetzt werden.
- der Formalismus wurde in Common Lisp implementiert und für die Verarbeitung von Beschreibungen, wie sie in der Grammatikbeschreibung häufig vorkommen, optimiert.
- auf der Beschreibungsseite wurden verschiedene Ansätze zur Repräsentation von Domänenwissen untersucht und auf ihre Adäquatheit für den Einsatz in einem ganzheitlichen Repräsentationsformalismus geprüft. Ergebnis dieser Unter-

suchung ist die Entscheidung für eine Modellierung nach dem Vorbild des “Upper Model” des PENMAN Projekts.

- auf dem Gebiet der Grammatikbeschreibungen wurde der neu entwickelte Formalismus und seine Implementierung mit unterschiedlichen fragmentarischen Beschreibungen in verschiedenen linguistischen Formalismen getestet. Besonderes Gewicht hatten dabei Beschreibungen im Formalismus der HPSG, da dieser Formalismus eine typisierte Darstellung von der Theorie her gut unterstützt.
- eine umfangreiche Beschreibung von Daten aus dem Teilproblemberich der Subkategorisierungseigenschaften französischer Verben wurde durchgeführt, um die Formulierbarkeit von linguistisch motivierten Generalisierungen zu untersuchen. Solche Generalisierungen waren in den seither existierenden Formalismen nicht direkt innerhalb der Beschreibungssprachen ausdrückbar und erforderten häufig externe Zusatzverarbeitungsschritte, die die Transparenz des Verarbeitungsprozesses beeinträchtigten.
- am Beispiel deutscher, französischer und englischer Sprachdaten wurde die Interaktion von Wohlgeformtheitsbedingungen auf verschiedenen linguistischen Beschreibungsebenen untersucht und gezeigt, daß sich diese Interaktion durch den allgemeinen constraint-basierten Ansatz, den der von uns entwickelte Formalismus verfolgt, elegant beschreiben und effizient verarbeiten läßt.
- eine umfangreichere Beschreibung wurde im Rahmen der Forschungskooperation mit ATR und NTT für das Japanische erstellt, die auf der Grammatik, die bei ATR entwickelt wurde, aufbaut. Eine Beschreibung für das Deutsche steht noch aus.

Forschungskontakte

Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI)

(*Barth, Wahlster, Richter*)

Forschungsinstitut für Anwendungsorientierte Wissensverarbeitung (FAW), Ulm

(*Rösner*)

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

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

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

Polnische Akademie der Wissenschaften, Warschau (*Bolc*)

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

IBH, Schieberdingen (*Hammer*)

Battelle Institut, Frankfurt (*Haug*)

ATR, Kyoto, Japan (*Kurematsu*)

Carnegie-Mellon-University, ICMT, Pittsburgh, USA

(*Carbonell, Levine, Nirenburg, Nyberg*)

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*)

2.1.5 Abteilung Kombinatorik und Numerische Verfahren

Leiter	<i>Knödel</i>
Mitarbeiter	<i>Berger, Ebinger, Eusterbrock (bis Juli), Hruschka (bis Juni), Liedtke (ab Juni), Mijderwijk, Photien (ab Oktober)</i>
Hilfskräfte	<i>Bertol, Eberhardt, Fritz, Hölzlein, Weber</i>

Zufallsorakel und probabilistische Modelle

(Werner Ebinger)

Eine Orakelmaschine rechnet „relativ“ zu einer Orakelsprache. Ein zufällig gewähltes Orakel heißt Zufallsorakel. Die Menge der effizient lösbarer Probleme P mit Zufallsorakel ist gleich der probabilistischen Komplexitätsklasse BPP (Bennett, Gill). Hier wurde nach weiteren ähnlichen Zusammenhängen gesucht.

Ein weiteres probabilistisches Berechnungsmodell sind interaktive Beweissysteme. Ende 1989 wurde durch internationale Zusammenarbeit über elektronische Post ein Durchbruch erzielt ($IP = PSPACE$). Dieser Durchbruch regte Forschungen über andere Charakterisierungen von interaktiven Beweissystemen an.

Methoden der Programmverifikation

(Thomas Liedtke)

Unter dem Aspekt, praktikable Methoden zur Verifizierung von realen prozeduralen Programmen zu finden, werden mehrere „konventionelle“ Methoden auf ihre Automatisierbarkeit hin untersucht. Es werden dabei gänzlich verschiedene Vorgehensweisen untersucht. So z.B. *inductive assertion*, *subgoal induction*, *intermittend assertion*, prädikatenlogische, flußdiagramm- und schemabezogene. Auch Methoden graphentheoretischen Ursprungs wie z.B. relationalgebraische Methoden kommen in Betracht.

Anstrebenswerte Ziele sind u.a.:

- real existierende Programme, so wie sie auf natürliche Art geschrieben werden (i.G. zu Lehrbuchbeispielen), zu verifizieren.
- lange Programme mit möglichst geringer Benutzerinteraktion zu verifizieren.
- geeignete Benutzerumgebungen zu definieren
- verschiedene Methoden kombiniert programmsensitiv zu verwenden

-
- Aussagen über totale Korrektheit möglichst gut zu approximieren

Geplant ist u.a. auch der Einsatz von bereits existierenden interaktiven Umgebungen für Induktionsbeweise über algebraische Strukturen.

Separation der Eigenwerte von Matrizenpolynomen

(*Sven Mijderwijk*)

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

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

Echtzeitdatenverarbeitung auf dem PC

(*Uwe Berger*)

Dieses Forschungsprojekt wird in Zusammenarbeit mit der Abteilung Betriebsssoftware durchgeführt und ist in Abschnitt 2.1.1, Seite 12, beschrieben.

Forschungskontakte

Institut für Informatik der Universität Innsbruck (*R. Albrecht*)

Universität Clausthal (*B. Quatember*)

Universität für Bodenkultur Wien (*K. Prachar*)

Universität Ulm (*Köbler, Schöning, Thierauf*)

Universität Würzburg (*Buntrock, Hertrampf, Wagner*)

Universität Leipzig (*G. Heinig*)

2.1.6 Abteilung Programmiersprachen und ihre Übersetzer

Leiter	Göttler (<i>Lehrstuhlvertretung bis 31.3.</i>), Ludewig (<i>kommissarisch ab 1.4.</i>)
Mitarbeiter	Günthör, Bräunl (<i>bis 31.7.</i>), Kübler, Schusser (<i>bis 30.4.</i>), Welsch, Zell (<i>bis 31.1.</i>)

Mit dem Wintersemester 1989/90 endete die Lehrstuhlvertretung durch Herrn Dr. H. Göttler, Universität Erlangen. In dieser Zeit hat er zwei Projekte aus Erlangen auch in Stuttgart weiterverfolgt: JULIA, ein Programm zur Realisierung eines juristischen Literaturarchivs, und PAGG, ein System zur Editor-Entwicklung auf der Grundlage von Graphgrammatiken. Die beiden Projekte sind im Jahresbericht 1989 ausführlich beschrieben.

Auch die Arbeiten von Herrn Dr. A. Zell wurden bis einschließlich März 1990 noch in dieser Abteilung durchgeführt; sie laufen jetzt im IPVR und sind dort dokumentiert.

Integration von logischer und objektorientierter Programmierung (*Christoph Welsch*)

Die Gefahr einer solchen Integration besteht einerseits darin, daß die Akkumulation der verschiedenen Konzepte zu einer kritischen Komplexitätszunahme führt, welche die erhofften Vorteile zunichte macht; andererseits, daß durch die willkürliche Kombination von Konzepten ursprüngliche Stärken eines Programmierparadigmas verloren gehen.

Ziel dieses Projekts war deswegen eine einfache und homogene Verbindung der jeweiligen Programmierkonzepte. Zunächst wurde untersucht, inwieweit diese miteinander verträglich sind. Es zeigte sich, daß die Konzepte Klasse, Vererbung und Nachricht ohne weiteres mit der logischen Programmierung in Einklang zu bringen sind, das Konzept des Objekts aber nur mit Einschränkungen. Die Tatsache, daß Objekte veränderlichen Zustand besitzen, führt zu ernsten Konflikten mit der mathematischen Logik.

Ausgehend von diesen Überlegungen wurden verschiedene Ansätze zur Integration objektorientierter Konzepte in die logische Programmierung untersucht, wobei der Aspekt der Zustandsänderung zunächst ausgeklammert blieb. Vererbung wurde unter drei Aspekten gesehen: Als Spezialisierungsstruktur, als Teilmengenbeziehung und als Hilfsmittel zur Code-Wiederverwendung. Daraus ergaben sich drei verschiedene Interpretationen für Klassen: Eine Klasse ist Objektbeschreibung, Objektmenge und Programmbaustein.

PARALLAXIS

(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.

Das Projekt ist im August 1990 mit dem Wechsel von Herrn Bräunl an das IPVR gegangen.

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 Teilkasse 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 wurden zunächst existierende Metaebenen–Architekturen beschrieben und nach der zugrundeliegenden Wissensrepräsentationsmethode klassifiziert. Genauer analysiert wurden logische Metaebenen–Architekturen. Ausgehend von Konzepten und einfachen Metainterpretern wurden Iterative–Deepening Prolog und zeitbeschränktes logisches Programmieren theoretisch und praktisch untersucht.

Die Fähigkeit der Reflexion über das eigene Schließen steht in engem Zusammenhang mit der Fähigkeit, sinnvolle Schlußfolgerungen über das Wissen und das Verhalten anderer intelligenter Agenten (Personen oder Systeme) durchführen zu können. Dafür wurden zuerst die Konzepte auf mehrere Agenten erweitert, und theoretische Aussagen zum “reasoning about knowledge” hergeleitet. Schließlich wurde ein Ansatz beschrieben, der Handlungen, Sensoren und Effektoren in das Gesamtkonzept integriert und als logische Basis für teil autonome Systeme dienen kann.

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

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 wurden insgesamt 3 Simulatoren für neuronale Netze entwickelt: NetSim, ein Simulator für Sun Workstations (SunView, C) unter Unix, NeuroSim, ein Macintosh–basierter Simulator (Mac Toolbox, Pascal), sowie ab Frühjahr 1990 SNNS (Stuttgarter Neuronale Netze Simulator). Als Nachfolger von NetSim ist SNNS der anspruchsvollste und effizienteste dieser Simulatoren (in C und X–Windows implementiert) für eine große Klasse von Unix Workstations. SNNS 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 Beschreibungssprache, Nessus, erlaubt die Beschreibung komplexer neuronaler Netze auf eine einfache Weise aus einer höheren Beschreibungssprache.

Die Arbeiten an SNNS werden im Rahmen eines größeren Projektes derzeit am Institut für Parallele und Verteilte Höchstleistungsrechner fortgeführt, wobei SNNS parallelisiert werden und die Basis eines massiv parallelen Simulationssystems für neuronale Netze bilden soll. Dazu steht seit kurzem am IPVR ein Rechner mit SIMD–Architektur (MasPar MP 1216) mit über 16.000 Prozessoren zur Verfügung.

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. Bukys, 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*)

2.1.7 Abteilung Software Engineering

Leiter	<i>J. Ludewig</i>
Wiss. Mitarbeiter	<i>H. Lichter, M. Deininger, K. Schneider, J. Schwille (ab 1.3.)</i>
Programmierer	<i>M. Schneider, A. Georgescu (ab 17.4.)</i>
Verwaltungsangestellte	<i>Frau U. Günthör</i>

Projekt: Objektorientierte Software–Entwicklung und Prototyping

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

Im Rahmen dieser Arbeit wurde ein Kernsystem entwickelt, mit dem Software–Architekturen interaktiv erstellt und verwaltet werden können. Begleitend dazu wurde ein Modell für Software–Architekturen entwickelt, das sowohl die Vorteile des objekt–orientierten Ansatzes als auch notwendige Sicherheitsaspekte berücksichtigt.

Projekt: Software–Prozeß–Modell

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

Im Juni 1990 (Tag der offenen Tür) wurde der zweite Prototyp vorgestellt; dieser war in Smalltalk–80 implementiert und nur zur Klärung konzeptioneller Fragen bestimmt. Im Herbst 1990 wurde die Entwicklung einer ersten Version von SESAM begonnen.

Forschungskontakte

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

ETH Zürich (*N. Wirth, C.A. Zehnder*)

Universität Mainz (*F. Wankmüller*)

Zahlreiche Anbieter von Software-Werkzeugen (CASE-Tools)

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

2.1.8 Abteilung Theorie der Informatik

Leiter	Diekert (<i>Lehrstuhlvertretung bis 31.3.90</i>), <i>Gunzenhäuser (kommissarisch vom 1.4.90 bis 30.9.90)</i> , <i>Knödel (kommissarisch ab 1.10.90)</i> ,
Mitarbeiter	<i>Münchow, Reinhardt (beurlaubt vom 1.5.90 bis 31.7.90), Reuß</i>
Gastprofessoren	<i>Schmidt (vom 23.4.90 bis 31.8.90), Posthoff (ab 22.10.90)</i>
Hilfskräfte	<i>Drehmann, Ehlert, Faulstich, Huppenbauer, Jahke, Jung, Ketelhut, Nguyen, Räumschlüssel, Schedel, Schumacher, Stahl, Weberruß, Zink</i>

Forschungsprojekt: Konfluente Semikommation und Komplexität

(Volker Diekert, Klaus Reinhardt)

Um Umformungen auf nebenläufigen Prozessen beschreiben zu können, werden Spur-Ersetzungssysteme als Verallgemeinerung von Semi–Thue–Systemen verwendet. Wichtig bei solchen Ersetzungssystemen ist das Wortproblem, d.h. man möchte entscheiden, ob zwei Wörter äquivalent sind. Dazu betrachtet man die Eigenschaften *noethersch* und *konfluent*, die es ermöglichen eine irreduzible Normalform zu berechnen. Wir betrachten endliche Semikommationssysteme als Unterklasse von Spur–Ersetzungssystemen und konnten zeigen, daß deren Konfluenz nun entscheidbar ist, indem wir ein graphentheoretisches Kriterium entwickelten. Wir konnten zeigen, daß sich die Existenz endlicher, vollständiger Präsentationen zwischen Spurmonoiden auf die Existenz eines vollständigen Semikommationssystems reduzieren läßt. Ferner konnten wir durch eine Graphkonstruktion mit Hilfe des graphentheoretischen Kriteriums zeigen, daß das Problem der Existenz endlicher, vollständiger Präsentationen Σ_2^p –vollständig ist und die Entscheidung der Konfluenz eines gegebenen Semikommationssystems $co-NP$ vollständig ist. Dadurch wurde auch ein offenes Problem aus einer früheren Arbeit von Diekert und Vogler gelöst.

Forschungsprojekt: Charakterisierung von Komplexitätsklassen durch Erweiterungen des Nichtdeterminismusbegriffs und formalsprachliche Methoden

(Klaus Reinhart)

Das nichtdeterministische Erkennen von Sprachen entspricht der Frage nach der Existenz eines akzeptierenden Berechnungspfades eines Automaten. Dieses Modell kann erweitert werden, indem man ein Abwechseln von existentieller und universeller Quantifizierung ermöglicht. In diesem Fall werden *alternierende* Automaten verwendet. Eine andere Möglichkeit ist, die Anzahl der akzeptierenden Berechnungspfade einer nichtdeterministischen Maschine zu betrachten, dies führte zur Definition der Zählklassen.

Ausgehend von Alternierungsmodellen von W. Ruzzo, J. Simon und M. Tompa und R. Ladner und N. Lynch wurde von K.J. Lange das Konzept der leeren Alternierung entwickelt. Die Alternierung wird dabei dadurch eingeschränkt, daß der Keller bzw. ein nicht logarithmisch platzbeschränktes Band beim Alternieren leer sein muß. Damit haben wir folgende Ergebnisse:

Mit einer leer alternierenden polynomiell zeitbeschränkten Turingmaschine wird die von Wagner definierte Orakelkomplexitätsklasse $\Theta_2^p := L^{NP}$, in der viele Optimierungsprobleme liegen, charakterisiert. (Volle Alternierung erreicht hier *PSPACE*.)

Mit einem leer alternierenden polynomiell zeitbeschränkten Kellerautomat mit logarithmisch platzbeschränktem Band wird bei beliebiger Alternierung P , bei konstanter Alternierung *LOGCFL* und bei \log^k Alternierung die Schaltkreiskomplexitätsklasse AC^k charakterisiert. (Volle Alternierung erreicht bei konstanter Alternierung Stufen der polynomiellen Hierarchie und bei beliebiger Alternierung *PSPACE*.) Mit der ‘semi unbounded’ Alternierung, die von Venkateswaran beschrieben wurde, läßt sich damit auch SAC^k charakterisieren.

Diese Ergebnisse wurden auf dem Komplexitätstheorie–Workshop in Berlin vorgetragen und zu STRUCTURE91 eingereicht.

Gleiche Ergebnisse werden auch mit dem LOG–Abschluß von entsprechenden 1-weg Kellerautomaten und linearen Grammatiken aus der Diplomarbeit im vorhergehende Jahr erreicht. Der LOG–Abschluß ist wie folgt definiert: 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 $LOG(S) := \{Y \mid \exists X \in S \text{ mit } Y \leq^{\log} X\}$.

Es ist deshalb interessant, Komplexitätsklassen auf diese Weise mit formalsprachlichen Methoden zu charakterisieren, da für Kellerautomaten gezeigt werden konnte, daß ein nichtdeterministischer Automat dieser Art mehr Sprachen erkennen kann, als ein deterministischer Automat; bei Turingmaschinen mit Komplexitätsbeschränkung sind dies jedoch ungelöste Probleme, zu deren Lösung die Betrachtung entsprechender formalsprachlicher Klassen eventuell beitragen kann. Beispielsweise kann ein ‘parity’–Kellerautomat (ein Wort liegt in der akzeptierten Sprache, wenn die Anzahl der ak-

zeptierenden Berechnungspfade ungerade ist) Sprachen erkennen, die nicht kontextfrei sind. Für ‘parity’-Komplexitätsklassen konnte eine derartige Trennung bisher nicht gezeigt werden.

Forschungsprojekt: Kommunikationskonzepte in Verteilten Systemen

(*Joachim Münchow*)

Mögliche große Distanzen zwischen den Verarbeitungseinheiten in Verteilten Systemen erlauben ohne größeren Leistungseinbruch nicht, die einzelnen aktiven Elemente des Systems mit Auslastungsinformationen der anderen aktiven Elemente zu versorgen, wie es für das Routing als Teilbereich eines Kommunikationskonzeptes wünschenswert wäre. Auch die quasilokale Vermittlung von Steuerungsinformationen im Netz, als Beispiel werde hier das Prinzip der Auftragsanziehung genannt, bedeutet einen Kommunikations- und Verwaltungszuwachs, wodurch sich die nutzinformationsbezogene Belastbarkeit des Netzes reduziert.

Erarbeitet wurde ein Algorithmus für die Wegoptimierung in belasteten Netzen bei beliebig vorgegebenen Verbindungsstrukturen. Der Algorithmus wurde implementiert und seine Tauglichkeit auf simulativem Wege nachgewiesen.

Derzeit wird ein Konzept erarbeitet, mit dem man der oben angeführten Problematik (Unkenntnis der globalen Auslastungszustände) mit vertretbaren Leistungseinbußen begegnen kann.

Forschungskontakte

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. Odmanski, W. Zielonka*)

Universität Ulm, BRD (*J. Köbler, U. Schöning, Th. Thierauf*)

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

Technische Universität München, BRD

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

Humboldt-Universität Berlin, BRD (*C. Damm*)

Tokyo Woman’s Christian University, Japan (*E. Moriya*)

2.2 Veröffentlichungen

Böcker, H.D.
Eden, H.
Fischer, G.

Interactive Problem Solving Using LOGO
Lawrence Erlbaum Associates,
Hillsdale, New Jersey, 1990.

Böcker, H.D.
Herczeg, J.
Herczeg, M.
Mahling, A.

Beyond Visualization: Knowing and Understanding
In Visualization in Human–Computer Interaction, 7th
Interdisciplinary Workshop on Informatics and Psycho-
logy, Schärding, Austria, May 1988, Lecture Notes in
Computer Science, Vol. No. 439, pp. 16-26
Springer Verlag, Berlin – Heidelberg – New York, 1990

Böcker, H.D.
Herczeg, J.

1) *TRACK — A Trace Construction Kit*
CHI-90, Human Factors in Computing Systems Confe-
rence Proceedings, pp. 415-422
ACM SIGCHI/HFS, 1990

2) *Browsing Through Program Execution*
Proceedings of INTERACT'90, IFIP Conference on
Human–Computer Interaction, pp. 991-996
IFIP, 1990

3) *What tracers are made of*
OOPSLA'90 Proceedings, pp. 89-99
ACM, 1990

Böcker, H.D.
Hohl, H.
Schwab, Th.

1) $\Upsilon\pi\text{ADAPTE}\rho$ — *Ein adaptives Hypertext System zur*
Präsentation von Lerninhalten.
In Hypertext und Hypermedia, S.230-234, Darmstadt,
April 1990. Springer Verlag, Informatik–Fachberichte,
Nr. 249.

2) $\Upsilon\pi\text{ADAPTE}\rho$ — *Individualizing Hypertext*
in: Proceedings of the Third IFIP conference on
Human–Computer Interaction — Interact '90,
Cambridge, U.K.
North Holland, Amsterdam – New York – Oxford – To-
kyo, August 1990

- 3) *Individualisierte, auf ein Benutzermodell gestützte Präsentation von Lerninhalten*
In GI 20. Jahrestagung II, S.340-348, Stuttgart, Oktober 1990. Springer Verlag, Informatik Fachberichte, Nr. 258
- Böcker, H.D.** 1) *Beyond MIDI: Knowledge-Based Support for Computer Aided Composition*
Proceedings of the International Computer Music Conference 1990, S. 284-287, Glasgow, 1990.
ICMC Glasgow 1990 Limited.
- Mahling, A.**
- Wehinger, R.**
- 2) *Ein intelligenter Tutor für den Kompositionsunterricht*
Studioblätter des Studios für Elektronische und Computer Musik der Staatlichen Hochschule für Musik und Darstellende Kunst, Stuttgart, (3):45-52, 1990.
- Bräunl, Th.** 1) *Units in Procedural Programming Languages*
5th Australian Software Engineering Conference 1990, Sydney Australien, Mai 1990, pp. (5)
- 2) *Massiv parallele Programmierung mit dem Parallaxis-Modell*
Springer-Verlag, Berlin Heidelberg New York, Informatik-Fachberichte Nr. 246, Juli 1990
- 3) *Transparent Massively Parallel Programming with Parallaxis*
ISSM International Conference on Parallel and Distributed Computing, and Systems, New York NY, Okt. 1990
- Bräunl, Th.** *The Parallaxis Project*
- Barth, I.**
- Sembach, F.** *Projektdemonstration Joint Conference on Vector and Parallel Processing, VAPP IV & CONPAR 90, Zürich, Sep. 1990*
- Burkhardt, W. H.** 1) *Limitations to Parallel Processing*
Proceedings, International Phoenix Conference on Computers and Communications, Phoenix, Arizona, März 1990

-
- 2) *Quantification of Performance, Connectivity, and Load-Distribution in Multiprocessor Systems*
Proceedings, 33nd Midwest Symposium on Circuits and Systems, Calgary, Canada, August 1990
- Emele, M.** 1) *A Fixed-Point Semantics for Feature Type Systems*
In: Proceedings of The Second International Workshop on Conditional and Typed Rewriting Systems (to appear in the Springer LNCS series), Montreal, June 11-14, 1990
- Zajac, R.** 2) *Typed Unification Grammars*
In: Proceedings of the 13th International Conference on Computational Linguistics (CoLing 90), Helsinki, 20-24 August, 1990
- Diekert, V.** *On confluent semi-commutations-decidability and complexity results*
Reinhardt, K. *TUM-19041, SFB-Bericht Nr. 342/21/90 A, Technische Universität München, Oktober 1990*
- Göttler, H.** 1) *Statements on: 'Use of Graph Grammars in Applications'*
erscheint in: Proc. 'Graph-Grammars and their Application to Computer Science, 4th International Workshop', Bremen, März 1990
- 2) *Computer Graphics for Software Engineering*
Supplement Proc. EUROGRAPHICS'90, Montreaux, September 1990, (eingeladener Vortrag)
- Göttler, H.** *Das juristische Literaturarchiv JULIA*
Günzel, U. *DSWR (Zeitschrift für Datenverarbeitung, Steuer, Wirtschaft, Recht), 1+2/1990, C.H.Beck-Verlag, München, S. 25-27 und 30-34*
- Himmelreich, B.**
- Göttler, H.** *Use Graph Grammars to Design CAD-Systems!*
Günther, J. *erscheint in : Proc. 'Graph Grammars and their Application to Computer Science, 4th International Workshop', Bremen, März 1990*
- Nieskens, G.**
-

- Gunzenhäuser, R.** *Max Bense: Wegbereiter für eine moderne Informatik-Bildung.*
In (1) Semiosis Heft 57/58, Agis-Verlag, Baden-Baden 1990 und (2) Zeichen von Zeichen für Zeichen, Festschrift für Max Bense, Agis-Verlag, 1990
- Heid, U.** 1) *Zur Lexikonarchitektur für ein constraint-basiertes maschinelles Übersetzungssystem*
Erscheint in: Dieter Seelbach/Jürgen Rolshoven (eds.) Akten des XX. deutschen Romanistentages, Sektion Computerlinguistik
- 2) *Syntactic Information in (Machine) Translation Dictionaries — towards a modular architecture for bilingual dictionaries*
in: Proceedings of the Fifth International Symposium on Lexicography, Kopenhagen 1990
- Herczeg, J.** *Combining Hypermedia Browsing with Formal Queries*
Jerke, K.-H. *Proceedings of INTERACT'90, IFIP Conference on Human-Computer Interaction, pp. 593-598*
Lesch, A.
Rößler, H.
Schwab, T. *IFIP, 1990*
- Homeister, Dieter** 1) *Paralleles Booten von Transputern*
Elektronik. Heft 5, März 1990, S.84-89.
- 2) *Semaphores at the Transputer Instruction Level*
Occam Users Group Newsletter. No. 13, July 1990, pp. 46-50
- Ludewig, J.** 1) *Wie man Informatiker hält — Über Softwareleute und ihre Arbeitsumgebung*
Beilage zur TECHNISCHEN RUNDSCHAU 82 (1990), Heft 35, S.10-13
- 2) *Einleitung zur Sitzung Software-Entsorgung*
in A. Reuter (Hrsg.): GI — 20. Jahrestagung. Informatik-FB 257, Band I.
Springer, Berlin usw., S.193-194.
(Beiträge dazu folgen auf S.195-232)

- 3) *Skriptum Informatik I, II*
vdf (Verlag der Fachvereine) Zürich, 1988, 3. Auflage
1991
- Ludewig, J. (Hrsg.)** *Software- und Automatisierungsprojekte — Beispiele aus der Praxis*
Teubner, Stuttgart
- Rathke, Ch.** *Frames and Object-oriented Programming*
Workshop Notes from the 8th National Conference on Artificial Intelligence (AAAI-90), Object-Oriented Programming in AI, Boston, August 1990
- Rathke, Ch.** *Redesign eines Expertensystems zur Unterstützung bei Konfigurieren von CNC-Steuerungen*
König, R. *Beiträge zum 4. Workshop Planen und Konfigurieren, Forschungsinstitut für anwendungsorientierte Wissensverarbeitung, FAW-B.90008, Ulm, Mai 1990*
- Reinhardt, K.** *Hierarchies over the context-free Languages*
Proc. of 6th IMYCS, LNCS, 464,
Springer 1990, S.214-224
- Schweikhardt, W.** *Entwicklung von Ein- und Ausgabemedien für blinde Rechnerbenutzer*
in R. Peschke, M. Fries, U. Kalina (Hrsg.), Materialien zur Schulentwicklung — Heft 12, Hessisches Institut für Bildungsplanung und Schulentwicklung, Informationstechnische Bildung für Sehgeschädigte, Fachtagung des Modellversuchs HERAKLES, 21.-22.2.1990.
- Weber, G.**
- 1) *Interactive dialog technologies on the pin-matrix device in Access to Visual Computer Information by Blind Persons (Hrsg. K. Fellbaum).*
TU Berlin, Einsteinufer 25, 1000 Berlin 10.
S.97-100, 1990

 - 2) *Mensch-Maschine Interaktion auf der Basis von aktiven Zeigehandlungen bei der Stuttgarter Stiftplatte*
in Informationstechnische Bildung für Sehgeschädigte (Hrsg. R. Peschke, M. Fries and U. Kalina),
Verlag Moritz Diesterweg: Frankfurt, S.39-50, 1990

- 3) *Two mice for blind users of the pin-matrix device*
in Proc. of 6th International Workshop on Computer Applications for the Visually Handicapped.
Infovisie Magazine, 4(3), S.1-13, 1990

- 4) *FINGER — a language for gesture recognition*
in Proceedings of INTERACT 90 (Hrsg. D. Diaper et. al.), North Holland, S.689-694, 1990

Weber, G.
Deconinck , F.
Stephanidis, C.

Access to pictorial information for blind users
in Proc. of North Sea Conference on Biomedical Engineering, 1990

Weber, G.
Stephanidis, C.
et al.

Accessability of WIMPS-based user interfaces by visually impaired people
Proc. of ECART conference, Maastricht, ECART c/o Zandbergsweg 11, 6432 CC Hoensbroek, S.8., 1990

Zajac, R.

A Relational Approach to Translation
In: Proceedings of the 3rd International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages, Austin, Texas, Juni 1990

Zell, A.
Korb, Th.
Sommer, T.
Bayer, R.

A Neural Network Simulation Environment
Applications of Neural Networks Conf., Proc. SPIE s 1990 Aerospace Sensing Intl. Symposium, Vol. 1294, 16-20 April 1990, Orlando, Florida

2.3 Berichte

- Bräunl, Th.** *Parallaxis User Manual. A Massively Parallel Programming System*
Barth, I. *Universität Stuttgart, Bericht der Fakultät Informatik, Nr. 3/90, März 1990, pp. (77)*
- Bräunl, Th.** *Realisierung von Maßeinheiten in Pascal-Programmen*
Krauskopf, K. *Universität Stuttgart, Bericht der Fakultät Informatik, Nr. 4/90, August 1990*
- Emele, M.** *A Generator for HPSG*
Universität Stuttgart, Institut für Informatik, Internes Papier
- Eusterbrock, J.** *Algorithmensynthese : Integration deduktiver und induktiver Komponenten*
Universität Stuttgart, Bericht der Fakultät Informatik, Nr. 2/90
- Gunzenhäuser, R.** *Neuere Entwicklungen des rechnerunterstützten Lernens — Ein Überblick über Lernsysteme und Lernsoftware im Bereich der Informatik als Grundlage für die Arbeit einer Softwaregruppe Informatik der Akademischen Software Kooperation (ASK).*
Maier, D. *Studie, Institut für Informatik, Universität Stuttgart, 1990*
- Herczeg, J.** *The Internal Query Component of the HYPERQUERY System*
Schwab, T. *Technical Report, 1990*
- Lagally, K.**
- 1) *WRG — ein neuer Generator für Top-Down-Parser mit automatischer Fehlerbehandlung*
Universität Stuttgart, Bericht der Fakultät Informatik, Nr. 1/90
 - 2) *ARABTEX — eine Erweiterung von LATEX zur Verarbeitung arabischer Texte*
Universität Stuttgart, Bericht der Fakultät Informatik, Nr. 7/90
- Ziegler, B.** *An even faster string search algorithm?*
Universität Stuttgart, Bericht der Fakultät Informatik, Nr. 2/90
-

2.4 Vorträge

Böcker, H.D.

- 1) *Programmvisualisierung durch direkte Manipulation grafischer Objekte*
Freie Universität Berlin, Institut für Informatik
Februar 1990
- 2) *Knowledge-Based Human Machine Communication within the WISDOM*
Project AAAI Spring Symposium on Knowledge-Based Human-Computer Communication,
Stanford, California
März 1990
- 3) *TRACK — A Trace Construction Kit*
CHI'90 — Human Factors and Computing Systems,
Seattle, Washington
April 1990
- 4) *Knowledge-Based Human Machine Communication.*
IBM T.J. Watson Research Center, Yorktown Heights,
New York
April 1990
- 5) *Programm-Visualisierungen — Visuelle Programmierung*
Lehrgang Software-Ergonomie
Technische Akademie Esslingen
Mai 1990
- 6) *What Tracers Are Made Of*
OOPSLA/ECOOP'90, Ottawa, Canada
Oktober 1990
- 7) *Individualizing Hypertext*
NATO Advanced Research Workshop on Cognitive Modelling and Interactive Systems. Eindhoven
November 1990
- 8) *Information Retrieval as a New Field of Human Computer Interaction (Eingeladener Vortrag)*
Internationaler Workshop „Intelligente Schnittstellen zu Informationssystemen.“ Darmstadt
November 1990

- Bräunl, Th.** 1) *Massively Parallel Programming and Applications in Parallaxis*
Engeladener Vortrag zum Informatik-Kolloquium des International Computer Science Institute (ICSI), University of California Berkeley, Okt. 1990
- 2) *Massiv parallele Programmierung*
Engeladener Vortrag zum Informatik-Kolloquium der Bergischen Universität Gesamthochschule Wuppertal, 20. Nov. 1990, Wuppertal
- Deininger, M.** *Ein Schema zur Klassifikation von Metriken. Rechnergestützte Softwarebewertung, Workshop der Technischen Universität Magdeburg*
Magdeburg, 18.-19. Oktober 1990
- Emele, M.** *Organizing Linguistic Knowledge for Multilingual Generation*
Heid, U. *13th International Conference on Computational Linguistics (CoLing90),*
Momma, S. *Helsinki*
Zajac, R.
- Emele, M.** 1) *A Fixed Point Semantics for Typed Feature Structures*
Zajac, R. *International Workshop on Conditional and Term Rewriting Systems, Concordia University, Montréal*
- 2) *Typed Unification Grammar*
13th International Conference on Computational Linguistics (CoLing90), Helsinki
- 3) *The meaning of Typed Feature Structures — semantics and implementation*
International Workshop on Constraint-based Approaches to Natural Language Generation
- Göttler, H.** *Computer Graphics for Software Engineering*
Supplement Proc. EUROGRAPHICS'90, Montreaux, September 1990, (eingeladener Vortrag)
- Gunzenhäuser, R.** 1) *Informatikanwendungen in Pädagogik und Didaktik*
Landespolizeischule Freiburg
Januar 1990
-

- 2) *Wissensbasierte Mensch–Computer–Kommunikation*
FAW Ulm / Universität Ulm
März 1990
 - 3) *Der anwendungsorientierte Informatiker aus der Sicht*
der Universitäten
Berufsakademie Stuttgart
April 1990
 - 4) *Wissensbasierte Mensch–Computer–Kommunikation*
Universität Oldenburg
Mai 1990
 - 5) *Wissensbasierte Mensch–Computer–Kommunikation*
Universität Stuttgart–Hohenheim
Mai 1990
 - 6) *Datenstrukturen und Algorithmen*
Seminar IBM Deutschland GmbH
Mai und September 1990
 - 7) *Lehrgang Software–Ergonomie*
Lehrgangsleitung und Referate
Technische Akademie Esslingen
Mai 1990
 - 8) *Neuere Entwicklungen des rechnerunterstützten*
Lernens
Kultusministerium Rheinland/Pfalz Lehrerfortbildung,
Mainz
Mai 1990
 - 9) *Campus–wide applications of computer–learning.*
Internationale Tagung ICCAI 1990, FernUniv. Hagen
Juni 1990
 - 10) *Informatikausbildung an den Hochschulen in der Bun-*
desrepublik Deutschland
IBM–Symposium, Berlin
Juni 1990
 - 11) *Wissensbasierte Mensch–Computer–Kommunikation*
Universität Rostock
September 1990
-

Gunzenhäuser ff

- 12) *Informatikausbildung an den Hochschulen in der Bundesrepublik Deutschland*
GI-Jahrestagung Stuttgart
Okttober 1990
- 13) *Tutorium: Rechnerunterstütztes Lernen*
DIA — GI Jahrestagung 1990
Okttober 1990
- 14) *Dialog- und Benutzermodellierung*
ETH Zürich, Informatik Kolloquium
November 1990
- 15) *Neuere Entwicklungen des rechnerunterstützten Lernens*
Universität Stuttgart-Hohenheim
November 1990
- 16) *Informatikausbildung an Hochschulen in den alten Bundesländern der Bundesrepublik*
Bonn
Dezember 1990
- 17) *Rechnerunterstütztes Lernen in der Ausbildung*
Kolloquium der Fakultät Architektur der Universität Stuttgart
Dezember 1990

Hanakata, K.

- 1) *A new object-oriented language COOL for knowledge representation and language processing*
Advanced Telecommunications Research Institute International (ATR, Kyoto Japan)
Februar 1990
 - 2) *Multiple Values in COOL and its Database characteristics*
International Institute for Advanced Study of Social Science Institute, Numazu Japan
March 1990
-

- Heid, U.**
- 1) *Syntactic information in (machine) translation dictionaries — towards a modular architecture for bilingual dictionaries*
Fifth International Symposium on Lexicography, Kopenhagen
 - 2) *Subcategorization Hierarchies for Multilingual Generation*
International Conference on Computational Lexicography, Balatonfüred, Ungarn
- Heid, U.**
- Momma, S.**
- Interactions between Linguistic Constraints*
International Workshop on Constraint-based Approaches to Natural Language Generation
- Herczeg, J.**
- 1) *Knowledge-Based Construction Kits*
AAAI Spring Symposium on Knowledge-Based Human-Computer Communication Stanford
März 1990
 - 2) *Browsing Through Program Execution*
INTERACT'90, IFIP Conference on Human-Computer Interaction Cambridge
August 1990
- Hohl, H.**
- ΤπΑΔΑΡΤερ — Ein adaptives Hypertextsystem zur Präsentation von Lerninhalten*
Hypertext/Hypermedia '90 Workshop, GMD-IPSI Darmstadt,
April 1990
- Homeister, Dieter**
- Dammert, Jürgen**
- Paralleles Ray-Tracing*
Tagung der Gesellschaft für Mikroelektronik Stuttgart
Februar 1990
- Homeister, Dieter**
- Dynamische Lastverteilung in Transputernetzen*
2. Transputertreffen der Universität Stuttgart
Stuttgart
Juli 1990

Knödel, Walter

*Optimale Wege in großen Verkehrsnetzen
Jahrestagung der OR–Gesellschaften von Schweiz,
Österreich und Deutschland
Wien
August 1990*

Lichter, H.

- 1) *Die Entwicklung und Umsetzung von System–Prototypen. Arbeitsverfahren in der Software–Entwicklung,
Workshop der GI–FG 2.1.1 „Software–Engineering“
Königswinter, 16.–18. Mai 1990*
- 2) *Die Konstruktion von interaktiven Anwendungen.
Seminar „Objektorientierte Programmierung“
IBM Sindelfingen, 17.–21. September 1990*

Ludewig, J.

- 1) *Die Software–Entwickler und ihre Arbeitsumgebung.
Informatik '90, Seminar der SVD (Schweizerische
Vereinigung für Datenverarbeitung)
Brunnen/Schwyz, 2.–4. April 1990*
 - 2) *Penelope sollte heiraten — zwanzig Jahre Software
Engineering an der Schwelle zur Praxis.
2. msp–Informationsmanagement–Kongreß
Berlin, 14.–15. Mai 1990*
 - 3) *CASE (Computer Aided Software Engineering): Tools,
Lies, and Video Games
Vorträge in den GI–Regionalgruppen Heidelberg und
Stuttgart, 9. Mai und 6. Juni 1990*
 - 4) *Alte und neue Sprachkonzepte, Einstufung und Be–
wertung. SVD–Tagung Ablösung der 3. Generations–
Sprachen?
Zürich, 22. 6. 1990*
 - 5) *15 Jahre Software–Entwurfssprachen — Ein kritischer
Überblick zum Thema CASE. Software Engineering —
Methoden und Werkzeuge
Technische Hochschule Leipzig, 6./7. September 1990*
-

- 6) *Software-Prüfung. Tutorium der Deutschen Informatik-Akademie (DIA) im Rahmen der GI-Jahrestagung 1990*
Stuttgart, 8.10.1990 (zusammen mit K. Frühauf und H. Sandmayr)
- 7) *Software-Qualitätssicherung — eine Illusion?.*
Herbsttagung der SAQ-Sektion Basel Regio
Basel, 23. Okt. 1990
- 8) *Konventionelle Programmiersprachen und objektorientierte Ansätze. Software-Symposium Objektorientierte Konzepte*
IBM Herrenberg, 5.-7. November 1990
- 9) *Software Engineering in der Praxis der 90'er Jahre.*
Eröffnungsvortrag zum Kolloquium der ABB Informatikschule Informatik-Praxis in den 90'er Jahren
Baden/Schweiz, 4.12.1990
- 10) *Seminare und Vorträge bei verschiedenen Firmen (Datenstrukturen und Algorithmen, Software Engineering, Programmiersprachen)*

Mahling, A.

- 1) *Musical Tools for Smalltalk-80*
Institute de recherche et coordination acoustique/musique (IRCAM)
Centre Georges Pompidou Paris
August 1990
 - 2) *Beyond MIDI: Knowledge-Based Support for Computer Aided Composition*
International Computer Music Conference '90 Glasgow
September 1990
 - 3) *Computer music and software development tools for Smalltalk-80*
University of Glasgow Computing Science Department
Glasgow
September 1990
-

- Mahling, A. ff**
- 4) *Computerunterstütztes Komponieren*
10. Arbeitstagung Mensch–Maschine–Kommunikation
Arbeitsgruppe IV: Kunst Maschine Kommunikation
Königswinter
November 1990
- Rathke, Ch.**
- 1) *Frames and Object–Oriented Programming*
AAAI Workshop on Object–Oriented Programming in
AI, Boston, USA,
30.7.1990
- 2) *Framebasierte Wissensrepräsentation mit Hilfe objekt-*
orientierter Programmierung
Kolloquiumsvortrag, Bergische Universität — Gesamt-
hochschule Wuppertal, Fachbereich Mathematik,
8.11.1990
- 3) *Framebasierte Wissensrepräsentation mit Hilfe objekt-*
orientierter Programmierung
Kolloquiumsvortrag, Universität Kaiserslautern,
Fachbereich Informatik,
15.11.1990
- Reinhardt, K.**
- 1) *Hierarchien über den kontextfreien Sprachen*
10. Workshop über Komplexitätstheorie, effiziente
Algorithmen und Datenstrukturen
Bonn
30. Januar 90
- 2) *Leere Alternierung*
12. Workshop Komplexitätstheorie und effiziente
Algorithmen
Berlin
2. Oktober 90
- 3) *Hierarchies over the context-free Languages*
6. International meeting of young computer scientists
Smolenice, CSFR
22. November 90
-

Schwab, T.

- 1) *Planbeschreibungen für die automatische Erkennung von Benutzerhandlungen in interaktiven Systemen*
Kolloquiums-Vortrag,
Wissenschaftliches Zentrum der IBM, Heidelberg
Februar 1990
- 2) *Individuelle, auf ein Benutzermodell gestützte Präsentation von Lerninhalten*
GI-Jahrestagung, Stuttgart
Oktober 1990
- 3) *ΤπΑΔΑΡΤερ — Individualizing Hypertext*
(visual presentation)
Interact '90, Cambridge
August 1990

Schweikhardt, W.

Bildschirmtext- und Graphikdarstellung für Blinde
Informationsveranstaltung der Firma metec,
Ingenieur-Gesellschaft mbH und des Landesgewerbe-
amts Baden-Württemberg zum Thema
Berufliche Integration von Blinden und Sehbehinderten
an EDV-Arbeitsplätzen

Tausend, B.

Erweiterung von Wissensbasen durch analoge Hinweise
3. Fachgruppentreffen der Fachgruppe Maschinelles
Lernen
GMD Bonn
Juli 1990

Bell, S.

- 1) *Einsatz neuerer Forschungsarbeiten für blinde Benutzer*
Arbeitsgruppe der Concerted Action on Technology and
Blindness, Datchet, London, Großbritannien
7.-8. Februar 1990

Weber, G.

- 2) *Mensch-Maschine Interaktion auf der Basis von aktiven Zeigehandlungen bei der Stuttgarter Stiftplatte*
Expertentagung des Modellversuchs Herakles,
Eppenheim
20.-22. Februar 1990

Weber, G. ff

- 3) *GUT — Ein Programm zur taktilen Graphik- und Textwiedergabe*
Arbeitstreffen von Sehbehindertenpädagogen, Nürnberg
27.-28. April 1990 (Vortrag durch D. Kochanek)
- 4) *Graphische Benutzerschnittstellen für Blinde*
Arbeitsgruppe der Concerted Action on Technology and Blindness, Istron Bay, Kreta
1.-4. Mai 1990
- 5) *Benutzerschnittstellen für Blinde und Sehbehinderte*
Universität Heraklion, Kreta
5. Mai 1990
- 6) *FINGER — a language for gesture recognition*
Konferenz INTERACT '90, Cambridge, Großbritannien
29. August 1990
- 7) *Two mice for blind users of the pin-matrix device*
6th International Workshop on Computer Applications for the Visually Handicapped, Leuven, Belgien
20. September 1990
- 8) *Graphics for blind persons and access to graphical user interfaces by the Blind,*
Arbeitstreffen der Concerted Action on Technology and Blindness, Florenz, Italien
10.-12. Dezember 1990

Welsch, Ch.

Objektorientierte Programmentwicklung
Hewlett-Packard Böblingen,
Mai 1990

**Welsch, Ch.
Lichter, H.**

Objektorientierte Programmierung
IBM Sindelfingen,
September 1990

Yoshimoto, K.

Type Hierarchy in Japanese Syntax
International Workshop on Constraint-based Approaches to Natural Language Generation

Zajac, R.

*A Relational Approach to Translation
3rd International Conference on Theoretical and
Methodological Issues in Machine Translation of
Natural Languages
Austin, Texas,
Juni 1990*

Zell, A.

- 1) *Visuelle Repräsentation der räumlichen und personellen Struktur eines Unternehmens in Hypercard
Hypertext/Hypermedia-Fachtagung der SI,
Universität Basel,
April 1990*

- 2) *A Neural Network Simulation Environment
Applications of Artificial Neural Networks,
Orlando, Florida, USA,
April 1990*

2.5 Tagungen

- Burkhardt, W. H.** *Mitglied des Programmkommittees der 20. Jahrestagung der GI
Stuttgart
8.-12. 10. 1990*
- Emele, M.
et.al.** *Workshop über constraint-basierte Formalismen für die Generierung natürlicher Sprache
Bad Teinach
26.-30. November 1990*
- Gunzenhäuser, R.** 1) *Vorsitzender des Programmausschusses des Fachgesprächs Intelligente Lernsysteme, GI-Jahrestagung Stuttgart
Oktober 1990*
2) *Mitglied des Programmkommittees der 20. GI-Jahrestagung Stuttgart
1990*
3) *Mitglied des Programmausschusses der 2. internationa-
nalen Tagung Computers and Handicapped People
(ICCHP)
Zürich
1990*
4) *Mitglied des Programmausschusses der GI-Tagung Informatik in Schule und Ausbildung
Oldenburg
1990*
- Knopik, Th.** 2) *Vorsitzender des Organisationskommittees der 20. Jahrestagung der Gesellschaft für Informatik Stuttgart
1990*
- Ludewig, J.** 1) *Organisation und Leitung der TR-Werkstatt Software-
Metriken
Gerzensee bei Bern
1./2.11.1990*
2) *Mitglied des Programmkommittees der GI-Jahrestagung 1990 (GI'90)*
3) *Mitglied des Programmkommittees der GI-Fachtagung Requirements Engineering 1991 (RE'91)*
-

2.6 Herausgabe von Zeitschriften

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

2.7 Implementierungen

Abteilung Betriebsssoftware

ARABTEX *ArabTEX — eine Erweiterung von LATEX zur Verarbeitung arabischer Texte*
Sprache: Metafont, TEX
K. Lagally

PI *Parser-Interpreter (Prototyp)*
Sprache: Modula-2
Th. Schöbel

Abteilung Computersysteme

SYSEDIT-Erweiterung *MIPRO — Erstellung von Mikroprogrammen*
Sprache: C
S. Rust

XACT-Erweiterung *SYNA — Synthese von endlichen Automaten*
Sprache: C
H.-G. Zipperer

Interface Builder *Softwaresystem zur Gestaltung graphischer Schnittstellen*
Sprache: SUN-C
H.-G. Zipperer

UCD *Universeller Cross-Disassembler*
Sprache: Turbo-Pascal
K. Krause

STRANGE *Disketten Utility zur Analyse beliebig formatierter Disketten*
Sprache: Turbo-Pascal
K. Krause

ALFONS und EGON *Fonteditor — Scannervorlagen-Umwandlung in Softfonts für LaserJet*
Sprache: Turbo-Pascal
K. Krause

HORN C

Übersetzer von Hornklauseln nach C
Sprache: C
J. Walter, Th. Förch

SPARC-Prozessor Karte
J. Walter; M. T. Schneider

Transputer-PC Interface Karte
J. Walter, Werkmann

Abteilung Dialogsysteme

XIT

Ein User Interface Toolkit für das X Window System
Sprache: Common Lisp
J. Herczeg, H. Hohl, M. Ressel, Th. Schwab

HYPERQUERY

Ein interaktives Retrievalsystem für Reiseinformation
Sprache: Common Lisp
J. Herczeg, H. Hohl, M. Ressel, Th. Schwab

COOL

Prototyp einer X-Window-Benutzeroberfläche für
COOL
Sprache: C
K. Hanakata

GUT

Programm zur taktilen Wiedergabe von Graphik und
Text auf Punktschriftdruckern
G. Weber

SPEAK

Programm zum Vorlesen von Texten mit einem deut-
schen Sprachausgabegerät
G. Weber

FONTED und FONTIT

Programme zur Bearbeitung von Zeichensätzen
G. Weber

SBT5

Programm zum interaktiven Arbeiten mit einer kleinen
Punktschriftanzeige
G. Weber

GRAPHDRUCK

Programm zur Ausgabe von Graphiken als tastbare
Punktemuster auf Papier mit äquidistanten Punkten
in waagrechter und senkrechter Richtung
A. Werner

GRAPHDOK

Programm zur tastbaren Wiedergabe abgetasteter Graphiken gedruckter Dokumente
W. Schweikhardt, A. Werner

Abteilung Intelligente Systeme

GERMMORPH

Programmsystem zur Wortklassenbestimmung und Lemmatisierung deutscher Wortformen
Sprache: CommonLisp
E. Lehmann

FRAMETALK

Objektorientierte Sprache zur Repräsentation von Wissen
Sprache: CommonLisp und Clos
Ch. Rathke

3 Fakultätsbezogene Aufgaben

3.1 Dekanat der Fakultät Informatik

Dekan

Prof. Dr. A. Reuter

Prodekan

Prof. Dr. K. Lagally

Sekretariat

Frau K. Erz

3.2 Zentrale Fakultätseinrichtungen

3.2.1 Bibliothek

Wissenschaftl. Beauftragter	<i>Ebinger</i>
Bibliothekarin	<i>Röger</i>
Programmbetreuung	<i>Schlebbe</i>
Wissenschaftl. Hilfskräfte	<i>Busch, Behrens, Finkbeiner, vom Orde, Westermann, Dickenberger</i>

Die Überarbeitung des Bibliotheksbestandes in Bezug auf das aktuelle CR-Schema konnte im Berichtsjahr für alle Sachgruppen im wesentlichen abgeschlossen werden. Das Ergebnis steht in IFIBIB zur Verfügung. Alle Eintragungen des systematischen Kataloges basieren nun auf einem einheitlichen Klassifikationsschema.

Im Berichtsjahr wurde die Ausstattung der Bibliothek in der Breitwiesenstraße geplant mit dem Ziel, der Bibliothek künftig eine angemessene Präsentation des Bestandes in einer benutzerfreundlichen Umgebung zu ermöglichen.

In diesem Zusammenhang stellte sich erneut die Frage nach der Aufstellung der Bücher. Die Argumente entweder zugunsten der alphabetischen oder der systematischen Aufstellung wurden diskutiert. Dabei wurde auch festgestellt, daß der systematische Katalog (CR-Klassifikation) nur sehr wenig benutzt wird.

3.2.2 Rechnernetz

Mitarbeiter

F. Fabian, M. Fabian, H. Sammet

Im Rechnernetzbereich war 1990 hauptsächlich ein Jahr der Planung für den neuen Informatik-Standort in Stuttgart-Vaihingen. Das „Projekt Breitwiesenstraße“ wurde mit erheblichem Zeitaufwand nach den Bedürfnissen der Fakultät und im Umfang der bereitgestellten Mittel mit einem neuen Rechner-Netzwerk und einer dem Bedarf entsprechenden Elektrotechnik geplant. Der Ausbau begann im Sommer und wird Anfang 1991 fertig werden. Aufgrund des bevorstehenden Umzugs wurde das z.Zt. genutzte Rechnernetz nur noch in den notwendigsten Fällen erweitert.

Zum Jahresende 1990 waren ca. 250 Rechner unterschiedlicher Größenordnung an das vorrangig auf Ethernet-Technologie basierende Rechnernetz der Informatik angeschlossen (siehe auch Abb. 3.1 und 3.2):

- 6 SUN Server (4/260, 4/370, 4/110)
- 3 HP Server (320)
- 3 Mehrbenutzersysteme (HP 835, 840)
- 72 SUN Arbeitsplatzrechner (3/50, 3/60, 3/80, 4/20, 4/60)
- 18 MAC II Arbeitsplatzrechner
 - 1 Cadmus Arbeitsplatzrechner
 - 2 DEC Server (5400)
- 26 DEC Arbeitsplatzrechner (VS3100, DEC2100, DEC3100, VAX3100)
- 11 Apollo Server und Arbeitsplatzrechner
 - 7 Lisp-Maschinen (Symbolics, TI-Explorer)
 - 8 IBM Arbeitsplatzrechner (PS2)
- 90 Personal Computer (HP Vectra, PC-AT)
- 7 X-Terminals (NCD, VT1200)
- 3 Terminalserver

Hinzu kommen noch ca. 50 Terminals, die größtenteils via Terminalserver Zugang zu den Rechnern am Informatik-Netz haben.

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

Das Informatik-Rechnernetz wurde im Jahre 1990 geringfügig weiter ausgebaut:

- Anbindung eines weiteren Teilnetzes mittels eines zusätzlich als Router genutzten DEC-Servers.
- Anbindung eines Tokenring-Netzes mittels PS2 Rechner als Router.
- Anbindung eines zuvor unabhängigen LocalTalk-Netzes mittels Bridge-Software auf einem MAC II.
- Anschluß verschiedener leistungsfähiger Rechner (2x VAX 6420, IBM 9370, Sequent Mehrprozessor-System).

Die Datensicherung für viele der mit eigenen Platten ausgestatteten Rechner erfolgt über das „Gesamt–Backup“ auf zwei Exabyte–Geräte (je 2 GB pro 8mm Video–Tape) jede Nacht in der Zeit von 22.00 Uhr – 6.00 Uhr. Es werden 27 Maschinen verschiedener Hersteller (Sun, HP, Dec, IBM) unter verschiedenen Betriebssystemen (SunOS, HP–UX, Ultrix, AIX) gesichert, was einen Bedarf von ca. 13 Tapes (26 GB) monatlich erfordert. Bis auf kleine Kompatibilitäts–Probleme zwischen SunOS und HP–UX, läuft das System recht zuverlässig, so daß verlorengegangene Files problemlos restauriert werden konnten.

3.2.3 Workstations

Mitarbeiter

M. Fabian

Im Laufe des Jahres 1990 sind bei den SUN–Workstations (SUN 3 und 4) insgesamt 34 Service–Fälle aufgetreten. Die erfolgten Reparaturen verteilen sich auf die entsprechenden Hardware–Teile wie folgt:

Service–Statistik 1990					
Hardware	CPU	Monitore	Platten	Sonst.	Gesamt
Anzahl Ausfälle	8	9	11	6	34

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

3.2.4 Elektrotechnik

Mitarbeiter

H. Sammet

Bei einem Großteil der vorhandenen SUN 3 Arbeitsplatzrechner wurden einzelne Komponenten ausgetauscht (Lüfter, Netzteile) um die Geräuschentwicklung am Arbeitsplatz zu verringern.

Die als Terminals genutzten Atari–Rechner erwiesen sich als nicht sehr robust. Ein Teil der defekten Geräte konnte wieder instand gesetzt werden.

Im Rahmen einer 2,5 jährigen Ausbildung haben 5 angehende Kommunikationselektroniker ein 6–monatiges Praktikum an der Fakultät absolviert. Anschließend haben sie die Facharbeiterprüfung vor der IHK abgelegt, die von 4 erfolgreich bestanden wurde.

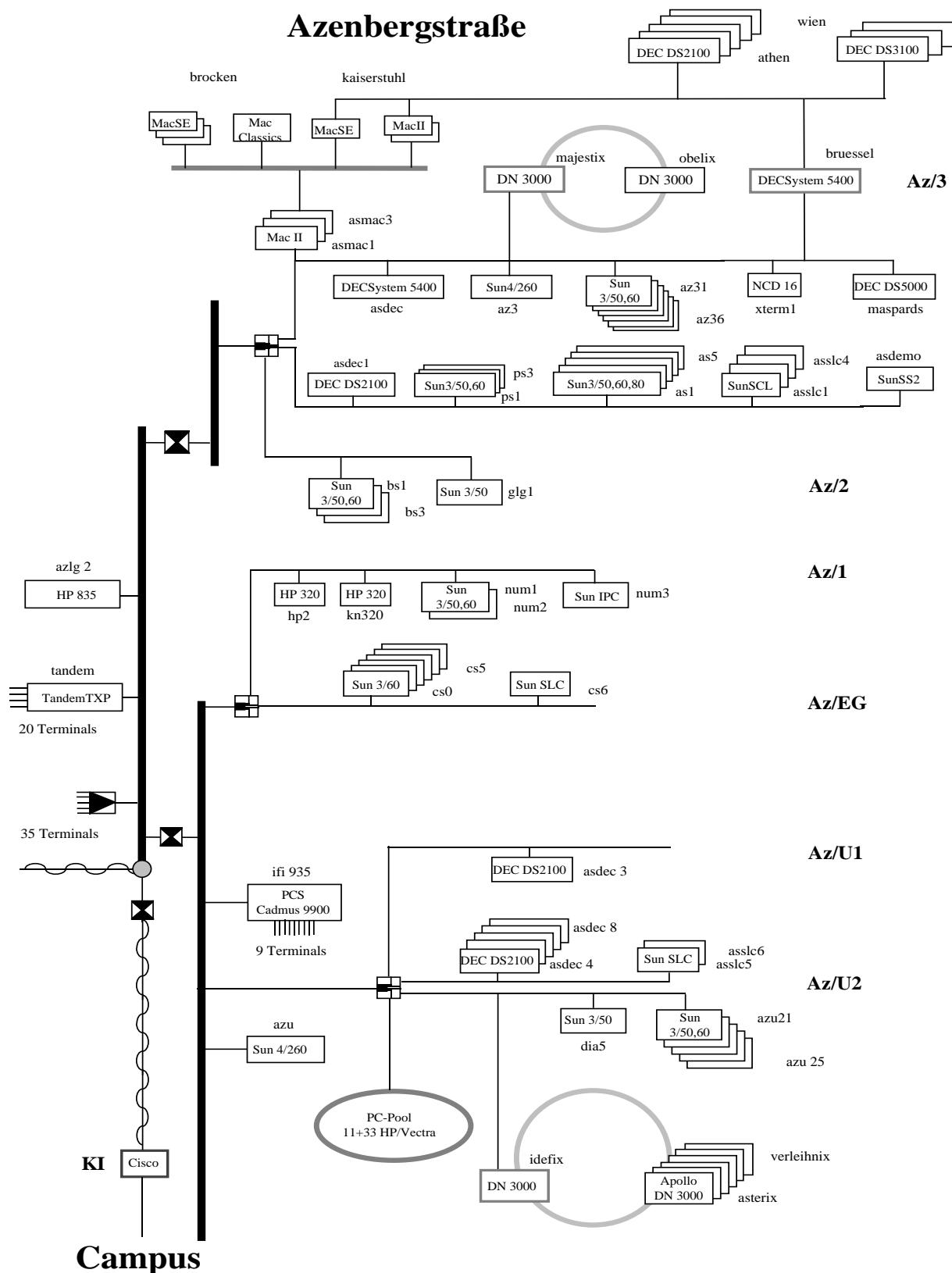


Abbildung 3.1: Informatik-Rechnernetz: Azenbergstraße

Forststraße 86

Herdweg 51

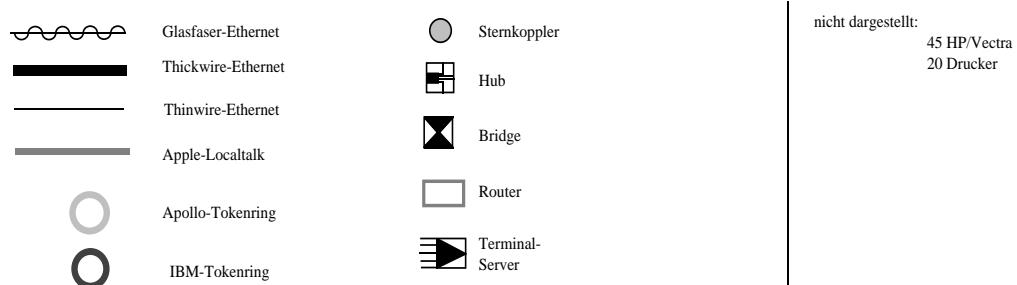
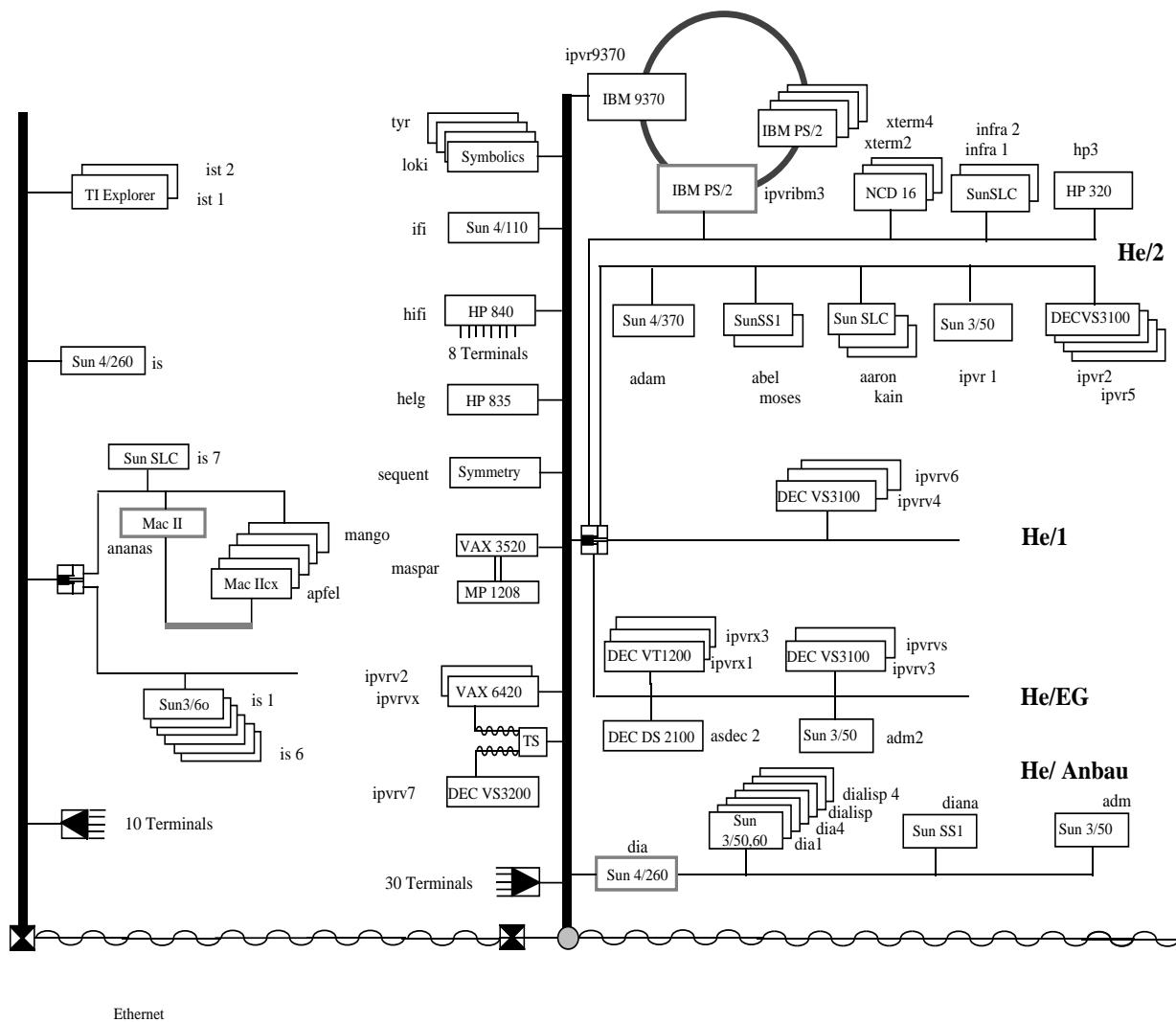


Abbildung 3.2: Informatik-Rechnernetz: Forststraße, Herdweg

3.2.5 Mehrbenutzersysteme

Mitarbeiter

Kohl, Finger, H. Schneider, Wahi

Zwei der im Vorjahr angemieteten vier HP–Mehrbenutzersysteme (Typ HP 9000/835, je 32 MB Hauptspeicher, 1.1 GB Plattenspeicher) wurden nach Beendigung der Mietzeit zurückgegeben, der gleichzeitig geplante Ausbau der anderen beiden verzögerte sich, er wird erst 1991 abgeschlossen werden.

Die bei den als Terminals angeschlossenen ATARI 1040ST inzwischen hohe Defekthäufigkeit führte zum Entschluß, defekte Geräte nur in Eigenarbeit und nur noch dann zu reparieren, wenn Ersatzteile durch Ausschlachten defekter Geräte gewonnen werden können.

Die beantragten X–Terminals dürften daher eher ein Ersatz, als, wie beabsichtigt, eine Ergänzung der bisherigen Terminals werden.

3.2.6 PC–Pools

Beauftragter

Hersmann

Wissenschaftl. Hilfskräfte

Dürschke, Ebersbach, He, Hofmann, Ly Huynh, Mardassi, Milan, Nguyen, Nitsche, Pätzold, Sacher, Schäfer, Sing, Schwenzer, Utz, Waigand, Waitzmann F., Waitzmann H., Wittig

Im Jahre 1990 wurde zur bestehenden Software folgende Software installiert:

ADA–Compiler

Textverarbeitungssystem KomfortText.

Im Jahre 1990 wurden wie im Jahr 1989 Praktika zu den Vorlesungen *Einführung in die Informatik*, *Grundlagen der Informatik* und *Informatik III* sowie verschiedene Software–Praktika, Studienarbeiten und Diplomarbeiten auf diesen Rechnern durchgeführt.

Die Anbindung an das Netz der Fakultät hat sich sehr bewährt.

3.3 Lehre

3.3.1 Aufbau des Informatikstudiums

3.3.1.1 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/Verkehrsingenieurwesen
- 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.

3.3.1.2 Nebenfachstudium Informatik

Informatik wird als Nebenfach bzw. technisches Schwerpunktstudium 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.

3.3.1.3 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.

3.3.2 Lehrveranstaltungen

3.3.2.1 Lehrangebot im Sommersemester 1990

A. Grundstudium für Hörer anderer Fakultäten

Grundlagen der Informatik II (für Studiengang techn. orient. Diplom-Kaufm.)	2 V 1 Ü	Glatthaar (<i>LA</i>), Gunzenhäuser Glatthaar, Gunzenhäuser
Grundlagen der Informatik II (für Studiengang Physik, Luft- und Raumfahrt)	2 V 1 Ü	Gunzenhäuser, Schweikhardt Gunzenhäuser, Schweikhardt
Grundlagen der Informatik II (für VD Maschinenwesen und Verfahrenstechnik)	3 P	Baitinger, Böhm

B. Pflichtlehrveranstaltungen

2. Semester

Analysis II für Informatiker	2 V 1 Ü	Kolbe Kolbe
Lineare Algebra und Analytisch Geometrie	4 V 1 Ü	Roggenkamp Roggenkamp
Differentialgleichungen für Informatiker	2 V 1 Ü	Kolbe Kolbe
Einführung in die Informatik II	4 V 2 Ü	Ludewig Ludewig
Physikalische und elektrotechnische Grundlagen II	2 V 1 Ü	Baitinger Baitinger, Rettig

4. Semester

Numerik	3 V 1 Ü	Berger Berger
Software-Praktikum I	4 P	Ziegler
Aufbau von Datenverarbeitungsanlagen	3 V 1 P	Burkhardt Burkhardt, Homeister

Hardware–Praktikum	4 P	<i>Burkhardt, Homeister, Walter</i>
--------------------	-----	---

6. Semester und 8. Semester

Entwurf und Analyse von Algorithmen und Datenstrukturen	2 V	<i>Zell</i>
	1 Ü	<i>Zell</i>
Berechenbarkeit und Komplexitätstheorie	3 V	<i>Schmidt</i>
	1 Ü	<i>Schmidt</i>
Betriebssysteme	4 V	<i>Lagally</i>
	1 Ü	<i>Lagally</i>
Verteilte Systeme	2 V	<i>Haban (LA)</i>

C. Wahlpflichtveranstaltungen

Codierungstheorie	3 V	<i>Reuß</i>
	1 Ü	<i>Reuß</i>
Entscheidungstheorie	2 V	<i>Ebinger</i>
Prädikatenlogik	3 V	<i>Mijderwijk</i>
	1 Ü	<i>Mijderwijk</i>
Kombinatorik	2 V	<i>Münchow</i>
Formale Semantik	2 V	<i>Lagally</i>
	1 Ü	<i>Lagally</i>
Nichtprozedurale Programmierung	2 V	<i>Lichter, Ludewig</i>
Graphische Datenverarbeitung	2 V	<i>Grieger</i>
Entwurf kundenspezifischer Schaltungen	2 V	<i>Burkhardt, Zipperer</i>
	1 Ü	<i>Burkhardt, Zipperer</i>
Mikroprogrammierung	3 V	<i>Ebert</i>
E/A–Organisation	2 V	<i>Hieber</i>
Entwurf von Schaltnetzen/Schaltwerken	2 V	<i>Baitinger</i>
	1 Ü	<i>Baitinger</i>
CAD, CAD/CAM–Automatisierung des technischen Informationsflusses	1 V	<i>Storr</i>
	1 Ü	<i>Storr</i>
Echtzeitdatenverarbeitung	3 V	<i>Eggenberger</i>

Rechnerunterstütztes Lernen	2 V 1 Ü	Gunzenhäuser Gunzenhäuser
Künstliche Intelligenz I	3 V 1 Ü	Lehmann Lehmann
Datenschutz	2 V	Biller (LA)
Software-Ergonomie	2 V	M. Herczeg (LA)
Konzepte der Datenfernverarbeitung	2 V	Lutz (LA)
Kleinrechnerbetriebssysteme (für PCs)	2 V	Eggenberger
Kleinrechnerbetriebssysteme	2 V	Schimpf
Techniken der Wissensrepräsentation	1 V 1 Ü	Rathke Rathke
Informationssysteme II	2 V	Reuter
Datenbank-Anwendungssysteme	2 V	Reuter
Text- und Listenverarbeitende Verfahren	2 V	Hanakata
Fachpraktikum: Interaktive und Intelligente Systeme	4 P	Böcker, Knopik, Rathke, Hanakata
Fachpraktikum: Rechnerarchitektur	4 P	Burkhardt u. Mitarb.
Fachpraktikum: Datenbanken	4 P	Reuter u. Mitarb.
Software-Entwicklung mit ADA	2 V 1 Ü	Ludewig u. Mitarb. Ludewig u. Mitarb.

D. Seminare

Dialogdesign, Interaktionsmodelle und Werkzeuge zum Bau von Benutzerschnittstellen	2 S	Fehrle (LA), Reddeemann, Knopik
Hypermedien	2 S	Knopik, Weber
Verteilte Transputer-Anwendungen	2 S	Walter
Effiziente Parallelverarbeitung	2 S	Homeister
UNIX-Konzepte	2 S	Armbruster
Expertensysteme	2 S	Lehmann, Burkert
Nicht-klassische Logiken	2 S	Reinhardt

E. Hauptseminare

Bildverstehen	2 HS	<i>Hanakata</i>
Visuelles Programmieren	2 HS	<i>Böcker, Schwab, Gunzenhäuser</i>
Konnektionistische Modelle	2 HS	<i>Zell, Strothotte</i>
Leistungsanalysen von Transaktionssystemen	2 HS	<i>Reuter</i>
Objektorientierte Datenbanken	2 HS	<i>Reuter, Wächter, Heuser</i>
Paralleles Programmieren	2 HS	<i>Bräunl, Eggenberger</i>
Parallele Programmierumgebungen	2 HS	<i>Reuter, A. Maier</i>
Maschinelles Lernen	2 HS	<i>Lehmann, Tausend</i>
Software Engineering	2 HS	<i>Mitarb. Ludewig</i>
Programmsynthese	2 HS	<i>Eusterbrock, Reuter</i>

F. Wahlveranstaltungen

Objektorientierte Programmierung in COOL	2 V	<i>Hanakata</i>
Durchführung wissenschaftlicher Arbeiten	2 K	<i>Ludewig u. Mitarb.</i>

G. Kompaktkurse

Kompaktkurs C		<i>Hanakata</i>
Kompaktkurs PROLOG		<i>Knopik</i>
Kompaktkurs SMALLTALK		<i>Herczeg</i>
Kompaktkurs LISP		<i>Forster</i>
Kompaktkurs OCCAM		<i>Lanchès</i>
Parallele logische Programmierung		<i>Zell</i>

3.3.2.2 Lehrangebot im Wintersemester 1990/91

A. Grundstudium für Hörer anderer Fakultäten

Grundlagen der Informatik I (für Studiengang Physik, Luft- und Raumfahrt)	2 V 1 Ü	Gunzenhäuser, Schweikhardt Schweikhardt
Grundlagen der Informatik I (für Studiengang techn. orient. Diplom-Kaufm.)	2 V 1 Ü	Glatthaar (LA) Gunzenhäuser
Grundlagen der Informatik I (für VD Maschinenwesen und Verfahrenstechnik)	2 V 1 Ü	Baitinger Baitinger, Lanchès

B. Pflichtlehrveranstaltungen

1. Semester

Analysis I	5 V 2 Ü	Werner, Lesky jr. Werner, Lesky jr.
Analysis I (Zusatzübungen)	2 Ü	Werner, Lesky jr.
Grundlagen der Mathematik	4 V 2 Ü	Degen Degen
Einführung in die Informatik I	4 V 2 P	Lagally Lagally
Physikalische und elektrotechnische Grundlagen I	2 V 1 Ü	Burkhardt Burkhardt, Walter

3. Semester

Wahrscheinlichkeitstheorie und Warteschlangen	2 V 1 Ü	Ebinger Ebinger
Logik	4 V 1 Ü	Reuß Reuß
Kombinatorische und sequentielle Netzwerke	2 V 1 Ü	Eggenberger Eggenberger
Einführung in die Informatik III	4 V 2 Ü	Reuter Reuter

5. Semester

Automatentheorie und formale Sprachen	3 V 1 Ü	<i>Posthoff</i> <i>Reinhardt</i>
Nichtklassische Logiken in der Informatik	2 V	<i>Posthoff</i>
Programmiersprachen und Compilerbau	4 V 1 Ü	<i>Schwinn</i> <i>Schwinn</i>
Syntaxanalyse	2 V	<i>Schwinn</i>
Formale Semantik	2 V 1 Ü	<i>Lagally</i> <i>Lagally</i>
Rechnerarchitektur	4 V 1 Ü	<i>Burkhardt, Walter,</i> <i>Zipperer</i> <i>Burkhardt, Walter,</i> <i>Zipperer</i>
Interaktive und intelligente Systeme	4 V 1 Ü	<i>Gunzenhäuser,</i> <i>Lehmann</i> <i>Gunzenhäuser,</i> <i>Knopik, Lehmann</i>
Einführung in die Parallele Programmierung	2 V 4 P	<i>Bräunl</i> <i>Bräunl</i>
Informationssysteme / Datenbanken	4 V 1 Ü	<i>Reuter</i> <i>Reuter</i>
Fertigungsdatenbanken	2 V	<i>Kupper</i>
Verteilte Systeme	1 V	<i>Haban (LA)</i>
Projektmanagement	2 V	<i>Biller (LA)</i>

C. Wahlpflichtveranstaltungen

Graphentheorie	2 V 1 Ü	<i>Knödel</i> <i>Knödel, Liedtke</i>
Kombinatorik II	2 V	<i>Münchow</i>
OR–Methoden	2 V 1 Ü	<i>Berger</i> <i>Berger</i>
Parallele Programmierung von Transputer–Systemen	2 V 1 Ü	<i>Baitinger</i> <i>Baitinger</i>

Entwurf großer Systeme	2 V	<i>Endres (LA)</i>
Software-Engineering	4 V 1 Ü	<i>Ludewig Ludewig</i>
Neuere Entwicklungen bei Hard- und Softwarestrukturen	2 V	<i>Ebert (LA)</i>
Lokale Rechnernetze	2 V	<i>Ebert (LA)</i>
Leistungsmessung von Systemen	2 V 1 Ü	<i>Hieber (LA) Hieber</i>
CAM-, CAP-, CAD/NC-Automatisierung des technischen Informationsflusses I	1 V 1 Ü	<i>Storr Storr</i>
Spezifikation digitaler Systeme	2 V	<i>Ryba</i>
Rechnergestützer Schaltungsentwurf	2 V 1 Ü	<i>Baitinger Baitinger</i>
Musteranalyse und Bildverstehen	2 V	<i>Hanakata</i>
Deduktionsverfahren	2 V	<i>Schönfeld</i>
Natürlichsprachliche Systeme	2 V 1 Ü	<i>Lehmann Lehmann</i>
Symbolmanipulation	2 V	<i>Rathke, Forster</i>
Graphische Datenverarbeitung I	1 V 1 Ü	<i>Grieger Grieger</i>
Systemprogrammierung II	2 V	<i>Eggenberger</i>
Methoden der Systemanalyse	2 V	<i>Seeland (LA)</i>
Matrizennumerik	2 V	<i>Mijderwijk</i>

D. Seminare

Mehrprozessor-Transputer-Systeme	2 S	<i>Homeister</i>
Maschinelles Lernen: Lernen durch Analogie	2 S	<i>Tausend</i>
CASE	2 S	<i>K. Schneider</i>
3D-Computergraphik	2 S	<i>Bräunl</i>
Spezielle Aspekte kundenspezifischer Schaltungen	2 S	<i>Zipperer</i>

E. Hauptseminare

Neuere Entwicklungen in Computer-Systemen	2 HS	<i>Burkhardt</i>
Formale Spezifikation	2 HS	<i>Lichter, Ludewig</i>
Integrierter Systementwurf	2 HS	<i>Baitinger u. Mitarb.</i>
Wissensrepräsentation	2 HS	<i>Lehmann, Burkert</i>
Visuelle Programmierung	2 HS	<i>Gunzenhäuser</i>
Objektorientierte Wissenspräsentation in COOL	2 HS	<i>Hanakata</i>

F. Wahlveranstaltungen

Durchführung wissenschaftlicher Arbeiten	2 K	<i>Ludewig u. Mitarb.</i>
--	-----	---------------------------

G. Kompaktkurse

PROLOG		<i>Knopik</i>
Smalltalk		<i>Herczeg, Mahling</i>
MODULA-2-Kompaktkurs		<i>Deininger, K. Schneider</i>
ADA		<i>Lichter, Schwille</i>
Graphik-Normen (GKS, GKS-3D, Phigs)		<i>Grieger</i>
LISP		<i>Forster</i>

3.3.3 Informatik-Kolloquium

10.1.	Bärtschi, Martin ETH Zürich	<i>Datenbankunterstützung für die Integration von Ingenieurwerkzeugen</i>
10.1.	Kochs, Hans-Dieter ESWE AG., Wiesbaden	<i>Integrierte Prozeßautomatisierungssysteme zur Überwachung, Optimierung und Steuerung von Produktionsanlagen</i>
16.1.	Dittrich, Klaus R. Uni Zürich, IFI	<i>Objektorientierte Datenbank-Technologie: Wo wir stehen und wohin wir gehen (sollten)</i>
26.1.	Gentzsch, W. FH Regensburg	<i>FB Allgemeinwissenschaften, Mathematik und Informatik</i>
31.1.	Hoppe, Ulrich GMD-IPSI, Darmstadt	<i>Lernen durch Beobachtung als Methode des Wissenserwerbs für planerkennende Hilfesysteme</i>
2.2.	Brauer, Johannes Johann-Wolfgang-Goethe-Universität, Frankfurt/Main	<i>Schnittstellenspezifikation in offenen integrierten CAD-Systemen</i>
6.2.	Ibrahim, Bertrand Informatik-Zentrum der Universität Genf	<i>A development environment for computer based learning</i>
7.2.	Ehlers, Manfred University of Maine, USA	<i>Zur Integration von digitaler Bildanalyse und Informationssystemen in der Fernerkundung</i>
9.2.	Stadel, Manfred Siemens AG., Abt. Software-Technik, München	<i>Disziplinierte Anwendung von Vererbung und Polymorphismus</i>
9.2.	Franzen, Helmut TFH Berlin, Fachbereich: Informatik	<i>Petri-Netze (ein methodisches Bindeglied zwischen Informatik und Ing.-Wissenschaften)</i>
12.2.	Zehnder, Carl. A. ETH Zürich	<i>Informatik-Projektführung — Kunst oder Handwerk?</i>
13.2.	Levi, Paul Institut für Informatik der TU München	<i>Wissensbasiertes Sehen und Planen autonomer Agenten</i>

13.2.	Hanakata, Kenji IfI, Uni Stuttgart	<i>Problemlösen beim Bildverstehen</i>
1.3.	Lemke, Andreas C. University of Colorado	<i>Research on design environment at the university of colorado</i>
24.4.	Kastens, Uwe Universität GH-Paderborn	<i>Paralleler Code und parallele Hardware erzeugt durch Übersetzungsmethoden</i>
26.4.	Strobach, Peter Siemens AG., Forschung und Entwicklung, München	<i>Algorithmische Konzepte des maschinellen Bewegungssehens</i>
8.5.	Winkler, Jürgen F.H. Siemens AG., Zentrale Forschung, München	<i>Object-CHILL — eine objektorientierte Erweiterung von CHILL</i>
9.5.	Fischer, G. University of Colorado	<i>Kooperative, wissensbasierte Designumgebungen für die Gestaltung, Nutzung und Wartung von Software</i>
16.5.	Ploedereder, Erhard Tartan Laboratories, Monroeville/PA, USA	<i>Sprachevolution am Beispiel „Ada“</i>
22.5.	Göttler, Herbert Friedrich-Alexander-Universität Erlangen-Nürnberg	<i>Erstellung von intelligenten, interaktiven, graphischen Editoren</i>
29.5.	Grosch, Josef GMD Forschungsstelle an der Universität Karlsruhe	<i>Automatische Generierung von Compilern</i>
5.6.	Böttcher, Stefan IBM Deutschland GmbH., Institut für wissensbasierte Systeme Stuttgart	<i>Logikprogrammierung und deduktive Datenbanken in PROTOSt-l</i>
26.6.	Roos, Gerhard Institut für Mikroelektronik Stuttgart	<i>3-Dimensionale CMOS Integration am IMS</i>
27.6.	Gawlick, Dieter TP-West Laboratory, Mountain View, CA, USA	<i>Der Mythos der Zugriffs-Lücke</i>

- 3.7. **Niemann, H.**
Lehrstuhl für Informatik 5
(Mustererkennung), Universität
Erlangen–Nürnberg
*Ein Ansatz zur wissensbasierten
Analyse industrieller Szenen*
- 10.7. **Schmidt, Dieter**
University of Cincinnati und
Universität Stuttgart
*Computer Algebra und die Berech-
nung der Bahn des Mondes*
- 30.10. **Johnson, Peter**
Queen Mary and Westfield
College University of London
*User Interaction — a mapping
between, domain, users, tasks and
computers*
- 11.12. **Posthoff, Ch.**
Universität Chemnitz
*Computerschach und künstliche
Intelligenz*
- 18.12. **Arnold, L.**
Technische Universität
Chemnitz, Sektion
Informationstechnik
Digitale Bildkommunikation

3.3.4 Habilitationen und Examensarbeiten

3.3.4.1 Dissertationen

Matheis, Hans *Informationsverwaltung für Software-Projekte*

Hauptbericht: *Ludewig*

Mitbericht: *Zehnder (ETH Zürich)*

Rust, Stefan *Zur Automatisierung des Systementwurfs integrierter Schaltungen*

Hauptbericht: *Burkhardt*

Mitbericht: *Höfflinger (Institut für Mikroelektronik Stgt.)*

Welsch, Christoph *Integration von Konzepten der objektorientierten und logischen Programmierung*

Hauptbericht: *Barth (DFKI Kaiserslautern)*

Mitbericht: *Ludewig*

3.3.4.2 Diplomarbeiten

- Ackermann, Joachim** *Verifizierer von Konsistenzbedingungen*
Prüfer: Reuter (IPVR)
Betreuer: Liebelt (IPVR)
- Albicker, Klemens** *Entwurf und Realisierung von Visualisierungsalgorithmen für die grafikunterstützte Simulation*
Prüfer: Storr (ILR)
Betreuer: Junghans (ILR)
- Ammann, Carina** *Analytische Modelle für das Leistungsverhalten paralleler Hash-Join-Algorithmen*
Prüfer: Reuter (IPVR)
- Athanassiu, Eleutherios** *Design, Entwurf und Implementierung einer Benutzeroberfläche zur graphischen Darstellung von Performancewerten*
Prüfer: Reuter (IPVR)
Betreuer: Zink (IPVR)
- Bachmann, Ulf** *Implementierung eines Partitionierungsverfahrens zur Verteilung von Prozessen auf konfigurierbare Transputernetze*
Prüfer: Baitinger (IPVR)
Betreuer: Lanchès (IPVR)
- Baier, Andreas** *Kommunikations- und Schnittstellenprobleme beim Entwurf und der Nutzung von Expertensystemen*
Prüfer: Gunzenhäuser
- Barth, Ingo** *Entwicklung eines Compilers für Parallaxis mit dynamischen Verbindungsstrukturen*
Prüfer: Ludewig
Betreuer: Bräunl
- Bassler, Thomas** *OBJ-PROLOG: Logische Programmierung mit Feature-Termen und Constraints*
Prüfer: Göttler
Betreuer: Welsch
- Baum, Winfried** *Automatische Erzeugung von Blockdiagrammen*
Prüfer: Burkhardt
Betreuer: Rust

Becker, Wolfgang	<i>Kooperatives Planen unabhängiger Agenten</i> Prüfer: Reuter (IPVR) Betreuer: Zell
Beckmann, Franz	<i>Ein Referenzmodell für die Benutzerschnittstelle eines integrierten Systems zur Unterstützung von Kernaufgaben der Systemanalyse</i> Prüfer: Gunzenhäuser Betreuer: Boidol (Softwarehaus IKOSS)
Beha, Horst-Dieter	<i>Ein Generator für Strukturbrowser</i> Prüfer: Göttler Betreuer: Welsch
Behm, Roland	<i>Objektorientierte Programmierung in Concurrent Prolog</i> Prüfer: Reuter (IPVR) Betreuer: Zell
Beil, Thomas	<i>Objekte mit dynamischen Protokollen in Smalltalk-80</i> Prüfer: Göttler Betreuer: Welsch
Bell, Siegfried	<i>Erweiterung von Wissensbasen durch analoge Hinweise</i> Prüfer: Lehmann Betreuer: Tausend
Binder, Joachim	<i>Test und Programmierung einer Adapterkarte zur Anbindung einer E/A-Einheit an den VME-Bus und Installation der Adapterkarte in ein VME-System</i> Prüfer: Ebert (AEG-Ulm) Betreuer: Goldrian (AEG-Ulm)
Blank, Karlheinz	<i>Table Manager: Ein objektorientiertes Werkzeug zur Erstellung und Manipulation von Tabellen</i> Prüfer: Gunzenhäuser Betreuer: Fehrle (IBM)
Bollmann, Stefan	<i>Erzeugung und Darstellung von Stereobildern</i> Prüfer: Grieger (ILR), Gunzenhäuser Betreuer: Grieger (ILR)
Bohn, Fred-Michael	<i>Entwurf und Aufbau eines Transputer-Subsystems für hochauflösende 24-Bit Farbgrafik</i> Prüfer: Ebert (AEG-Ulm) Betreuer: Homeister (IPVR)

Bohner, Markus	<i>Eine Entwicklungsumgebung für interaktive Benutzerschnittstellen in der objektorientierten Sprache COOL mit dem X Window System</i> Prüfer: Hanakata
Both, Ralph	<i>Ein Softwarewerkzeug zur interaktiven Spezifikation von Lexikoneinträgen für natürlichsprachliche Systeme</i> Prüfer: Lehmann Betreuer: Burkert, Forster
Brand, Wolfgang	<i>Programmieren mit semiformalen Datenstrukturen</i> Prüfer: Strothotte (IBM)
Briel, Edgar	<i>Simulation des Systemverhaltens typischer paralleler Rechnerstrukturen</i> Prüfer: Baitinger (IPVR) Betreuer: Lanchès (IPVR), Rettig (IPVR)
Brummer, Peter	<i>Stochastisch lernende neuronale Netze — Konzeptionen — Algorithmen — Erste Versuche</i> Prüfer: Knödel
Buhl, Lothar	<i>Modular Attribute Grammars</i> Prüfer: Endres (IBM) Betreuer: Steinhoff (IBM)
Butz, Jochen	<i>Entwurf einer fenstergestützten Bedieneroberfläche zum Betrieb einer flexiblen Fertigungszelle</i> Prüfer: Gunzenhäuser, Warnecke (IFF) Betreuer: Sieger (IFF)
Dammert, Jürgen	<i>Paralleles Ray-Tracing unter Helios auf Transputern</i> Prüfer: Baitinger (IPVR), Ebert (AEG-Ulm) Betreuer: Homeister (IPVR)
Dotzek, Karl	<i>Strategien zur Pronominalisierung</i> Prüfer: Studer (IBM), Lehmann Betreuer: Novak (IBM)
Draxler, Markus	<i>Realisierung des „mathematical laboratory“ MATLAB in einer RSYST-Umgebung</i> Prüfer: Rühle (RUS) Betreuer: Lagally, Otter (DLR)

Duttle, Klaus	<i>Ein globales Verdrahtungsprogramm für Makrozellen</i> Prüfer: Burkhardt Betreuer: Rust
Dworski, Claus	<i>Erweiterung des Einplatinencomputers EMUF 08 zum autonomen Betrieb und Peripherieanschluß</i> Prüfer: Burkhardt Betreuer: Zipperer
Ebhart, Dieter	<i>Entwurf und Vergleich von Auftragsverteilungsalgorithmen für punktweise entkoppelte Probleme</i> Prüfer: Ebert (AEG-Ulm) Betreuer: Homeister (IPVR)
Fach, Peter W.	<i>Aufgabenorientierte Endbenutzer-Dokumentationen für Benutzerschnittstellen mit direkter Manipulation</i> Prüfer: Strothotte (IBM)
Förderreuther, Johannes	<i>SILO-Implementierung eines Optimierers für die Systemebene des graphischen Entwurfssystems SYSEDIT</i> Prüfer: Burkhardt Betreuer: Rust
Gehring, Steffen	<i>Integration von Freiformflächen und Optimierung in Ray Tracing Verfahren</i> Prüfer: Ebert (AEG-Ulm) Betreuer: Grieger (ILR)
Geltz, Markus	<i>Ein graphischer Browser für gerichtete Graphen</i> Prüfer: Gunzenhäuser Betreuer: Böcker
Gerlach, Stefan	<i>Entwurf eines anwendergerechten Regelschemas zur Netzplanauswertung</i> Prüfer: Göttler, Bullinger (IFF) Betreuer: Thaler (IAO)
Geschke, Thomas	<i>Interprozeßkommunikation mit zentralem Vermittlerprozeß</i> Prüfer: Lagally Betreuer: H. Maier (UVA Unverzagt GmbH)
Geuder, Uwe	<i>Maßnahmen zur Erhöhung der Datenverfügbarkeit in Datenbanksystemen</i> Prüfer: Reuter (IPVR)

Greitmann, Christiane	<i>Untersuchung und Implementierung der Anbindung von Fenstersystemen an EIFFEL</i> Prüfer: Baitinger, Gunzenhäuser Betreuer: Ryba (IPVR)
Günther, Volker	<i>Entwurf eines modellgestützten Diagnosemoduls und seine Integration in ein diagnostisches Gesamtkonzept</i> Prüfer: Lehmann Betreuer: Steger (IPA)
Günthör, Roger	<i>Das Sachbearbeitermodell: Ein Entwicklungssystem für modulare und verteilte Programme aus dem Bereich der kommerziellen Datenverarbeitung</i> Prüfer: Göttler
Härdtner, Monika	<i>Implementierung und Bewertung verschiedener Hash-Join-Verfahren</i> Prüfer: Reuter (IPVR) Betreuer: Schiele (IPVR)
Harm, Oliver	<i>Design und Implementierung eines Protokollanalyse-Tools</i> Prüfer: Lagally Betreuer: Winterling (IBM)
Hohl, Hubertus	<i>Ein graphik-orientiertes Werkzeug zum Browsing in Objektstrukturen</i> Prüfer: Gunzenhäuser Betreuer: Herczeg, Schwab
Hölzlein, Markus	<i>Eine nichtparametrische Schätzung der Erneuerungsfunktion</i> Prüfer: Knödel Betreuer: Walk (Mathematik A)
Huppenbauer, Helmut	<i>UND-parallele Ausführung von Logikprogrammen auf der Basis des RAPiD-Datenflußmodells: Ein Implementierungskonzept für Multicomputer</i> Prüfer: Lagally Betreuer: Schimpf
Huynh, Van Loc	<i>VISMAP: Ein System zur Visualisierung und Manipulation von Baumstrukturen</i> Prüfer: Gunzenhäuser, Bullinger (IFF) Betreuer: Trefz (IAO), Koller (IAO)

Janosch, Manfred	<i>SILSIM-Implementierung eines SIL-Simulators für die Systemebene des graphischen Entwurssystems SYSEDIT</i> Prüfer: Burkhardt Betreuer: Rust
Jacobsen, Oliver	<i>Geometrische Figurenerkennung und graphische Bildverarbeitung mit GKS und PHIGS</i> Prüfer: Gunzenhäuser, Grieger (ILR) Betreuer: Grieger (ILR)
Jensen, Kai-Uwe	<i>Ein methodischer Vergleich bestehender User-Interface-Management-Systeme</i> Prüfer: Gunzenhäuser Betreuer: W. Maier (HP)
Kaja, Manfred	<i>Auswahl und Implementierung von Optimierungsverfahren für eingebettete Prozeduren (Interpolationsroutinen) in SQL-Anfragen</i> Prüfer: Reuter (IPVR) Betreuer: Neugebauer (IPVR)
Kaiser, Andreas	<i>Parallel-C-Compiler für Transputer</i> Prüfer: Ebert (AEG-Ulm), Lagally Betreuer: Homeister (IPVR)
Kochanek, Dirk	<i>Ein Hypertext-System für blinde Zeitungsleser</i> Prüfer: Gunzenhäuser Betreuer: Weber
Koczy, Oliver	<i>Verfahren zur Personalzuteilung bei zyklischer Dienstplanerstellung</i> Prüfer: Lagally Betreuer: Böhm (IPVR), Leopold (IPA)
König, Rainer	<i>Redesign eines Expertensystems für Konfigurationsaufgaben</i> Prüfer: Lehmann Betreuer: Rathke
Korb, Thomas	<i>Entwurf und Implementierung einer Beschreibungssprache für neuronale Netze</i> Prüfer: Reuter (IPVR) Betreuer: Zell
Krauskopf, Karsten	<i>Ein massiv paralleles Verfahren zur Stereobildauswertung</i> Prüfer: Ludewig Betreuer: Bräunl

Krauß, Rainer	<i>Precompiler zur Analyse von Transaktionsprogrammen</i> Prüfer: Reuter (IPVR) Betreuer: Liebelt (IPVR)
Kremm, Edwin	<i>Partitionierung von VLSI-Chips unter technologischen Randbedingungen</i> Prüfer: Baitinger (IPVR) Betreuer: Hartmann (IFT München)
Kutschker, Siegmar	<i>Entwicklung und Implementierung eines Verfahrens zur Minimierung unvollständig spezifizierter Mealy-Automaten</i> Prüfer: Eggenberger, Knödel Betreuer: Berger
Liebelt, Sabine	<i>Entwicklung und Untersuchung von massiv parallelen Hidden-Surface- und Raytracing-Algorithmen</i> Prüfer: Ludewig Betreuer: Bräunl
Magino, Frank	<i>Entwurf und Realisation eines Modells zur Verwaltung von Logistik-Basisdaten unter dBase</i> Prüfer: Ebert (AEG-Ulm) Betreuer: Ordnung (Mercedes-Benz)
Merling, Dieter	<i>Entwicklung eines grafischen Basiseditors für Objekte und Beziehungen</i> Prüfer: Ludewig Betreuer: Lichter
Müller, Klaus	<i>Graphische Echtzeitdarstellung paralleler Deduktion</i> Prüfer: Reuter (IPVR) Betreuer: Duppel (IPVR)
Nitsche, Doris	<i>Syntaxanalyse von elliptischen Ausdrücken im Deutschen</i> Prüfer: Gunzenhäuser Betreuer: Machate (IAO)
Pauer, Gertrud	<i>Entwurf eines Autorensystems mit exemplarischem Einsatz zur Einführung in die Programmiersprache APL2</i> Prüfer: Gunzenhäuser Betreuer: Schweikhardt
Pauler, Gerold	<i>Entwurf eines ISDN-Koprozessors für den ISA-Bus (IBM-PC/XT/AT)</i> Prüfer: Lagally Betreuer: Ebert (AEG-Ulm)

Pfisterer, Jörg	<i>Konzeption und Implementierung eines Datenservers mittels Interprozeßkommunikation für ein verteiltes CIM-System in einem LAN auf der Basis von 4.3. BSD UNIX (Berkeley)</i> Prüfer: Ebert (AEG-Ulm) Betreuer: Thines (IFF)
Pirlein, Thomas	<i>Rekonstruktion von Hintergrundwissen für ein wissensbasiertes textverstehendes System</i> Prüfer: Lehmann, Studer (IBM) Betreuer: von Luck (IBM)
Quecke, Hans	<i>ARKI — Ein wissensbasiertes System zur Grundrißgestaltungen von Wohnungen</i> Prüfer: Gunzenhäuser Betreuer: Böcker
Rabenda, Jörg	<i>Implementierung eines Front-End Compilers für die Programmiersprache TURING</i> Prüfer: Lagally
Recker, Martin	<i>Erstellen einer tastbaren Orientierungsseite zu gedruckten Dokumenten</i> Prüfer: Gunzenhäuser Betreuer: Schweikhardt
Reichert, Lars	<i>Formale Beschreibungen von direktmanipulativen Benutzerschnittstellen</i> Prüfer: Strothotte (IBM)
Renschler, Ursula	<i>Entwurf und Implementierung einer experimentellen Komponente zur CONTRACT-Verarbeitung</i> Prüfer: Reuter (IPVR) Betreuer: Wächter (IPVR), Gugel (IPVR)
Richau-Lojen, Jürgen	<i>Implementierung und Bewertung verschiedener Hash-Join-Verfahren</i> Prüfer: Reuter (IPVR) Betreuer: Schiele (IPVR)
Robitschko, Stefan	<i>Erprobung und Bewertung moderner CNC- und SPS-Steuerungen im Kommunikationsverhalten mit einem Leitrechner: bei Einsatz der international genormten Protokollarchitektur MAP und MMS</i> Prüfer: Eggenberger Betreuer: Fischer (Mercedes-Benz)

Roos, Claudia	<i>Spezifikation eines Auskunftssystems zur Verwaltung von Informationen über Programme und Dokumente</i> Prüfer: <i>Lagally</i> Betreuer: <i>Holder (Mercedes-Benz), Böhm (IPVR)</i>
Ruzicka, Georg	<i>Floorplanerstellung für Entwürfe der RT-Ebene in SYSE-DIT</i> Prüfer: <i>Burkhardt</i> Betreuer: <i>Rust</i>
Schadt, Christof	<i>Entwurfsumgebung für den physischen DB-Entwurf</i> Prüfer: <i>Reuter (IPVR)</i> Betreuer: <i>Haberhauer (IPVR)</i>
Schäfer, Gert	<i>Message-Bus für System- / Netzwerk-Management Anwendungen</i> Prüfer: <i>Ebert (AEG-Ulm)</i>
Schedel, Friedrich	<i>Programmsystem zur Durchführung von Umlegungsrechnungen im Netzen des öffentlichen Personennahverkehrs und deren Darstellung auf PostScript-fähigen Laserdruckern</i> Prüfer: <i>Knödel, Heimerl (IEV)</i> Betreuer: <i>Dobeschinsky (IEV)</i>
Schick, Karl-Wilhelm	<i>Implementierung einer Parallelen Deduktiven Datenbank — Optimierung und parallele Evaluierung</i> Prüfer: <i>Reuter (IPVR)</i> Betreuer: <i>Duppel (IPVR)</i>
Schindler, Jörg	<i>Design einer Sprache zur Beschreibung paralleler Abläufe</i> Prüfer: <i>Reuter (IPVR)</i> Betreuer: <i>Schiele (IPVR)</i>
Schloß, Bernhard	<i>Entwurf und Implementierung von Layoutfunktionen für die automatische Generierung von Makrozellen</i> Prüfer: <i>Burkhardt</i> Betreuer: <i>Rust</i>
Schmidt, Harald	<i>Implementierung eines Literatur-Informationssystems</i> Prüfer: <i>Ebert (AEG-Ulm)</i>
Schmidt, Ursula	<i>Modellierung und Simulation einer massiv parallelen Anwendung</i> Prüfer: <i>Reuter (IPVR)</i> Betreuer: <i>Schiele (IPVR)</i>

Schnaithmann, Dieter	<i>Einbindung wissensbasierter Techniken in eine operative DV-Umgebung</i> Prüfer: Gunzenhäuser Betreuer: Merz (BTB GmbH, Stuttgart)
Schneider, Mark-Tell	<i>Entwicklung einer SPARC-Prozessor-Karte mit verschiedenen Hardware-Entwurfssystemen</i> Prüfer: Burkhardt Betreuer: Walter
Schöbel, Thomas	<i>Ein neues Verfahren zum Erkennen kontextfreier Sprachen</i> Prüfer: Lagally
Schuler, Stefan	<i>Entwurf und Realisierung eines „Struktur berücksichtigenden“ Programmeditors für die Sprache Pascal</i> Prüfer: Lagally
Schwille, Jürgen	<i>Konzeption und Entwicklung eines Werkzeugs zur Simulation von objektorientierten Programm-Architekturen</i> Prüfer: Ludewig Betreuer: Lichter
Seltmann, Uwe	<i>Wissenserwerb für regelbasierte Systeme</i> Prüfer: Gunzenhäuser Betreuer: Merz (BTB GmbH, Stuttgart)
Sembach, Frank	<i>Entwicklung eines symbolischen Debuggers für das parallele Sprachensystem Parallaxis/PARZ</i> Prüfer: Ludewig Betreuer: Bräunl
Sommer, Tilman	<i>Eine graphische Benutzeroberfläche unter X-Windows zur Manipulation und Visualisierung neuronaler Netze</i> Prüfer: Reuter (IPVR) Betreuer: Zell
Sonntag, Christof	<i>Ein Simulator neuronaler Netze für den Macintosh II</i> Prüfer: Reuter (IPVR) Betreuer: Zell
Spang, Anton	<i>Implementierung eines Compilers zur Übersetzung der Anweisungsliste der DIN-IEC 65 A in die Programmiersprache der Steuerung Bosch CL300</i> Prüfer: Eggenberger Betreuer: Hegel (IFF)

Sundermann, Michael	<i>Kürzeste Wege in großen Verkehrsnetzen</i> Prüfer: Knödel
Thies, Markus Andreas	<i>Interaction Control Manager. Ein System zum Navigieren durch Interaktionen und Pläne</i> Prüfer: Gunzenhäuser Betreuer: Fehrle (IBM)
Thomaschewski, Alfred	<i>Interaktives Werkzeug zur Darstellung und Analyse paralleler Abläufe</i> Prüfer: Reuter (IPVR) Betreuer: Gugel (IPVR)
Vojkoviz, Vojko	<i>Implementierung einer parallelen deduktiven Datenbank, parallele Evaluierungsverfahren für rekursive Prädikate</i> Prüfer: Reuter (IPVR) Betreuer: Duppel (IPVR)
Wahl, Susanne	<i>Ergonomische Betrachtung von Benutzerschnittstellen bei Expertensystemen</i> Prüfer: Gunzenhäuser, Bullinger (IFF) Betreuer: Gräble (IFF)
Waibel, André	<i>Systembeobachtung in einem objektorientierten verteilten Betriebssystem</i> Prüfer: Eggenberger, Bullinger (IFF) Betreuer: Meitner (IAO)
Weilbach, Jörg Michael	<i>Entwicklung und Implementierung von Heuristiken zur Bil- dung von Planungsalternativen von Produkt- und Maschi- nengruppen im Rahmen des Fertigungsinselplanungssystems FIPS</i> Prüfer: Lagally Betreuer: Böhm (IPVR), Hallwachs (IAO)
Weiβ, Jürgen	<i>Vorhersage der Lastfaktoren von Joinberechnungen auf Mehrprozessor Systemen</i> Prüfer: Reuter (IPVR) Betreuer: Duppel (IPVR)
Wörner, Bernhard	<i>Parallelisierungsansätze in der Finite-Elemente-Methode</i> Prüfer: Reuter (IPVR)

Zeller, Martin *Erweiterung eines Compilers für die Sprache PROTOS-L
um ein Modulkonzept*
Prüfer: Studer (IBM), Schönfeld (IBM)
Betreuer: Beierle (IBM)

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

3.3.4.3 Studienarbeiten

Ahle, Lothar	<i>Eine Benutzeroberfläche für ein Petri-Netz-basiertes Montage-Simulationswerkzeug</i> Prüfer: Gunzenhäuser Betreuer: Krauth (IFF)
Anders, Raimund	<i>Studienarbeit Nr. 937</i> Prüfer: Ebert (AEG-Ulm)
Antz, Dirk	<i>Benutzerschnittstelle zur Unterstützung des Entwurfs konsistenzerhaltender Datenbankanwendungsprogramme</i> Prüfer: Reuter (IPVR) Betreuer: Liebelt (IPVR)
Bader, Hans-Jürgen	<i>Grundlagen des internationalen Standardformats STEP</i> Prüfer: Lagally Betreuer: Mohrmann und Wolters (debis Systemhaus GmbH)
Banhart, Matthias	<i>Auswahl und Realisierung eines Backup-Konzepts für die Teilprojekte des PWAB-Projekts „Naturmeßfeld Horkheimer Insel“</i> Prüfer: Reuter (IPVR) Betreuer: Neugebauer (IPVR)
Barth, Ingo	<i>Entwicklung eines Compilers für die parallele Programmiersprache Parallaxis</i> Prüfer: Barth (DFKI Kaiserslautern) Betreuer: Bräunl
Bäumer, Dirk	<i>Konzeption und Realisierung eines Tutorsystems für die SMALLTALK-80 Programmierumgebung</i> Prüfer: Ludewig Betreuer: Lichter
Beck, Jochen	<i>Schutzmechanismen gegen Computerviren</i> Prüfer: Ebert (AEG-Ulm)
Becker, Roland	<i>Implementierung eines Vererbungskonzepts für modulares Prolog</i> Prüfer: Göttler Betreuer: Welsch

Becker, Wolfgang	<i>Time–Reasoning Komponente in Prolog</i> Prüfer: Göttler Betreuer: Zell
Beutelschieß, Frank	<i>Entwicklung und Evaluierung von Werkzeugen zur regelbasierten Programmierung unter Prolog</i> Prüfer: Schönenfeld (IBM) Betreuer: Werneke (IBM)
Beutenmüller, Ulrich	<i>Fehlertolerante Ausführung paralleler Abläufe am Beispiel der Fertigungssteuerung</i> Prüfer: Reuter (IPVR) Betreuer: Kißling (IPVR)
Bosch, Monika	<i>Einbindung von Erweiterungen in die relationale Anfragesprache SQL zur Selektion von Meßwerten über Interpolationsverfahren</i> Prüfer: Reuter (IPVR) Betreuer: Neugebauer (IPVR)
Brommundt, Andreas	<i>Entwicklung eines Window–Managers mit flexibler interaktiver Benutzeroberfläche auf Grundlage des standardisierten Graphiksystems PHIGS</i> Prüfer: Bullinger (IFF) Betreuer: Seidel (IAO)
Burkert, Holger	<i>Entwicklung eines Daten– und Verarbeitungsmodells für den Kontext</i> Prüfer: Reuter (IPVR) Betreuer: Wächter (IPVR)
Büttner, Frank	<i>Untersuchungen zum Leistungsverhalten von SQL/DS bei Berücksichtigung verschiedener Tabellen– und Indexstrukturen</i> Prüfer: Endres (IBM) Betreuer: Krämer (IBM)
Christ, Oliver	<i>Die Nutzbarmachung eines maschinenlesbaren Standardlexikons durch Transformation in eine CLOS–basierte lexikalische Datenbasis</i> Prüfer: Lehmann Betreuer: Heid, Momma

Dinkelacker, Robert	<i>Interaktive Grafiktools mit dem 3D-Grafikstandard PHIGS+</i> Prüfer: Hanakata Betreuer: Wedig (MPI)
Dörre, Ingrid	<i>Entwurf einer Benutzungsschnittstelle für eine Terminologiedatenbank</i> Prüfer: Gunzenhäuser Betreuer: R. Mayer (IFF)
Draxler, Markus	<i>Entwurf und Analyse eines Rechners mit dem Clipper-Modul</i> Prüfer: Burkhardt
Drehmann, Kerstin	<i>SimTerm — Eine einfache Termbeschreibungssprache</i> Prüfer: Lehmann Betreuer: Rathke
Dung, Van Tran	<i>Entwurf und Spezifikation eines prototypischen CONTRACT-Managers</i> Prüfer: Reuter (IPVR) Betreuer: Wächter (IPVR)
Eisenmann, Gerd	<i>Entwurf und objektorientierte Implementierung eines Diagramm-Editors</i> Prüfer: Gunzenhäuser, Bullinger (IFF) Betreuer: Trefz (IAO), Weisbecker (IAO)
Eleni, Filippidou	<i>Verteilung zufälliger Punkte im Würfel</i> Prüfer: Knödel
Engelhardt, Stefan	<i>Graphische Darstellung von Syntaxbeschreibungen</i> Prüfer: Göttler Betreuer: Bräunl
Ermisch, Frank	<i>Ein Hypercard-Informationsprogramm zum Studienplan der Fakultät Informatik</i> Prüfer: Barth (DFKI Kaiserslautern) Betreuer: Zell
Fabricius, Joachim	<i>Vorgehensweise bei Leistungsuntersuchungen von SQL/DS unter Berücksichtigung verschiedener Indexcharakteristiken</i> Prüfer: Endres (IBM) Betreuer: Schulz (IBM)

Feldner, Wolfgang	<i>VIS/VAS — Ein Veranstaltungs-Informations- / Veranstaltungs-Autoren-System</i> Prüfer: Gunzenhäuser Betreuer: Knopik
Förch, Thomas	<i>Ein Compiler zur Übersetzung von Hornklauseln in C-Prozeduren</i> Prüfer: Burkhardt Betreuer: Walter
Fritz, Ronald	<i>Spezifikation eines multifunktionalen Experimentierboards für Logic Cell Arrays (LCA)</i> Prüfer: Baitinger (IPVR)
Geltz, Markus	<i>Ein graphischer Browser für gerichtete Graphen</i> Prüfer: Gunzenhäuser Betreuer: Böcker
Gmeinwieser, Norbert	<i>Entwicklung und Aufbau einer Experimentierplatine für programmierbare Gate-Arrays</i> Prüfer: Burkhardt Betreuer: Zipperer
Greiner, Thorsten	<i>Implementierung des RSA-Verfahrens</i> Prüfer: Knödel
Grün, Birgit	<i>Ein formularorientiertes Constraint-System</i> Prüfer: Lehmann Betreuer: Rathke
Hanisch, Jürgen	<i>Schaltungsentwurf und Layout von Multiplizierern mit einem 3D-CMOS Prozeß</i> Prüfer: Baitinger (IPVR) Betreuer: Roos, Rettig (IPVR)
Harrëus, Thomas	<i>Entwurf eines DAC-Peripherickarte und zugehöriger Anwendungssoftware zur Sprach-Ein/Ausgabe mit einem PC</i> Prüfer: Eggenberger
Hecker, Christoph	<i>Visualisierung von Voronoi-Diagrammen</i> Prüfer: Knödel
Huebner, Ralf	<i>Dithering auf einem Tintenstrahldrucker</i> Prüfer: Böhm (IPVR) Betreuer: Dammert (IPVR)

Hung, do Cuong	<i>Erkennung der Hardware-Komponenten im PC-System</i> Prüfer: Eggenberger
Kauer, Dietmar	<i>Interface Builder für Sun View</i> Prüfer: Burkhardt Betreuer: Zipperer
Kleiner, Thomas	<i>Erweiterung eines Mikroprogrammcompilers zur algorithmischen Beschreibung und Simulation mikroprogrammierter Steuerwerke</i> Prüfer: Burkhardt Betreuer: Zipperer
Koch, Martin	<i>Implementierung einer hochgenauen Gleitpunktarithmetik mit Intervallarithmetik</i> Prüfer: Böhm (IPVR) Betreuer: Homeister (IPVR)
Kochanek, Dirk	<i>Taktile Graphik- und Textwiedergabe</i> Prüfer: Gunzenhäuser Betreuer: Weber
Krauskopf, Karsten	<i>Ein Precompiler zur Realisierung von Datenobjekten mit Einheiten</i> Prüfer: Göttler Betreuer: Bräunl
Kreissl, Wolfgang	<i>Erlernen des vorteilhaften Einsatzes von Farbe in Benutzerschnittstellen</i> Prüfer: Strothotte (IBM)
Kubiak, Günther	<i>Leistungsuntersuchungen am relationalen DBMS INGRES zur Bestimmung von geeigneten Strukturen von DB-Schemata am Beispiel einer Meßwertendatenbank</i> Prüfer: Reuter (IPVR) Betreuer: Neugebauer (IPVR)
Kutschera, Peter	<i>Eine graphische Oberfläche für den Entwurf, die Kontrolle und die Beobachtung verteilter Anwendungen</i> Prüfer: Reuter (IPVR) Betreuer: Haban (Mercedes-Benz)
Lal, Deepak	<i>Entwicklung und Implementierung einer Bedieneroberfläche für ein Anwendersystem mit Hilfe von X</i> Prüfer: Gunzenhäuser Betreuer: Hoppe (IFF)

Langbein, Karin	<i>VIS/VAS — Ein Veranstaltungs-Informations- / Veranstaltungs-Autoren-System</i> Prüfer: Gunzenhäuser Betreuer: Knopik
Laube, Dieter	<i>Entwurf und Spezifikation eines prototypischen CONTRACT-Managers</i> Prüfer: Reuter (IPVR) Betreuer: Wächter (IPVR)
Liebelt, Sabine	<i>Implementierung eines Datenbankauskunftssystems</i> Prüfer: Reuter (IPVR) Betreuer: Neugebauer (IPVR)
Lindorfer, Gerold	<i>Entwurf und Implementierung eines Laufzeitsystems zur Visualisierung der Aktivitäten von verteil und parallel ausgeführten Anwendungen</i> Prüfer: Reuter (IPVR) Betreuer: Haban (Mercedes-Benz)
Lieven, Oliver	<i>PCTERM — Eine Terminalemulation für PC's</i> Prüfer: Eggenberger
Mache, Niels	<i>Entwurf und Realisierung eines effizienten Simulatorkerns für neuronale Netzwerke</i> Prüfer: Reuter (IPVR) Betreuer: Zell
Mackamul, Harald	<i>Ein Baukasten für Brettspiele</i> Prüfer: Gunzenhäuser Betreuer: Böcker
Mardassi, Mohsen	<i>Entwurf und Implementierung eines Zeichensatzes für das arabische Alphabet</i> Prüfer: Lagally
Masmanidis, Ioannis	<i>Syntax und Compiler für eine Sprache zur Spezifikation Verteilter und Paralleler Abläufe</i> Prüfer: Reuter (IPVR) Betreuer: Wächter (IPVR)
Mellouli, Taïeb	<i>Implementierung eines Nicht-Resolutionstheorembeweisers in Prolog</i> Prüfer: Schönenfeld (IBM)

Menz, Falko	<i>Untersuchung von Simulationsmethoden und -techniken zu ihrer Programmierung</i> Prüfer: <i>Ludewig</i> Betreuer: <i>Lichter</i>
Merath, Frank	<i>Entwicklung und Implementation eines Regelsystems für FRAMETALK</i> Prüfer: <i>Lehmann</i> Betreuer: <i>Rathke</i>
Mühlbradt, Klaus	<i>Generierung von Ada-Paketrahmen aus einer Software-Architekturbeschreibung</i> Prüfer: <i>Ludewig</i> Betreuer: <i>Lichter</i>
Müller, Hiltrud	<i>Beschreibung paralleler Abläufe am Beispiel der technologischen Planung</i> Prüfer: <i>Reuter (IPVR)</i> Betreuer: <i>Kißling (IPVR)</i>
Naschold, Michael	<i>Lernprogramm für UNIX</i> Prüfer: <i>Burkhardt</i>
Nguyen, van Quan	<i>Entwurf eines Interpreters für die Sprache einer speicherprogrammierbaren Steuerung (SPS)</i> Prüfer: <i>Eggenberger</i> Betreuer: <i>Hengel (IFF)</i>
Norz, Roland	<i>Bau eines Modula-P Precompilers für den Sequent-Symmetry Parallelrechner</i> Prüfer: <i>Ludewig</i> Betreuer: <i>Bräunl</i>
Pauer, Gertrud	<i>Entwurf und Implementierung eines Programms zum Erstellen und Verwalten eines Bildlexikons für gehörlose Schüler</i> Prüfer: <i>Gunzenhäuser</i> Betreuer: <i>Schweikhardt</i>
Pimiskern, Joachim	<i>Automatische Komposition mittels Graphgrammatiken</i> Prüfer: <i>Göttler</i>

Raich, Udo	<i>Entwurf und Realisierung einer flexiblen Benutzerschnittstelle einer Meßumgebung für Nicht-Standard Datenbankanwendungen</i> Prüfer: Reuter (IPVR) Betreuer: Schiele (IPVR)
Raichle, Bernd	<i>Entwicklung eines Daten- und Verarbeitungsmodells für den Kontext</i> Prüfer: Reuter (IPVR) Betreuer: Wächter (IPVR)
Räumschüssel, Stefan	<i>Erstellen eines Berechnungsmoduls und einer Benutzeroberfläche auf X-Windows für ein Verzahnungsentwurfsprogramm</i> Prüfer: Gunzenhäuser Betreuer: Köchling (IMG)
Recker, Martin	<i>Der Sortieralgorithmus Merge-Insertion ist nicht optimal</i> Prüfer: Knödel Betreuer: Eusterbrock
Reczko, Martin	<i>Überwachte Lernverfahren für mehrlagige, nicht-rekurrente neuronale Netzwerke</i> Prüfer: Lehmann Betreuer: Tausend, Sträßle (IPE)
Rieth, Peter	<i>Einbindung von Erweiterungen in die relationale Anfrage-sprache SQL zur Selektion von Meßwerten über Interpolationsverfahren</i> Prüfer: Reuter (IPVR) Betreuer: Neugebauer (IPVR)
Robitschko, Stefan	<i>Entwurf und Implementierung einer Langzahl-Arithmetik für Kreisteilungskörper</i> Prüfer: Lagally Betreuer: Lagally
Ruhland, Michael	<i>Beschreibungsmethoden für graphische Benutzeroberflächen am Beispiel X-Windows</i> Prüfer: Gunzenhäuser Betreuer: Weber
Sander, Frank	<i>Ein Programm, das lernt, Rubik's Clock zu lösen</i> Prüfer: Barth (DFKI Kaiserslautern) Betreuer: Zell

Schmidt, Stefan *Ein Gerätetreiber für das Betriebssystem OS/2*
Prüfer: Eggenberger
Betreuer: Kerker (HP)

Schönherr, Iris *Erweiterung einer Benutzerschnittstelle zur Datenaufbereitung für ein Netzplanprogramm*
Prüfer: Gunzenhäuser
Betreuer: Horlacher (Institut für Wasserbau)

Schwanzer, Ingrid *Vergleich von Tabellenkalkulations- und 4GL-Systemen anhand einer ingenieurspezifischen Anwendung*
Prüfer: Reuter (IPVR)
Betreuer: Haberhauer (IPVR),
Hidlacher (Institut für Wasserbau)

Schwenkreis, Friedemann *Interprozeßkommunikation über TCP/IP am Beispiel von MOVIE.BYU*
Prüfer: Baitinger (IPVR)
Betreuer: Teutsch (Institut für Wasserbau)

Sembach, Frank *Entwicklung eines Simulators für die parallele Zwischen sprache PARZ*
Prüfer: Barth (DFKI Kaiserslautern)
Betreuer: Bräunl

Spang, Anton *Übersetzung einer allgemeinen Programmiersprache für speicherprogrammierbare Steuerungen (SPS) in Sprachen handelsüblicher SPS*
Prüfer: Eggenberger, Bullinger (IFF)
Betreuer: Helgel (IAO), Rettich (IAO)

Stahl, Irene *Ein CLOS-Browser*
Prüfer: Lehmann
Betreuer: Rathke

Staiger, Ralph-Christoph *Massiv parallele Algorithmen zur Stücklistenauflösung*
Prüfer: Reuter (IPVR)
Betreuer: Duppel (IPVR)

Sundermann, Michael *Schnelle Fouriertransformation*
Prüfer: Knödel

Utz, Angelika	<i>Erstellung eines interaktiven graphischen Analyse-Interfaces für entscheidungsunterstützende Systeme</i> Prüfer: Gunzenhäuser Betreuer: Voß (IER)
Veigel, Andreas	<i>Ein Informationsprogramm über das IPVR in Stuttgart</i> Prüfer: Reuter (IPVR) Betreuer: Zell
Verba, Reinhard	<i>Massiv-parallele Algorithmen zur Lösung von Problemen der linearen Algebra</i> Prüfer: Ludewig Betreuer: Bräunl
Vetter, Dirk	<i>Die Weiterentwicklung der IBM-Expertensystem-Entwicklungs-Umgebungen von 'Expert System Environment' über 'IBM Knowledge Tool' zu 'AD/Cycle – The Integrated Reasoning Shell'</i> Prüfer: Gunzenhäuser Betreuer: Urban (IBM)
Waidelich, Andreas	<i>Implementierung eines baumorientierten Theorembeweisers</i> Prüfer: Lehmann Betreuer: Machate (IAO)
Walter, Volker	<i>Entwurf von massiv parallelen Simulated Annealing Algorithmen</i> Prüfer: Ludewig Betreuer: Bräunl
Werkmann, Hubert	<i>Entwicklung und Aufbau einer Transputer/PC Interface-Karte</i> Prüfer: Burkhardt Betreuer: Walter
Wichert, Michael	<i>EDDA — Ein graphischer Editor für LISP-Datenstrukturen</i> Prüfer: Gunzenhäuser Betreuer: Böcker
Wihofszky, Jochen	<i>Ein mausgesteuertes Vorleseprogramm für Blinde</i> Prüfer: Gunzenhäuser Betreuer: Weber

Wimpff, Roland	<i>Implementierung einer Schnittstelle zwischen Maschinensteuerungen und Fertigungsleitrechner auf Basis der international genormten Protokollarchitektur MAP</i> Prüfer: <i>Eggenberger</i> Betreuer: <i>Fischer (Mercedes-Benz)</i>
Wittmann, Holger	<i>Berechnung komplexer Körperschnitte im Octree-Modell mit 2 Grundkörpern</i> Prüfer: <i>Storr (Institut für Luft- und Raumfahrt)</i> Betreuer: <i>Joannides (Institut für Steuerungssysteme)</i>
Wollensak, Uta	<i>Konzeption und prototypische Implementierung einer dialogorientierten Prüfkomponente</i> Prüfer: <i>Bullinger (IFF)</i> Betreuer: <i>Wasserlos (IFF)</i>
Zimmerer, Peter	<i>Vergleich verschiedener Lernverfahren für neuronale Netze</i> Prüfer: <i>Reuter (IPVR)</i> Betreuer: <i>Zell</i>
Zink, Volker	<i>Lernen vom morphologischen Wissen</i> Prüfer: <i>Lehmann</i> Betreuer: <i>Tausend</i>

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

Abkürzungen

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

IPVR	Institut für Parallele und Verteilte Höchstleistungsrechner
ISO	International Organization for Standardization
IVS	Informatik-Verbund Stuttgart
LA	Lehrauftrag
MWK	Ministerium für Wissenschaft und Kunst Baden Württemberg
NTG	Nachrichtentechnische Gesellschaft
RUS	Rechenzentrum der Universität Stuttgart
SC	Subcommittee
SFB	Sonderforschungsbereich
SI	Schweizerische Informatikergesellschaft
SIG	Special Interest Group
SIGCHI	Special Interest Group Computer Human Interaction
TC	Technical Committee
WG	Working Group