

# **Lehrstuhl für Informatik 2 (Programmiersprachen und Programmiermethodik)**

**Anschrift:** Martensstr. 3, 91058 Erlangen

**Tel.:** +49.9131.85.27621

**Fax.:** +49.9131.85.28809

**E-Mail:** info@i2.informatik.uni-erlangen.de

## **Leitung:**

Prof. Dr. Michael Philippsen

## **Emeritus:**

Prof. em. Dr. rer. nat. Hans Jürgen Schneider

## **Sekretariat:**

Agnes Brütting

Elfriede Schörmal

## **Wiss. Mitarbeiter:**

Dipl. -Math Volker Barthelmann

Dr. -Ing. Ingrid Fischer

Dr. -Ing. Matthias Gröbner (bis 31. 07. 03)

Dipl.-Inf. Michael Klemm (ab 01.11. 03)

PD Dr. -Ing. Mark Minas (bis 31.10. 03)

Dipl. -Inf. Frank Strüber (bis 31.10. 03)

Ronald Veldema, Ph.D.

PD Dr. -Ing. habil. Peter Wilke

## **Gäste/Stipendiaten:**

Franz Forman

Dipl. -Inf. (FH) Bernd Hardung

Dr. Klaus Kreisel

PD Dr. Gabriella Kókai

Dr. Hermann Puhlmann

Dipl.-Inf. Dominic Schell

Dipl.-Inf. Szilvia Zvada

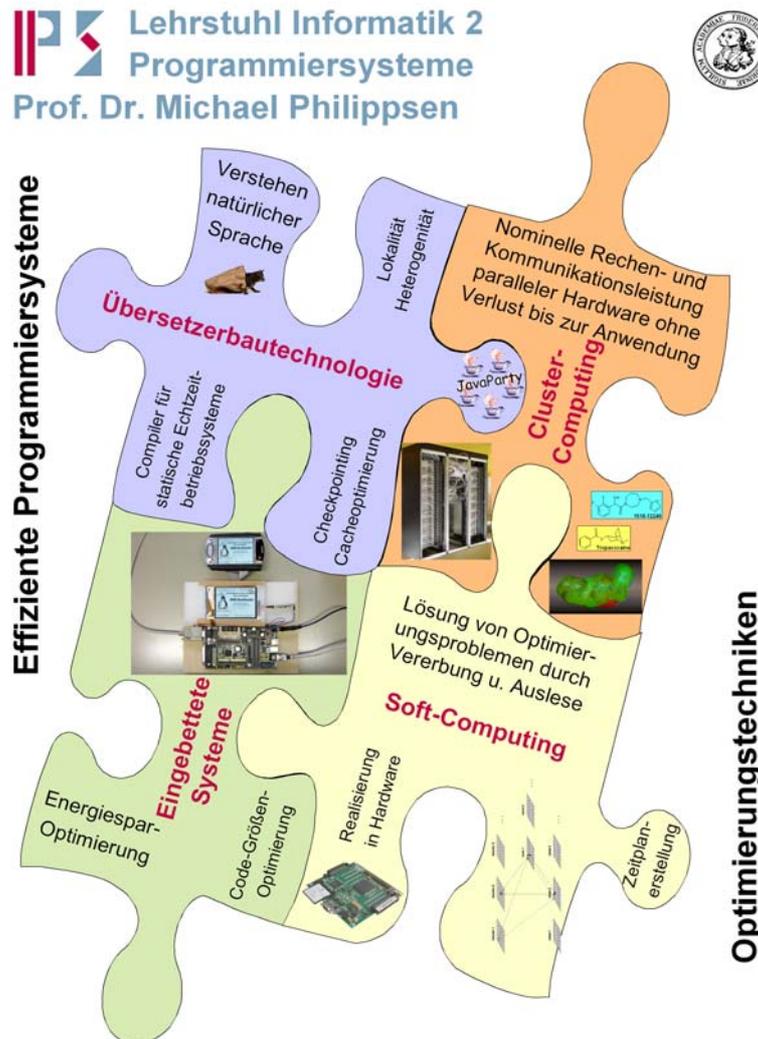
## **1 Einführung**

Im Mittelpunkt der Forschungsarbeiten des Lehrstuhls stehen parallele und verteilte Systeme und deren Programmierung sowie Programmiersysteme für eingebettete und mobile Systeme. Software (und deren Erstellung) für solche Systeme sollte nicht komplexer aber genauso porta-

bel, wartbar und robust sein, wie heute für Einprozessorsysteme und Arbeitsplatzrechner. Langfristiges Ziel ist es, Anwendungen die verfügbare Rechen- und Kommunikationsleistung möglichst ungebremst zur Verfügung zu stellen bzw. aus sehr begrenzten Systemen ein Maximum an Möglichkeiten herauszuholen. Um diese Vision zu verwirklichen, bedarf es bei sich fortentwickelnden technologischen Grundlagen (von Parallelrechnern über verteilte Systeme bis hin zu allgegenwärtigem eingebettetem Rechnen der Zukunft) erstklassiger technischer Einzelbeiträge in mehreren Teilgebieten der Informatik (z.B. Entwurf von Programmiersprachen, Optimierungstechniken für Übersetzer, Software-Architekturen, Werkzeuge zum Software-Engineering, Laufzeitsystem, Betriebssystemintegration, Entwurfsmuster, Parallelität, etc.) sowie einer gebietsübergreifenden Arbeitsweise.

## 2 Forschungsschwerpunkte

Die laufenden Arbeiten des Lehrstuhls gliedern sich in vier ineinander greifende Puzzle-Steine/Arbeitsgruppen. Durch gegenseitiges Nehmen und Geben entstehen Synergie-Effekte, die die Leistungen der einzelnen Gruppen über sich hinaus wachsen lassen.



Die **Übersetzerbau-Gruppe** arbeitet an neuen Analyse- und Optimierungstechniken, vor allem im Bereich der Laufzeitübersetzung. Fortschritte in diesen Bereichen werden von den Arbeitsgruppen **Cluster-Computing** und **Eingebettete Systeme** dringend zur Lösung ihrer jeweiligen

Forschungsarbeiten benötigt. Andererseits stellen gerade diese Forschungsarbeiten ein umfangreiches Evaluationsfeld für die Übersetzerbaugruppe dar. Im Bereich der eingebetteten Systeme wird zur Zeit an der Minimierung des erforderlichen Speicherbedarfs, der besseren Ausnutzung der Register in statischen Betriebssystemen sowie der Energieeinsparung gearbeitet. Im Bereich des Cluster-Computings steht eine Anwendung aus der Medizin im Mittelpunkt: das Entdecken von gemeinsamen Molekülfragmenten. Die Lösung dieses Anwendungsproblems (mit Hilfe von weiterzuentwickelnden Data Mining-Techniken) erfordert so viel Rechenleistung, dass herkömmliche Programmiermethoden für Rechnerbündel nicht ausreichen. Auch übliche Rechnerbündel reichen nicht - aus heterogenen Einheiten zusammengesetzte Grids sind erforderlich. Dies stellt harte Forderungen an die Übersetzerbaugruppe. Das vierte Puzzle-Teil, **Soft-Computing**, steht in Interaktion sowohl mit dem Arbeitsbereich Eingebettete Systeme als auch dem Bereich Cluster-Computing. Derzeit werden genetische Algorithmen in (eingebettete) Hardware umgesetzt. Geplant ist, genetische Algorithmen auf Rechnerbündeln zu implementieren, um die zur Verfügung stehende Rechenleistung zum schnelleren Durchforsten des Suchraums einzusetzen. Ein Arbeitsgebiet sind die Zeitplanungsverfahren - bekanntlich kann ein guter Stundenplan, der alle Randbedingungen erfüllt, nur mit heuristischer Suche gefunden werden. Genetische Algorithmen finden überraschend gute Lösungen in kurzer Zeit. Durch Verwendung von Rechnerbündeln ist ein weiterer Fortschritt zu erwarten, wenn der Übersetzerbau es ermöglicht, die nominelle Rechenleistung möglichst verlustfrei bis zur Anwendung zu bringen.

### 3 Forschungsprojekte

#### 3.1 Übersetzung für eingebettete Systeme

**Projektleitung:**

Prof. Dr. Michael Philippsen

**Beteiligte:**

Dr.-Ing. Volker Barthelmann, Dipl.-Inf. Dominic Schell, Ronald Veldema, Ph.D.

**Laufzeit: 1.1.2001 - 31.12.2004**

**Förderer:**

Deutsche Forschungsgemeinschaft  
3Soft GmbH

Ob im Auto oder zur Steuerung von Maschinen, in vielen elektronischen Geräten sind eingebettete Systeme ein unverzichtbarer Bestandteil. Mit zunehmender Komplexität und Größe der Anwendungen ergibt sich die Anforderung, Software für eingebettete Systeme in modernen Programmiersprachen zu entwickeln. Diese Sprachen wurden jedoch insbesondere für Desktop-Systeme entwickelt, welche mit verschwenderisch vielen Ressourcen ausgestattet sind (z.B. Prozessoren mit einem Takt im Gigahertz-Bereich, viel Hauptspeicher, Strom usw.). Eingebettete Systeme verfügen meist nur über sehr eingeschränkte Ressourcen. Möchte man hier den Einsatz moderner Programmiersprachen ermöglichen, so ist es das vornehmlichste Ziel, den mit der Verwendung dieser Sprachen einhergehenden hohen Ressourcenbedarf zu reduzieren.

Am Lehrstuhl werden insbesondere die folgenden drei Teilgebiete untersucht:

**1.) Stromsparoptimierung.** Für batteriebetriebene Systeme wie z.B. Mobiltelefone oder PDAs ist es wichtig, maßvoll mit der ihnen zur Verfügung stehenden Energie umzugehen. Um Strom zu sparen, besitzen viele Prozessoren die Möglichkeit, dynamisch zur Laufzeit ihre Taktfrequenz und Versorgungsspannung herab- und wieder heraufzusetzen. Diese Technologien werden Dynamic-Frequency-Scaling (DFS) und Dynamic-Voltage-Scaling (DVS) genannt. Am Lehrstuhl wurde im Jahr 2003 erforscht, inwiefern Übersetzer und Betriebssystem zusammenarbeiten können, um diese Technologien auszunutzen. Führt der Prozessor Programmteile aus, die viele Daten aus dem Speicher laden bzw. hineinschreiben, so muss er oft auf den meist wesentlich langsameren (Arbeits-)Speicher warten. Takt und Spannung können in diesem Fall herunterskaliert werden, ohne dass dies aus Sicht der Rechenzeit ins Gewicht fällt. Frühere Ansätze haben versucht, DVS entweder über das Betriebssystem oder direkt vom auszuführenden Programm aus zu realisieren. Bei unserem Ansatz wird das DVS vom Betriebssystem gesteuert, der Übersetzer bettet jedoch Anweisungen in das Programm ein, welche dem Betriebssystem Informationen über den Rechenzeitbedarf des aktuellen Programmabschnittes mitteilt. Das Betriebssystem erlangt somit Kenntnis über den aktuellen Rechenzeitbedarf einer jeden Anwendung und kann die jeweils optimale Frequenz und Versorgungsspannung bestimmen und einstellen.

**2.) Speicherbereinigung für Code.** Da der Arbeitsspeicher (neben Energie) eine weitere beschränkte Resource im Bereich eingebetteter Systeme ist, werden seit 2003 weitere Methoden untersucht, um den Arbeitsspeicherverbrauch eines Programmes zu verringern. Die Idee beruht darauf, dass nicht der gesamte Programmcode einer Anwendung komplett im Arbeitsspeicher liegen muss, um das Programm auszuführen. Das Laden des benötigten Codes wird normalerweise vom Betriebssystem übernommen, welches mittels virtuellem Speicher die Möglichkeit besitzt, Programmcode seitenbasiert in den Speicher zu laden bzw. wieder zu verdrängen. Die Größe der Seiten liegt dabei meist im Bereich mehrerer Kilobytes. Die Seiten enthalten mehr als nur den minimal benötigten Code. Bedingt durch ihre Architektur besitzen eingebettete Systeme oft keine Einheit zur Verwaltung von virtuellem Speicher oder sie müssen ganz ohne ein Betriebssystem auskommen. Die am Lehrstuhl entwickelte automatische Speicherverwaltung für Programmcode lädt den Code nicht seitenweise in den Speicher, sondern es werden nur kleine Code-Fragmente (Basisblöcke) geladen, welche garantiert vollständig zur Ausführung kommen. Wenn es zu einem Speicherengpass kommt, wird nicht mehr benötigter Code automatisch erkannt und aus dem Speicher entfernt. Die Laufzeitinformationen, welche die Speicherverwaltung über das auszuführende Programm hat, können verwendet werden, um das Programm dynamisch zu optimieren. Das Potential dieser Optimierungen soll in der Zukunft untersucht werden.

**3.) RAM-Optimierung.** Außerdem wurde im Jahr 2003 untersucht, inwieweit im Bereich gewisser eingebetteter Systeme eine engere Zusammenarbeit von Übersetzern und statischen Betriebssystemen Vorteile bei Effizienz und Sicherheit bringt. Das Interesse liegt hier speziell auf kleinen eingebetteten Systemen, die in Hochsprachen wie C programmiert werden und statische Echtzeitbetriebssysteme benutzen. Solche Systeme werden z.B. in der Automobilindustrie verstärkt eingesetzt. Einsparungen von einigen Dutzend Bytes können aufgrund der hohen Stückzahlen insgesamt Kosten in Millionenhöhe einsparen. Eine Studie, die den Overhead eines C-Übersetzers gegenüber direkt in Assembler entwickelten Programmen auf 100 - 1000% schätzt, zeigt das noch große Einsparungspotential durch Optimierungen zur Übersetzungszeit. Konkret wurden drei Möglichkeiten untersucht, den RAM-Verbrauch solcher Systeme zu reduzieren:

- Der Einfluss klassischer Compileroptimierungen auf den RAM-Verbrauch wurde gemessen. Eine verbesserte Auswahl aus existierenden Optimierungen hinsichtlich dieses Kriteriums konnte den RAM-Verbrauch einer kleinen Testreihe um fast 20% senken.
- Modulübergreifende Stackanalyse im Compiler wurde mit einem kommerziell erhältlichen

Post-Linktime Analysewerkzeug verglichen. Testergebnisse zeigen eine verbesserte Handhabung und höhere Zuverlässigkeit als die existierenden Ansätze. Durch statische Analyse des worst-case Stackverbrauchs können der Verschnitt an Stackspeicher minimiert und das Risiko für Stacküberläufe fast eliminiert werden.

- Als neues Verfahren wurden die von statischen Betriebssystemen gespeicherten Taskkontexte (d.h. Registersätze, die bei Unterbrechung eines Tasks gesichert werden müssen) minimiert. Für jeden Task werden möglichst wenige Register gesichert (die sowohl Werte enthalten als auch überschrieben werden können) und auch nur so viel RAM reserviert. Unsere Inter-Task Registervergabe erweitert dieses Verfahren noch.

Eine Implementierung für ein kommerziell in der Automobilindustrie eingesetztes Betriebssystem zeigt die Machbarkeit dieser Verfahren. Es gelang, sowohl die Handhabung zu verbessern als auch den Speicherbedarf einiger Benchmarks um meist 30%-60% zu reduzieren.

## 3.2 Cluster Computing

### **Projektleitung:**

Prof. Dr. Michael Philippsen

### **Beteiligte:**

Ronald Veldema, Ph.D., Dipl.-Inf. Michael Klemm

**Laufzeit: 1.4.2002 - 31.12.2006**

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

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

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

**Programmiersprache, Systementwurf:** Programmierer haben in der Regel das anwendungsspezifische Hintergrundwissen über bestehende Lokalitätszusammenhänge. Um guten Code zu erzeugen, muss der Übersetzer auch über dieses Wissen verfügen. Im Idealfall sollte der Übersetzer durch statische und dynamische Analyse in der Lage sein, das Wissen aus dem Programm-Code zu extrahieren - dies ist außer in datenparallelen Sprachen bisher leider kaum möglich. Der alternative Ansatz, besteht im Prinzip darin, dass der Programmierer sein Wissen explizit, z.B. mit Hilfe von leicht zu entdeckenden Pragmas, an den Übersetzer weitergibt. Of-

fen ist, wie man diese Wissensweitergabe sinnvoll ausgestaltet. Sinnvoll bedeutet dabei erstens, dass es für den Programmierer bequem bleiben muss, und zweitens, dass das codierte/entdeckte Wissen aussagekräftig genug sein muss, um vom Übersetzer oder dem Laufzeitsystem effizienzsteigernd genutzt werden zu können.

Noch sehr schlecht verstanden ist generell, wie man verteilte Systeme von vorneherein so entwirft, dass Lokalität per Design vorhanden ist.

Im Jahr 2003 konzentrierten sich die Arbeiten auf neue Optimierungstechniken im Übersetzer, um einige der genannten Probleme zu lösen. Aus früheren Ergebnissen ist bekannt, wie man entfernte Methodenaufrufe im allgemeinen effizient realisiert. Im Jahr 2003 verfolgten wir die Idee, für jede einzelne Aufrufstelle spezifisch optimierten Code zu erzeugen. Diese aufrufstellenspezifische Optimierung entfernter Methodenaufrufe konnte die Leistung nochmals um bis zu 20% steigern. Allerdings ist es manchmal besser, statt eines entfernten Aufrufs lokal zu arbeiten und stattdessen die benötigten Daten herbei zu holen. Zu dieser Frage wurde mit Heuristiken experimentiert. Als weiteres wichtiges Thema standen 2003 die heterogenen Rechnerbündel im Fokus. Es gelingt uns nun ohne Zusatzaufwand zur Laufzeit, ein gegebenes Programm so umzubauen, dass es an beliebiger Stelle von einem Knoten des Grids auf einen anderen Knoten des Grids umziehen kann, obwohl dieser eine völlig andere Hardware-Architektur hat.

### **3.3 Graphen und Graphtransformationen**

**Projektleitung:**

Prof. em. Dr. rer. nat. Hans Jürgen Schneider

**Beteiligte:**

Dr.-Ing. Ingrid Fischer, Prof. Dr. Mark Minas

**Laufzeit: 1.1.1995 - 1.10.2006**

**Förderer:**

Bayerisch-Kalifornisches Hochschulzentrum

**Mitwirkende Institutionen:**

Data Analysis Research Laboratory, Tripos Inc., San Francisco, USA

Graphen werden an vielen Stellen als intuitives Hilfsmittel zur Verdeutlichung komplizierter Sachverhalte verwendet. Außerhalb der Informatik trifft dies z.B. auf die Biologie oder Chemie zu, wo Moleküle graphisch modelliert werden. Innerhalb der Informatik werden Daten- bzw. Kontrollflussdiagramme, als Entity-Relationship-Diagramme oder Petri-Netze, zur Visualisierung sowohl von Software- als auch von Hardware-Architekturen häufig verwendet. Graphgrammatiken und Graphtransformationen kombinieren Ideen aus den Bereichen Graphentheorie, Algebra, Logik und Kategorientheorie, um Veränderungen an Graphen formal zu beschreiben.

Am Lehrstuhl für Programmiersysteme werden Graphen und Graphtransformationen in verschiedenen Bereichen eingesetzt.

- Data Mining auf Graphdatenbanken ist ein Gebiet, auf dem es in den letzten Jahren viele Entwicklungen gab. In Zusammenarbeit mit dem Data Analysis Research Laboratory der Tripos Inc., San Francisco, USA, wurde begonnen, zwei existierende Algorithmen auf diesem Gebiet zu kombinieren und zu parallelisieren. Desweiteren wurde versucht, Unschärfen in die Algorithmen zu integrieren.
- Eine Graphgrammatik kann benutzt werden, um eine Menge syntaktisch korrekter Diagramme zu definieren, d.h. Diagramme, die nach den Regeln eines bestimmten Anwendungsgebietes aufgebaut sind. Graphtransformationen erlauben dynamische Veränderungen derartiger Darstellungen und somit die Beschreibung der Entwicklung von Strukturen.
- Die zugrundeliegende Theorie ist ein attraktives Hilfsmittel, äußerst unterschiedliche Strukturen in einer einheitlichen Weise zu beschreiben, z.B. die unterschiedlichen Modelle für asynchrone Prozesse: Petri-Netze basieren auf "gewöhnlichen" Graphen, Statecharts verwenden hierarchische Graphen, die parallele logische Programmierung kann mit Hilfe sogenannter Dschungel graphentheoretisch interpretiert werden, und die Aktorsysteme lassen sich als Graphen darstellen, deren Markierungsalphabet eine Menge von Termgraphen ist.
- Im Zusammenhang mit den Arbeiten an einem neuen Lehrbuch über den kategoriellen Ansatz wurde im Jahr 2003 vornehmlich an einer Vereinheitlichung der effektiven Konstruktion der Ableitungsschritte gearbeitet und insbesondere die Situation bei nichteindeutigen Pushout-Komplementen untersucht. Dabei stellte sich heraus, dass die früher von Rosen und Parisi-Presicce betrachteten Normierungen Spezialfälle einer allgemeinen Konstruktion sind: Man kennt alle denkbaren Lösungen, wenn man das in gewissem Sinne minimale Pushout-Komplement konstruieren kann. In den interessanten Kategorien ist das der Fall.

### **3.4 Automatische Analyse natürlicher Sprache**

**Projektleitung:**

Dr.-Ing. Ingrid Fischer

**Beteiligte:**

Prof. Dr. Sylvia Weber Russell, Dipl.-Inf. Ricarda Dormeyer

**Laufzeit: 1.1.1999 - 31.12.2004**

**Förderer:**

Deutscher Akademischer Austauschdienst  
Fulbright Scholar Grant

**Mitwirkende Institutionen:**

University of New Hampshire

Die interessantesten und schwierigsten idiomatischen Beispiele enthalten lexikalische und syntaktische Modifikationen (z.B. "Er schloss die eiternde Wunde, in die er Salz gestreut hat"). Aber auch unveränderliche Idiome (z.B. "Er hat ins Gras gebissen") dürfen bei der automatischen Sprachanalyse nicht vernachlässigt werden. Der verfolgte Ansatz zur Idiomanalyse beruht auf zwei Vermutungen. Erstens können Idiome analysierbar sein, das heißt, die einzelnen Komponenten können bei der Analyse des Idioms eine Rolle spielen, müssen es aber nicht. Zweitens spielen bei vielen Idiomen Metaphern eine aktive Rolle im Verstehensprozess, besonders wenn die Idiome analysierbar und modifizierbar sind. Diese Vermutungen werden an verschiedenen Sprachverarbeitungssystemen getestet.

Im Jahre 2003 wurde vor allem der diskontinuierliche Charakter von Idiomen untersucht. Idiome können nicht als lange "Wörter" in ein Lexikon eingetragen werden, ihre einzelnen Bestandteile müssen erkennbar bleiben. Damit ähneln sie anderen diskontinuierlichen Konstituenten wie trennbaren Verbpräfixen oder Relativsätzen im Nachfeld. Für derartige Phänomene wurden zwei verschiedene Mechanismen untersucht:

- Mit Hilfe einer eingeschränkten Form von Hypergraphgrammatiken können Sprachen erzeugt werden, die normalerweise kontextsensitiv sind. Die Ersetzung von Hyperkanten lässt sich auf diskontinuierliche Konstituenten in natürlichen Sprachen problemlos übertragen und führt zu übersichtlichen und einfachen Grammatikregeln.
- Der am Lehrstuhl entwickelte Parser für Dependenzgrammatiken wurde in bezug auf diskontinuierliche Konstituenten erweitert und verbessert. Dies betrifft sowohl den Grammatikformalismus als auch das eigentliche Parsen der diskontinuierlichen Konstituenten.

### 3.5 Entwicklung und Anwendung von Evolutionären Algorithmen

**Projektleitung:**

PD Dr. Gabriella Kókai

**Beteiligte:**

Hans-Holm Frühauf, Xu Feng

**Beginn: 1.1.1999**

**Förderer:**

Bayerisches Staatsministerium für Wissenschaft, Forschung und Kunst  
Deutscher Akademischer Austauschdienst  
Frauenförderung aus dem Hochschul- und Wissenschaftsprogramm (HWP)

**Mitwirkende Institutionen:**

Fraunhofer-Institut für Integrierte Schaltungen IIS-A,  
Abteilung Hochfrequenz- und Microwellentechnik  
Fraunhofer Institut für Integrierte Schaltungen IIS-B,  
Abteilung Simulation/Topographie  
Universität Szeged

Das genetisch-logische Programmiersystem GeLog kombiniert Methoden genetischer Algorithmen und induktiv logischer Programmierung. Mit dem automatischen Lernsystem können

aus einer induktiv logisch formulierten Aufgabenstellung logische Programme erzeugt werden, die die gegebene Aufgabe lösen. Zum Lernen verwendet GeLog einen genetischen Algorithmus. GeLog verfügt über automatische Parameteradaptation und Metaevolution und vermeidet dadurch die bekannten Nachteile genetischer Algorithmen, wie die große Anzahl von Einstellmöglichkeiten für Evolutionsparameter sowie das Hängenbleiben in lokalen Optima.

Um dem Benutzer die Arbeit mit dem GeLog System möglichst einfach zu machen, wurde im Jahr 2003 eine graphische Benutzeroberfläche konstruiert. Außerdem wurde eine graphische Darstellung der Ergebnisse von GeLog ins System integriert. Im GeLog ist nun alles zentral bedienbar und komfortabel für den Benutzer. Bei der graphischen Illustration ist besonders die Möglichkeit der vergleichenden Darstellung unterschiedlicher GeLog-Durchgänge zu erwähnen.

Im Jahr 2003 gelang es uns erneut, genetische Algorithmen im Rahmen eines Kooperationsprojekts mit der Industrie einzusetzen. Wir entwickelten einen hardware-basierten genetischen Optimierer (HGA) zur Steuerung eines Empfängers mit intelligenten Antennen. Das Ziel der Arbeit war es, eine HGA-basierte Regelung mehrerer Hochfrequenztechnik-Komponenten der Empfängerschaltung für eine adaptive Gruppenantenne zu entwickeln. Die entwickelte Architektur des Optimierers ermöglicht es, durch die Kombination von Pipelining und Parallelisierung, eine Steigerung der Rechenleistung zum Vergleich zu Software-Implementation zu erzielen. Die Anwendung von HGA war nötig, weil wir eine sehr harte Zeitbedingung erfüllen müssen: die neue optimale Parametereinstellung der Antenne muss innerhalb von 42ns ermittelt werden. Das vorgeschlagene HGA-Modell wurde für ein Fraunhofer-R2-Board, mit einem Xilinx Virtex-E XCV600E FPGA bestückt, prototypisch implementiert. Bereits dabei wurde ein Beschleunigungsfaktor von über 500 im Vergleich zur ebenfalls durchgeführten Software-Implementierung erreicht.

### **3.6 Funktions-Software-Entwicklung für elektronische Steuergeräte im Automobilbereich**

#### **Projektleitung:**

PD Dr. Gabriella Kókai

#### **Beteiligte:**

Prof. Dr. Michael Philippsen, Dipl.-Inf. (FH) Bernd Hardung

**Laufzeit: 1.1.2003 - 31.12.2006**

#### **Förderer:**

Audi AG, Ingolstadt

Die Herausforderungen im Bereich der Funktions-Software-Entwicklung für elektronische Steuergeräte im Automobilbereich sind vielfältig. Auf der einen Seite ist das Thema Software-Entwicklung in dieser Domäne relativ neu, auf der anderen Seite ist man mit hohem Kostendruck und Qualitätsanforderungen konfrontiert. Um diesen offensichtlichen Widerspruch zu lösen, muss zunächst die spezielle Herausforderung der Wiederverwendbarkeit von Funktions-Software gelöst werden. In dieser Arbeit sollen speziell folgende zwei Punkte bearbeitet werden: 1.) Software-Integration von unterschiedlichen Herstellern: Funktionen und Code verschiedener Zulieferer sollen in Zukunft auf einem Steuergerät integriert werden können. Dies

bedingt unter anderem einen entsprechenden Funktionsentwicklungsprozess beim Automobilhersteller. 2.) Methode und Prozess zum Betrieb einer Funktionsdatenbank: Aus dieser Datenbank sollen Modelle und zielsystemspezifischer Code extrahiert werden können, um sie über Baureihen und unterschiedliche Hardware-Architekturen hinweg wiederverwenden zu können.

### **3.7 SIGNAL - Sofortprogramm Informatik am Gymnasium - Nachqualifikation von Lehrkräften (ehemals NELLI)**

#### **Projektleitung:**

Prof. Dr. Michael Philippsen

#### **Beteiligte:**

Forman, Franz, Dr. Klaus Kreisel, Dr. Hermann Puhlmann

**Laufzeit: 1.9.2002 - 31.8.2005**

#### **Förderer:**

Bayerisches Staatsministerium für Unterricht und Kultus  
Bayerisches Staatsministerium für Wissenschaft, Forschung und Kunst  
Europäische Kommission

#### **Mitwirkende Institutionen:**

Ludwig-Maximilian-Universität München (Prof. Dr. Hofmann, Prof. Dr. Kröger)  
Technische Universität München (Prof. Dr. Hubwieser)  
Universität Passau (Prof. Dr. Freitag)

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

Der immense Bedarf an Lehrkräften kann nicht über die traditionelle Lehramtsausbildung an den Universitäten gedeckt werden. Aus diesem Grund wurde im Jahr 2000 das Projekt Nelli (Netzgestützter Lehrverbund zur Lehrerausbildung in Informatik) gestartet und ein elektronisches Angebot von speziell zugeschnittenen Studien- und Unterrichtsmaterialien entwickelt. Der Lehrstuhl Informatik 2 hat hierzu das Algorithmik-Modul beigetragen. Im September 2001 nahmen bayernweit im Rahmen von Pilotkursen 75 Lehrkräfte das Studium der Informatik auf, davon 15 in Erlangen. Im Herbst 2003 haben diese Lehrerinnen und Lehrer die Erweiterungsprüfung in Informatik abgelegt.

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

Im ersten Kursjahr lernen die Teilnehmer überwiegend zu Hause mit elektronisch bereitgestellten Materialien. Sie geben über einen Gruppenarbeits-Server Übungen ab, die vom Tutor des Kurses korrigiert werden. Zum Erfahrungsaustausch und persönlichen Kontakt finden zudem monatliche Treffen statt. Im zweiten Kursjahr finden wöchentliche Präsenzveranstaltungen an der Universität statt. Dabei besuchen die Teilnehmer auch reguläre Veranstaltungen aus dem Angebot für das Informatikstudium.

Darüberhinaus wird seit Herbst 2003 das reguläre Lehramtsstudium Informatik für alle Schulformen angeboten; dazu wird eine Professur **Didaktik der Informatik** am Lehrstuhl Informatik 2 eingerichtet werden. Siehe Informationsseite (<http://www2.informatik.uni-erlangen.de/~philippsen/lehramt/>)

Im Vergleich mit den anderen Bundesländern wird Bayern nach Abschluss dieses Programms einen Spitzenplatz in Bezug auf den Stellenwert der Informatik an den Schulen einnehmen.

## 4 Veröffentlichungen und Vorträge

### 4.1 Veröffentlichungen

1. Z. Tóth: A Graphical User Interface for Evolutionary Algorithms. In: Journal Acta Cybernetica (2003), Nr. 16, S. 337-365
2. Bottoni, P.; Minas, M. (Hrsg.): Proc. of Graph Transformation and Visual Modeling Techniques. (GT-VMT'2002 Barcelona, Spain 11.-12.10.2002) Bd. 72/3 Electronic Notes in Theoretical Computer Science. Amsterdam : Elsevier Science Publishers, 2003. - elektronische Ausgabe, daher keine Seiten. ISBN 0444514082
3. Dormeyer, R.; Fischer, Ingrid; Weber-Russell, S.: A lexicon for Metaphors and Idioms. In: Deutschen Gesellschaft für Sprachwissenschaft Sprache, Wissen, Wissenschaft (Hrsg.) : 25. Jahrestagung (Arbeitsgruppe Semantisches Wissen im Lexikon Ludwig-Maximilian-Universität München 26.-28. Februar 2003). 2003, S. 105.
4. Drewes F.; Hoffmann B.; Minas M.: Context-exploiting shapes for diagram transformation. In: Machine Graphics & Vision 12 (2003), Nr. 1, S. 117-132
5. Erdmann, A.; Farkas, R.; Fühner, T.; Tollkühn, B.; Kókai, Gabriella: Mask and Source Optimization for Lithographic Imaging Systems. In: Frank Wyrowski (Hrsg.) : Proc. SPIE Annual Meeting (Wave-Optical Systems Engineering II San Diego, CA 08.2003). Bd. 5182. 2003, S. 88-102.
6. Farkas, R.; Kókai, Gabriella; Erdmann, A.; Fühner, T.; Tollkühn, B.: Optimization of one- and two dimensional masks in the optical lithography. In: GI (Hrsg.) : Treffen der Fachgruppe Maschinelles Lernen, Wissenentdeckung, Data Mining der Gesellschaft für Informatik (FGML 2003 Karlsruhe 06-08.10.2003). 2003, S. 56-64.
7. Fischer, Ingrid: Modelling Discontinuous Constituents with Hypergraph Grammers. In: J. Pfalz; M. Nagl; B. Böhlen (Hrsg.) : Proc. International Workshop on Applications of Graph Transformation with Industrial Relevance (AGTIVE'03 Charlottesville, USA 28.09.-01.10.2003). 2003, S. 175-180.
8. Fühner, T.; Kókai, Gabriella: Incorporating Linkage Learning into the GeLog Framework. In: Journal Acta Cybernetica 16 (2003), S. 209-228
9. Gröbner, M.; Wilke, Peter; Büttcher, S.: A Standard Framework for Timetabling Pro-

- blems. In: Burke, Edmund; Causmaecker, Patrick De (Hrsg.) : Selected Revised Papers from the 4th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2002 Gent, Belgien 21.08-23.08.02). Bd. 2740 Lecture Notes in Computer Science. Heidelberg : Springer Verlag, 2003, S. 24-38.
10. Gussmann, W.; Puhlmann, Hermann: Beispiele zu Haskell - Funktionale Programmierung zur Spezifikation und als Ergänzung des imperativen Paradigmas in der Schule. In: LOG IN (2003), Nr. 124, S. 64-69
  11. Hardung, Bernd; Wernicke, M.; Krüger, A.; Wagner, G.; Wohlgemuth, F.: Entwicklungsprozess für vernetzte Elektroniksysteme. In: VDI (Veranst.) : Elektronik im Kraftfahrzeug (11. Internationale Konferenz Baden-Baden 25.09.-26.09.2003). Bd. VDI-Bericht 1789. 2003, S. 77-97.
  12. Kókai, Gabriella: Development of methods how to avoid the overfitting-effect within the GeLog-system. In: Vasile Palade; Robert J. Howlett; Lakhmi C. Jain (Hrsg.) : Proc. Seventh International Conference on Knowledge-Based Intelligent Information Engineering Systems (KES'2003 University of Oxford, United Kingdom 03.-05.09.2003). Bd. 2774 Lecture Notes in Computer Science. Heidelberg : Springer Verlag, 2003, S. 958-966.
  13. Minas, M.: Bootstrapping visual components of the DiaGen specification tool with DiaGen. In: J. Pfaltz; M. Nagl; B. Böhlen (Hrsg.) : Proc. International Workshop on Applications of Graph Transformations with Industrial Relevance (AGTIVE'03 Charlottesville, USA 28.09.-01.10.2003). 2003, S. 391-405.
  14. Minas, M.: Visual specification of visual editors with DiaGen. In: J. Pfaltz; M. Nagl; B. Böhlen (Hrsg.) : Proc. International Workshop on Applications of Graph Transformations with Industrial Relevance (AGTIVE'03 Charlottesville, USA 28.09.-01.10.2003). 2003, S. 461-466.
  15. Minas, M.: XML-based specification of diagram editors. In: R. Bardohl; H. Ehrig (Hrsg.) : Proc. Uniform Approaches to Graphical Process Specification Techniques (UNIGRA'03) (Satellite Workshop to ETAPS'2003 Warschau/Polen 05.-06.04.2003). Bd. 82 Electronic Notes in Theoretical Computer Science. Amsterdam, NL : Elsevier Science Publishers, 2003, S. CD-ROM.
  16. Pang, J.; Fokkink, W.; Hoffmann, R.; Veldema, Ronald: Model Checking a Cache Coherence Protocol for a Java DSM Implementation. In: IEEE Computer Society (Hrsg.) : Proceedings of 17th International Parallel and Distributed Processing Symposium (IPDPS) (8th International Workshop on Formal Methodes for Parallel Programming: Theory and Applications (FMPPTA) Nizza, Frankreich 22.04.-26.04.03). 2003, S. CD-ROM.
  17. Prechtelt, L.; Unger, B.; Philippsen, Michael; Tichy, W.F.: A Controlled experiment on inheritance depth as a cost factor for maintenance. In: Journal of Systems and Software 65 (2003), Nr. 2, S. 115-126
  18. Puhlmann, Hermann: Informatische Literalität nach dem PISA-Muster. In: Peter Hubwieser (Hrsg.) : Informatische Fachkonzepte im Unterricht (10. GI-Fachtagung Informatik und Schule INFOS 2003 Garching 17.09.-19.09.2003). Bonn : GI Edition Lecture Notes in Informatics, 2003, S. 145-154. (Tagungsband INFOS 2003)
  19. Tollkühn, B.; Fühner, T.; Matiut, D.; Erdmann, A.; Semmler, A.; Küchler, B.; Kókai, Gabriella: Will Darwins's Law Help Us to Improve Our Resist Models? In: Theodore H. Fedynyshyn (Hrsg.) : Advances in Resist Technology and Processing XX (SPIE 5039 Santa Clara, USA 02.2003). 2003, S. 291-302.

20. Vanyi, Robert; Zvada, Szilvia: Avoiding Syntactically Incorrect Individuals via Parameterized Operators Applied on Derivation Trees. In: R. Sarker et al. (Hrsg.) : Proc. of the Congress on Evolutionary Computation (CEC2003 Canberra, Australien 08.-12.12.2003). Bd. 4. 2003, S. 2791-2798.
21. Veldema, Ronald; Philippsen, Michael: Compiler Optimized Remote Method Invocation. In: IEEE Computer Society (Veranst.) : Proc. 5th IEEE Conf. on Cluster Computing (CC 2003 Hong Kong 01.-04.12.2003). 2003, S. 127-136.

## **4.2 Vorträge**

- 29.01.03 Kókai, Gabriella  
Die Anwendung aus der Biologie abgeleiteter Lernverfahren auf die Molekularbiologie  
Habitationskolloquium, Universität Erlangen-Nürnberg, Erlangen
- 05.04.03 Philippsen, Michael  
Konzeption eines Lehramtsstudiengangs Informatik für alle Schulformen  
Landesvorstandssitzung Bayerischer Realschullehrerverbandg, Lauf
- 15.04.03 Puhlmann, Hermann  
Implementation kryptographischer Verfahren mit funktionaler Programmierung, Bundeskongress des Vereins zur Foerderung des mathematisch-naturwissenschaftlichen Unterrichts (MNU), Frankfurt/Main
- 02.05.03 Philippsen, Michael  
Überblick über JavaParty und die erforderlichen Optimierungen  
Informatik-Kolloquium, Universität Klagenfurt, Österreich

## **5 Habilitationen**

- 29.01.03 Kókai, Gabriella  
Erfolge und Probleme evolutionärer Algorithmen, induktiver logischer Programmierung und ihrer Kombination

## **6 Promotionen**

- 24.7.03 Gröbner, Matthias  
Ein Modell zur Beschreibung und Lösung von Zeitplanungsproblemen

## **7 Studien- und Diplomarbeiten**

### **7.1 Studienarbeiten**

- Böhm, Jan Felix: Hypergraphgrammatiken zur Analyse natürlicher Sprachen
- Tonka, Merlin: Möglichkeiten und Grenzen von J2ME

- Poppe, Carsten: Umsetzung XML-basierter Drag and Drop-Übungstypologien im Fremdsprachenerwerb mit Java

## **7.2 Diplomarbeiten**

- Ranger, Ulrike: Ein PATR-II-basierter Chart-Parser zur Analyse von Idiomen
- Summa, Oliver: Entwicklung einer graphischen Oberfläche für das GeLog-System
- Klemm, Michael: Dynamic Function Splicing in Jackal
- Tröger, Thomas: Ein Chartparser für natürliche Sprache auf der Grundlage der Abhängigkeitsgrammatik
- Ouadday, Chamseddine: Evaluierung von Content Management Systemen für den Einsatz im Intranet (von IT-Dienstleistern)
- Schnattinger, Oliver: Vergleich und Bewertung von Softwarebibliotheken für Evolutionäre Algorithmen
- Reiber, Andreas: Visuelle Spezifikation von Diagrammeditoren mit DiaGen

## **8 Bachelor- und Master Theses**

### **8.1 Master Theses**

- Xu, Feng: Integration, Simulation and Implementation of a Hardware Based Genetic Optimizer in Order to Adjust a Smart Antenna Receiver