

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

Honorarprofessoren:

Hon.-Prof. Dr.-Ing. Bernd Hindel (seit 01.10.2005)

Hon.-Prof. Dr.-Ing. Detlef Kips (seit 01.10.2005)

Emeritus:

Prof. em. Dr. Hans Jürgen Schneider

Sekretariat:

Agnes Brütting

Waltraud Rück

Elfriede Schörmal

Wiss. Mitarbeiter:

Dr. -Ing. Ingrid Fischer

Dipl.-Inf. Michael Klemm

Dipl.-Inf. Thorsten Meinel

Dipl.-Inf. Dominic Schell

Ronald Veldema, Ph.D.

PD Dr.-Ing. habil. Peter Wilke

Gäste/Stipendiaten:

Franz Forman

Dipl.-Ing. Hans Holm Frühauf

Dipl.-Ing. (FH) Bernd Hardung

Dr. Klaus Kreisel

PD Dr.-Ing. Gabriella Kókai

Dr. Hermann Puhmann

Dipl.-Inf. Szilvia Zvada

Externes Lehrpersonal:

Dr. rer. nat. Ralph Kummetz

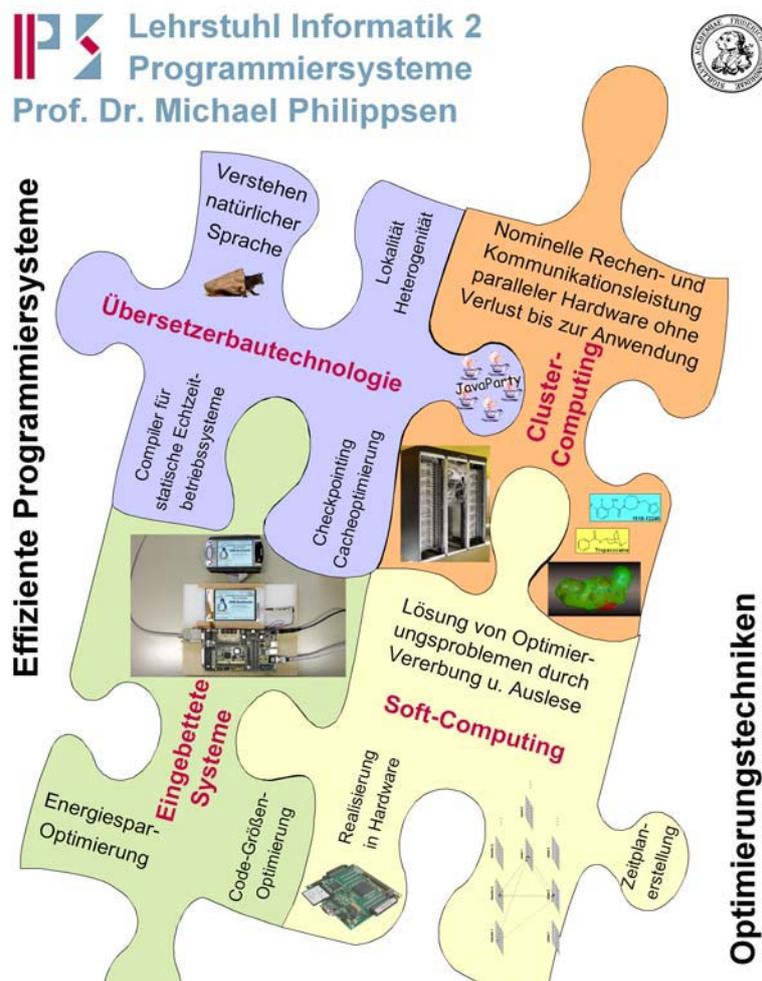
Dr.-Ing. Jörg Nilson

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 portabel, wartbar und robust sein, wie heute für Einprozessorsysteme und Arbeitsplatzrechner. Langfristiges Ziel ist es, den 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 Cluster Computing

Projektleitung:

Prof. Dr. Michael Philippsen

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

Laufzeit: 1.4.2002 - 31.12.2006

Kontakt:

Prof. Dr. Michael Philippsen

Tel.: +49.9131.85.27625, Fax: +49.9131.85.28809,

E-Mail: philippsen@informatik.uni-erlangen.de

Das Programmieren von Rechnerbündeln befindet sich derzeit noch in einer sehr maschinen-nahen 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. Offen 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 2005 wurde an neuen Funktionen für den Lehrstuhl-eigenen Übersetzer entwickelt.

- Ein verteiltes Sicherungsverfahren erlaubt es, eine im Rechnerbündel verteilt ausgeführte Applikation zu sichern und wiederherzustellen. Damit können Ausfälle eines Teils des Rechnerbündels abgefangen und der Verlust von Rechenzeit minimiert werden.
- Das Laufzeitsystem des DSM-Übersetzers wurde um eine eigene Kommunikationsbibliothek erweitert. Diese erlaubt die Kommunikation in heterogenen Rechnerbündeln mit unterschiedlichster Kommunikations-Hardware. Neben maximaler Effizienz wurde beim Design der Bibliothek verstärkt darauf geachtet, dass sie auch die Anforderungen eines Computational Grids vollständig unterstützt (u.a. Authentifizierung und Datenverschlüsselung).

3.2 Parallele Suche in Moleküldatenbanken

Projektleitung:

Prof. Dr. Michael Philippsen

Beteiligte: Dr.-Ing. Ingrid Fischer, Dipl.-Inf. Thorsten Meinl, Dipl.-Inf. Marc Wörlein
Dipl.-Inf. Alexander Dreweke, B. Sc. mult.

Beginn: 1.4.2004

Förderer: Bayerisch-Kalifornisches Hochschulzentrum

Kontakt:

Prof. Dr. Michael Philippsen
Tel.: +49.9131.85.27625, Fax: +49.9131.85.28809,
E-Mail: philippsen@informatik.uni-erlangen.de

Das Forschungsprojekt **ParMol** befasst sich mit der verteilten Suche häufiger Fragmente in Moleküldatenbanken. In seinem Rahmen sollen neue Verfahren entwickelt werden, um in großen Datenbanken interessante Zusammenhänge zwischen chemischen Strukturen zu finden. Diese Zusammenhänge sollen als Molekülfragmente, d.h. in Form von Zusammenhangsgraphen dargestellt werden. Ein Anwendungsgebiet ist z.B. die in vitro Vorhersage von toxischen Seiteneffekten von Medikamentenkandidaten schon vor in vivo oder klinischen Tests. Im Gegensatz zu bisher bekannten Verfahren sollen auch ähnliche Strukturen gefunden werden, die biologisch oder chemisch die gleiche Wirkung zeigen. Um den dadurch entstehenden deutlich höheren Rechenbedarf zu decken, werden die entwickelten Verfahren parallelisiert und so entworfen, dass sie sich auch für die Bearbeitung auf verteilten Ressourcen eignen.

Im Jahr 2005 wurden in einigen Studien- und Diplomarbeiten die bekannten sequentiellen Graphminer in einem einheitlichen Framework implementiert. Durch intensive Vergleiche konnte ein besseres Verständnis der einzelnen Verfahren gewonnen werden, das für die anstehende Parallelisierung sehr hilfreich ist. Darauf aufbauend wurde der erste Graphminer (MoFa) parallelisiert und auf dem 28-CPU-Altix-Rechner im Rechenzentrum ausgiebig getestet. Leider sind die durch Parallelisierung erreichten Laufzeiten noch unbefriedigend, so dass hier noch mehr Forschungsarbeit von Nöten ist.

Um ein besseres Verständnis für die Suchergebnisse zu bekommen und um den Fortschritt der Suche zu Visualisieren, wurde ein Molekülvisualisierer implementiert, der sich homogen in das bereits vorhandene Framework eingliedert.

In Zusammenarbeit mit dem PARES-Projekt wurden die sequentiellen Graphminer außerdem so erweitert, dass sie häufige Strukturen in von Compilern erzeugten Datenflussabhängigkeiten finden. Die dazugehörigen Codestücke können dann in separate Prozeduren ausgelagert werden, der Code wird somit verkleinert.

3.3 PARES - Optimierung für eingebettete Systeme

Projektleitung:

Prof. Dr. Michael Philippsen

Beteiligte: Dipl.-Inf. Dominic Schell, Dipl.-Inf. Alexander Dreweke, B. Sc. mult.,
Dipl.-Inf. Marc Wörlein

Stichwörter: Eingebettet Eingebettete Systeme

Laufzeit: 1.2.2003 - 31.12.2006

Kontakt:

Dipl.-Inf. Dominic Schell
Tel.: +49.9131.85.27599, Fax: +49.9131.85.28809,
E-Mail: schell@informatik.uni-erlangen.de

Als besonders dringend erscheint uns gegenwärtig die Verbesserung der Programmierwerkzeuge für eingebettete Systeme. Solche Systeme werden heutzutage zu oft noch sehr maschinen-nah programmiert. Das inzwischen bei der Programmierung von Arbeitsplatzrechnern übliche Abstraktions- und Komfortniveau (Objektorientierung, automatische Speicherbereinigung, Ausnahmebehandlung, Parallelität, Aspektorientierung, Komponenten, ...) scheint im Bereich der eingebetteten Systeme noch in weiter Ferne, wodurch Portabilität, Robustheit und Wartbarkeit der erstellten Software erheblich beeinträchtigt wird. Dies ist ein erhebliches volkswirtschaftliches Problem, das gerade auch deshalb bedeutsam ist, weil Europa auf diesem Feld nicht von Amerika dominiert wird. Fernziel muss es daher sein, das Abstraktionsniveau bei der Programmierung eingebetteter Systeme schrittweise zu erhöhen, indem Optimierungstechniken entwickelt werden, die trotz des erhöhten Abstraktionsniveaus "kleinen" Code garantieren.

Neben der offensichtlichen Frage, wie die bekannten Optimierungstechniken auf die Code-Größe wirken, drängen sich neue Einzelfragen auf. Während der RAM-Bedarf einer Applikation auf Desktop-Rechnern kaum eine Rolle spielt, ist dieser für eingebettete Systeme oft essentiell. Objektorientierter - vor allem bibliotheksbasierter - Code bietet ein erhebliches, bislang ungenutztes Potential für prozedurale Abstraktion zur Code-Verkleinerung. Auf Architekturen ohne MMU muss neu über das Laden von Code nachgedacht werden. Programmpfadanalysen zur Separierung wichtiger und unwichtiger Grundblocksequenzen wird eine wachsende Bedeutung erlangen.

Neben der Code-Größe kommt auch dem Aspekt der Energie-Effizienz eine wachsende Bedeutung als Zielgröße der Code-Optimierung zu. Hier muss der Übersetzer, ggf. im Zusammenspiel mit dem Betriebssystem, optimieren bzw. auf die Hardware-Parameter einwirken.

Schließlich kommt es gerade im Bereich der vernetzten eingebetteten Systeme regelmäßig zu Schwankungen der Verfügbarkeit. Hier sind Programmierparadigmen zu finden, die die Realisierung von Anwendungen erleichtern, welche auf Fehlerzustände nicht mit Totalausfall reagieren, sondern ihr Antwortverhalten (bzgl. Laufzeit oder Ergebnisqualität) gemächlich absenken.

Die Behandlung der nicht-uniformen Speicherzugriffshierarchie, die in verteilten Systemen neben Registern, Cache und Hauptspeicher um eine weitere Leistungsebene vertieft ist, stellt auch bei eingebetteten Systemen eine Herausforderung dar, da z.B. Flash-Speicher zu berücksichtigen sind.

Können eingebettete Systeme (ebenso verteilte Systeme) - der Tradition von Desktop-Prozessoren folgend - auch weiterhin mit der Illusion eines transparenten Zugriffs programmiert werden? Kann man durch statische Analyse Informationen über bestehende Lokalitätsbeziehungen zwischen Daten extrahieren? Welche Optimierungen sind dann möglich? Profitieren statische Analyse und Laufzeitmechanismen von einander? Wie können durch Programmanalyse Pre-Fetch- und Post-Store-Kommandos im generierten Code erzeugt werden, durch die Cache-Effekte überdeckt, Wartezeiten vermieden oder Energie gespart werden?

Die Forschungsarbeit 2005 lag schwerpunktmäßig bei den folgenden beiden Teilgebieten:

- **Speicherbereinigung für Code.** Wie bereits erwähnt, steht auf eingebetteten Systemen nur wenig RAM zur Verfügung. Es muss allerdings genug RAM verfügbar sein, um sowohl die Programmdateien als auch den Code zu halten. Die gegenwärtig verfolgte 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. Im Berichtszeitraum wurden unterschiedliche Speicherbereinigungsverfahren für die Code-Bereinigung adaptiert und evaluiert. Mark&Sweep-Speicherbereiniger haben sich als nur bedingt einsetzbar erwiesen. Durch den einfachen und intuitiven Algorithmus können sie zwar mit einem geringen Code-Speicherbedarf glänzen, jedoch ist das bekannte Problem der Speicherfragmentierung ein gewichtiger Nachteil. Ebenfalls untersucht wurden kopierende Speicherbereiniger, jedoch neigen die existierenden Algorithmen zu Speicherverschwendung. Optimal erscheint uns eine Mischform der beiden Algorithmen, der Mark&Compact-Speicherbereiniger, da er sowohl das Problem Fragmentierung als auch Speicherverschwendung elegant lösen kann. Nachdem die Speicherbereinigung für Code hinreichend untersucht und in die automatische Speicherverwaltung für Programmcode integriert wurde, sollen nun die Laufzeitinformationen, welche die Speicherverwaltung über das auszuführende Programm hat, verwendet werden, um das Programm dynamisch zu optimieren. Schlagwörter sind hier Programmpfadanalysen und Energieoptimierung.
- **Graphbasierte prozedurale Abstraktion.** Eine gängige Methode zur Code-Größenverkleinerung ist die prozedurale Abstraktion (PA): gleiche Code-Abschnitte im Programm werden gesucht und daraus eine einzige neue Prozedur erzeugt. Die Code-Abschnitte werden durch Aufrufe an die neu erzeugte Prozedur ersetzt, wodurch die Redundanz innerhalb eines Programms und somit seine Größe reduziert wird. Redundanz entsteht durch die Art und Weise, wie Übersetzer Code generieren (z.B. durch Schablonen). Die bisherigen PA-Ansätze betrachten das Programm als Folge von Instruktionen und suchen nach exakt gleichen Teilfolgen. Sind allerdings Instruktionen innerhalb einer Teilfolge vertauscht, wird sie nicht als weitere Instanz der gesuchten Folge erkannt. Somit fallen potentielle Code-Fragmente

für die PA aus der Analyse heraus und das Ergebnis wird suboptimal. Der am Lehrstuhl untersuchte Ansatz löst dieses Problem indem die Instruktionen eines Grundblocks statt in einer Folge in einen Datenflussgraphen (DFG) umgesetzt werden. Ein Graph-Mining-Werkzeug sucht in den DFGs nach gemeinsamen Fragmenten. In zwei grundlegenden Diplomarbeiten wurden Konzepte zur PA auf Graphen für die Intel x86- und ARM-Plattform untersucht, wobei sich die Anzahl der auslagerbaren Fragmente gegenüber den Instruktionsfolgen erhöhen lies.

3.4 Entwicklung und Anwendung von heuristischen Optimierungsverfahren

Projektleitung:

PD Dr.-Ing. Gabriella Kókai

Beteiligte: Hans Holm Frühauf, Martin Böhner, Tonia Christ, Kornélia Danyi, József Csontos

Beginn: 1.1.1999

Mitwirkende Institutionen: Bay Zoltán Alkalmazott Kutatási Alapítvány, Fraunhofer Institute Integrierte Schaltungen

Kontakt:

PD Dr.-Ing. Gabriella Kókai
Tel.: +49.9131.85.28996, Fax: +49.9131.85.28809,
E-Mail: kokai@informatik.uni-erlangen.de

Im Jahr 2005 gelang es uns erneut, heuristische Optimierungsalgorithmen im Rahmen eines Kooperationsprojekts mit der Industrie einzusetzen. Wir entwickelten verschiedene Hardware-basierte Optimierer zur Echtzeitsteuerung von adaptiven Antennen. Moderne, intelligente Antennen erzeugen räumlich eng begrenzte Strahlenbündel, die sich während des Betriebs elektronisch gesteuert schwenken lassen und mit einer Reihe von Hardware-Parametern eingestellt werden. Diese Einstellungen haben erheblichen Einfluss auf die Empfangsqualität der Signale mobiler Sender innerhalb der Reichweite der Gruppenantenne.

Ziel der Optimierung war eine bestmögliche (messbare) Empfangsqualität gewünschter mobiler Teilnehmer und eine maximale Unterdrückung störender bzw. unerwünschter Teilnehmer. Aufgrund der hohen Mobilität der Sender ist es notwendig, die Optimierungsstrategie als hochparallele Hardware zu entwerfen, um die strengen Echtzeitbedingungen zur Nachführung der Richtcharakteristik der Antenne zu erfüllen. Innerhalb der Untersuchungen wurden die wichtigsten Verfahren des Soft-Computing auf ihre Eignung evaluiert und die erfolgreichsten implementiert. Zusätzlich zu den Algorithmen wurden alternative Hardware-Strukturen für ASIC und FPGAs entworfen und validiert.

Unter anderen wurden erstmals zur blinden adaptiven Einstellung der Richtcharakteristiken „Ameisen-Kolonie-Optimierverfahren“ (ACO) eingesetzt. Besonderer Vorteil der ACOs für das Blinde Adaptive Beamforming ist deren hohe Adaptionsrate in hochdynamischen Umgebungen. Zudem lassen sie sich einfach und sehr effizient sowohl in Software als auch in Hardware implementieren. Mittels ACO konnte die dynamische Einstellung der Richtcharakteristik

in einem Maße erreicht werden, welche die Anwendung in Echtzeitsystemen erlaubt. Dieses Verfahren wurde patentiert.

Ebenfalls erstmals wurden zur blinden adaptiven Einstellung der Richtcharakteristiken „Simulated Annealing-Optimierverfahren“ (SA) eingesetzt. Besonderer Vorteil des Simulated Annealing für das Blinde Adaptive Beamforming ist deren gutes Optimiererergebnis bei einer moderater Adaptionsrate in semi-statischen und dynamischen Umgebungen. Zudem lässt sich diese einfach und sehr effizient, sowohl in Software als auch in Hardware implementieren. Das Patentverfahren läuft.

Außerdem wurde ein Forschungsprojekt zusammen dem Bay Zoltán Forschungsinstitut Szeged mit dem Titel "Anwendung von heuristischen Optimierungsmethoden in der DNA-Chip Herstellung" beantragt und bewilligt.

Die rasche Entwicklung in Technik und Informatik hat zur Entstehung eines neuen Forschungsgebiets, der Bioinformatik, geführt. Dieses bringt enorme Fortschritte für die Medizin (z.B. neuartige Medikamente). Eine der bedeutendsten Entwicklungen ist der DNA-Chip, der eine revolutionäre Neuheit auf dem Gebiet der medizinischen Diagnostik darstellt. Mit ihm kann der Nachweis der Anwesenheit von Krankheitserregern enorm beschleunigt werden. Gleichzeitig ist die Genauigkeit in der Bestimmung der Krankheitserreger in nie gekannter Präzision möglich.

Die Weiterentwicklung der Methode wird heute weniger durch die technischen Möglichkeiten als die hohen Kosten begrenzt. So kostet beispielsweise die Herstellung 80 gleichartiger Chips mit 20.000 Sondennukleotiden heute ungefähr 40.000 Euro. Der größte Teil der Kosten ergibt sich aus der Herstellung der sogenannten Masken. Deshalb verdient der Planungsprozess dieser Masken besondere Aufmerksamkeit. Das Ziel des gemeinsamen Projekts ist die Optimierung dieses Planungsprozesses unter Kostenaspekten.

3.5 Optimierung der Funktionsverteilung im Fahrzeugnetzwerk

Projektleitung:

PD Dr.-Ing. Gabriella Kókai

Beteiligte: Dipl.-Ing. (FH) Bernd Hardung

Laufzeit: 1.1.2003 - 31.12.2006

Förderer: Audi AG, Ingolstadt

Kontakt:

PD Dr.-Ing. Gabriella Kókai

Tel.: +49.9131.85.28996, Fax: +49.9131.85.28809,

E-Mail: kokai@informatik.uni-erlangen.de

Moderne Automobile haben komplexe, verteilte Systeme von elektronischen Steuergeräten, die mittels verschiedener Bussysteme vernetzt sind. Bis zu 80 Netzknoten befinden sich in einem einzelnen Fahrzeug. Die Steuergeräte realisieren eine große Anzahl von vernetzten Funktionen, die sich aus Software- und Hardware-Komponenten zusammensetzen. In diesem Projekt werden Möglichkeiten untersucht, Funktionen bei der Erstellung der Systemarchitektur des Fahrzeugnetzwerks unterschiedlich zu platzieren, ohne das Ziel ihrer Wiederverwendung zu gefährden. Anschließend wird untersucht, nach welchen Kriterien Funktionen optimal in diesem Netz-

werk verteilt werden können. Dazu werden verschiedene Verteilungskriterien wie z. B. Kosten oder Komplexität beleuchtet. Weiterhin wird eine Methode entwickelt, die es darauf aufbauend erlaubt, Funktionsverteilungen in gegebenen Netzwerktopologien zu bewerten und zu optimieren. Die Ergebnisse werden in einem Optimierungssystem implementiert und auf ihre Praxistauglichkeit hin überprüft.

2005 wurde ein Optimierungsrahmenprogramm implementiert, welches die Optimierung der Funktionszuordnung erlaubt. Die wichtigsten Verteilungskriterien wurden untersucht, implementiert und in das Rahmenprogramm integriert. Während der Untersuchung der Verteilungskriterien wurde eine Methode zur Optimierung von Varianten von Steuergeräten entwickelt, welche auf zwei Konferenzen publiziert wurde. Weiterhin wurde ein Datenmodell zur Speicherung aller erforderlichen Daten ausgearbeitet. Es wurden evolutionäre Algorithmen entwickelt, implementiert und in das Optimierungsrahmenprogramm integriert. Deren Optimierungsqualität wurde anhand eines realistischen Anwendungsbeispiels validiert. Um ein bestmögliches Optimierungsergebnis zu erhalten, wird momentan die Tauglichkeit von Ameisenkolonieoptimierungsalgorithmen untersucht.

3.6 Automatische Analyse natürlicher Sprache

Projektleitung:

Prof. em. Dr. Hans Jürgen Schneider

Beteiligte: Dr.-Ing. Ingrid Fischer

Laufzeit: 1.1.1999 - 28.2.2006

Kontakt:

Prof. em. Dr. Hans Jürgen Schneider

Tel.: +49.9131.85.27620, Fax: +49.9131.85.28809,

E-Mail: schneider@informatik.uni-erlangen.de

Zwei Problemfelder beim Parsen natürlicher Sprachen sind freie Wortstellung und diskontinuierliche Konstituenten. Freie Wortstellung ist im Deutschen auf Satzebene möglich. "Dieser Mann gab seiner Frau gestern das Buch." erlaubt die freie Wortstellung aller Konstituenten bis auf das Verb "gab", das immer an der zweiten Stelle eines Aussagesatzes steht. Auch diskontinuierliche Konstituenten sind häufig im Deutschen. In "Dieser Mann hat einen Hund gekauft, der beißt." gehört der Relativsatz zum Nomen "Hund", beide werden aber durch das Partizip 2 getrennt.

Die Beschreibung dieser Phänomene mit Hilfe von Grammatiken und Parsern stand 2005 im Vordergrund der Forschungsarbeiten. Dabei wird an zwei sehr unterschiedlichen Parsertypen gearbeitet, die trotzdem beide die Beschreibung der freien Wortstellung und der diskontinuierlichen Konstituenten zum Ziel haben:

- Der bereits existierende Parser für Zeichenketten erzeugende Hypergraphgrammatiken wurde für freie Wortstellung erweitert. Die Modellierung diskontinuierliche Konstituenten war mit diesem Parser bereits möglich. Für die freie Wortstellung wurden in Anlehnung an ID/LP Constraints eingeführt. Diese Constraints arbeiten jedoch nicht auf den Symbolen der Grammatik, die in der Hypergraphgrammatik durch Hyperkanten modelliert werden, sondern auf den Knoten zwischen den

Hyperkanten. Die Constraints erlauben es, eine (oder keine) Reihenfolge festzulegen, in denen die einzelnen Knoten besucht werden. Der existierende Earley-basierte Parser wurde so erweitert, dass er auch die Constraints verarbeiten kann.

- Für den Dependenzparser war die Situation umgekehrt. Hier war die Verarbeitung der freien Wortstellung kein Problem. Die Behandlung der diskontinuierlichen Konstituenten wurde 2005 basierend auf der Dissertation von Ricarda Dormeyer implementiert. Als Beispielsprache diente neben dem Deutschen auch Ungarisch, in dem beide Phänomene häufig vorkommen.

3.7 Graphen und Graphtransformationen

Projektleitung:

Prof. em. Dr. Hans Jürgen Schneider

Laufzeit: 1.1.1995 - 1.10.2006

Kontakt:

Prof. em. Dr. Hans Jürgen Schneider

Tel.: +49.9131.85.27620, Fax: +49.9131.85.28809,

E-Mail: schneider@informatik.uni-erlangen.de

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, 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.

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 markierten 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.

Am Lehrstuhl für Programmiersysteme wurden 2005 neue Ergebnisse auf folgenden Gebieten gefunden:

- Im Doppel-Pushout-Ansatz wird die linke Seite einer Produktion üblicherweise als injektiv vorausgesetzt, da der nichtinjektive Fall zu mehrdeutigen Ergebnissen führt. Die Behandlung von Markierungsänderungen mit Hilfe eines strukturierten Alphabetes führt aber auch im injektiven Fall zu Mehrdeutigkeiten. Eine bekannte Lösung für dieses Problem ist, nur die Transformation des zugrundeliegenden Graphen kategoriell, die Markierungsänderung aber separat zu behandeln. Im Gegensatz zu diesem Ansatz konnten wir alle Lösungen kategoriell charakterisieren, wobei auf ein Ergebnis von Kreowski und Ehrig aus dem Jahre 1976 zurückgegriffen wurde. Es konnte gezeigt werden, dass damit wichtige Anwendungsgebiete (Petri-Netze, Termgraphen, Datenbanken) abgedeckt werden. Dieses Ergebnis wurde in der "Ehrig-Festschrift" veröffentlicht.

- Wenn zwei oder mehr Produktionen auf einen Graphen angewandt werden können, kann das Ergebnis von der Reihenfolge der Anwendung abhängen, oder die zweite Produktion ist nicht mehr anwendbar, nachdem die erste angewandt wurde. Die meisten Autoren diskutieren das Kriterium zur Prüfung der Unabhängigkeit unter der strengen Voraussetzung, dass beide Seiten der Produktionen injektiv sind, obwohl der ursprüngliche Beweis nichtinjektive rechte Seiten zuließ. Wie einige Autoren explizit anmerken, genügt die Kategorie der strukturiert markierten Graphen der Unabhängigkeitsbedingung nicht. Wir konnten zeigen, dass eine unwesentliche Einschränkung der Struktur des Alphabets ausreicht, um die Gültigkeit des Unabhängigkeitstheorems für wichtige Anwendungen, wie Termgraphen und Datenbankanwendungen, zu garantieren.
- Mit Hilfe von Graphtransformationen werden nicht nur allgemeine Graphen generiert und analysiert, auch auf Zeichenketten basierende Sprachen und ihre Probleme können bearbeitet werden. Ein typisches Beispiel ist die kontextsensitive Sprache $a^n b^n c^n$. Sie kann mit kontextfreien Ersetzungsgrammatiken für Hyperkanten modelliert werden. Um mit diesen Hypergraphgrammatiken arbeiten zu können, wurde ein Parser entwickelt, der dem allgemeinen Earley-Parser ähnlich ist. In natürlichen Sprachen, dem Hauptanwendungsgebiet des Parsers, ist die freie Stellung der Konstituenten ein wichtiges Merkmal. Deswegen wurde der Parser ähnlich dem Vorgehen in ID/LP erweitert.

3.8 Zeitplanungs-Algorithmen

Projektleitung:

PD Dr.-Ing. habil. Dipl.-Inf. Peter Wilke

Stichwörter: Time-Tabling-Probleme

Laufzeit: 1.1.2004 - 31.12.2008

Kontakt:

PD Dr.-Ing. habil. Dipl.-Inf. Peter Wilke
Tel.: +49.9131.85.27624, Fax: +49.9131.85.8809,
E-Mail: wilke@informatik.uni-erlangen.de

Zeitpläne müssen in vielen unterschiedlichen Bereichen erstellt werden, z.B. in der Schulstundenplanung oder der Personaleinsatzplanung. Da es sehr mühsam ist, komplexe Zeitpläne wie Schulstundenpläne per Hand zu erstellen, werden die meisten Zeitpläne computerunterstützt generiert. Dazu wurde am Lehrstuhl in den vergangenen Jahren eine Software entwickelt, die es ermöglicht, die Planung unter zu Hilfenahme verschiedener Optimierungsalgorithmen durchzuführen. Diese Version der Zeitplanungssoftware wurde aus einer auf genetischen Algorithmen basierenden Version weiterentwickelt, wobei sich zeigte, dass einige Erweiterungen wegen der notwendigen Kompatibilität zur Grundversion nicht optimal implementieren ließen.

Im Jahr 2005 wurde daher ein komplett neuer Entwurf in Java implementiert. Die neue Version der Software hat die Kernkomponenten: Bedienoberfläche, Steuerung, Basis-Algorithmen, Editor für Beschränkungen, Datenbank, erweiterte Algorithmen sowie Ein- und Ausgabe.

Die Basisalgorithmen umfassen genetische Algorithmen, Branch-and-Bound, Simulated Annealing und Tabu Search. Die Datenbank basiert auf SQL-Datenbanken mit Hibernate als Umset-

zungsschicht. Die GUI basiert auf der Swing-Bibliothek, der Editor dient zur Eingabe der Ressourcen, Ereignisse und Beschränkungen. Die Steuerung verwendet einen Nachrichtenmechanismus, um die Berechnung auf verteilten Rechnern zu ermöglichen. Zur Qualitätssicherung wurden Testfälle mit JUnit implementiert.

Zum Ende des Jahres 2005 befinden sich die Softwarekomponenten Steuerung und Basis-Algorithmen in der Erprobungsphase. Datenbank, Bedienoberfläche und Editor stehen kurz vor der Fertigstellung. Die erste zur Veröffentlichung geeignete Version der Software wird auf der CeBIT 2006 vorgestellt

3.9 International Collegiate Programming Contest an der FAU

Projektleitung:

Prof. Dr. Michael Philippsen

Beteiligte: Dipl.-Inf. Alexander Dreweke, B. Sc. mult., Tilmann Spiegelhauer, Christian Riess, Dr.-Ing. Ingrid Fischer, Dipl. -Inf. Thorsten Meindl

Beginn: 01.11.2002

Kontakt:

Dipl.-Inf. Alexander Dreweke, B. Sc. mult.
Tel.: +49.9131.85.27923, Fax: +49.9131.85.28809,
E-Mail: dreweke@informatik.uni-erlangen.de

Die Association for Computing Machinery (ACM) richtet seit vielen Jahren den **International Collegiate Programming Contest (ICPC)** aus. Dabei sollen Teams aus je drei Studenten in fünf Stunden neun Programmieraufgaben lösen. Als Erschwernis kommt hinzu, dass nur ein Computer pro Gruppe zur Verfügung steht. Die Aufgaben erfordern solide Kenntnisse von Algorithmen aus allen Gebieten der Informatik, wie z.B. Graphen, Kombinatorik, Zeichenketten und Geometrie.

Der ICPC wird jedes Jahr in drei Stufen ausgetragen. Zuerst werden innerhalb der Universitäten in lokalen Ausscheidungen die maximal drei Teams bestimmt, die dann zu den regionalen Wettbewerben entsandt werden. Erlangen liegt im Einzugsbereich des **Southwestern European Regional Contest (SWERC)**, an dem u.a. auch Teams aus der Schweiz, Frankreich, Spanien und Portugal teilnehmen. Die Sieger aller regionalen Wettbewerbe der Welt (und einige Zweitplatzierte) erreichen die World Finals, die im Frühjahr des darauffolgenden Jahres stattfinden.

Im Jahr 2005 fanden erstmals zwei lokale Wettbewerbe an der FAU statt. Im Wintersemester wurde ein Mannschaftswettbewerb ausgetragen mit dem Ziel, neue Studierende für die Wettbewerbe zu begeistern. Jedes Team bestand aus maximal drei Studenten. Dreizehn Teams mit insgesamt 34 Teilnehmern nahmen am Winter-Wettbewerb teil, so viele wie nie zuvor.

Im Sommersemester fand zum wiederholten Mal das Hauptseminar "Hallo Welt - Programmieren für Fortgeschrittene" statt, um Studierende in Algorithmen und Wettbewerbsaufgaben zu schulen.

Der Wettbewerb im Sommersemester 2005 diente der Auswahl der Vertreter der FAU am SWERC 2005. Studierende der Fächer Mathematik, Informatik, Computational Engineering und Informations- und Kommunikationstechnologie nahmen teil. Insgesamt meldeten sich 37 Teilnehmer an. Die besten neun bildeten drei Teams mit jeweils drei Teilnehmern. Der Zehnt-

platzierte wurde als Ersatz ausgewählt. Aus der Gruppe der Wettbewerbsorganisatoren fuhr ebenfalls ein Student als Ersatzmann zum Wettbewerb nach Paris, ein zweiter Student organisierte als Trainer die Wettbewerbsvorbereitung.

Zwei Erlanger Teams erreichten die Top Ten auf Platz sechs und sieben. Das dritte Team beendete den Wettbewerb auf Platz 22 von 55. Dies ist das beste Gesamtergebnis, das die FAU bisher erzielt hat.

4 Publikationen

1. Monostori, Adam; Frühauf, Hans Holm; Kókai, Gabriella: Development of a model-library for quick estimation of resources of Field-Programmable Gate Arrays and Application Specific Integrated Circuits for automated Intellectual Property core generation with redFIR 2. In: Pécsi Tudományegyetem - Közgazdaságtudományi Kar (Hrsg.): 17th EURO Mini Conference: Continuous Optimization in Industry (17th EURO Mini Conference: Continuous Optimization in Industry Pécs, Hungary June 29 - July 1, 2005).
2. Monostori, Adam; Frühauf, Hans Holm; Kókai, Gabriella: Quick Estimation of Resources of FPGAs and ASICs Using Neural Networks. In: Bauer, Mathias; Kröner, Alexander; Brandherm, Boris (Hrsg.): LWA 2005 - Beiträge zur GI-Workshopwoche Lernen, Wissensentdeckung, Adaptivität (Workshop der GI-Fachgruppe "Maschinelles Lernen, Wissensentdeckung, Data Mining" (FGML) Saarbrücken, Germany 2005-10-10 - 2005-10-12). 2005, S. 210-215.
3. von Mammen, Sebastian; Jacob, Christian; Kókai, Gabriella: Evolving Swarms that Build 3D Structures. In: David Corne (Hrsg.): IEEE Congress on Evolutionary Computation (IEEE Congress on Evolutionary Computation Edinburgh, UK September 2nd--5th 2005). 2005, S. 1434- 1441.
4. Bauer, Elena; Kókai, Gabriella: Learning from Noise Data with the Help of Logic Programming Systems. In: Prof. Dr. Freund, Rudolf (Veranst.): WSEAS Transactions on Information Science and Application Artificial Intelligence, Knowledge Engineering, Data Bases (AIKED 2004) (Int.Conf. on Artificial Intelligence, Knowledge Engineering, Data Bases (AIKED 2004) Salzburg, Austria February 13-15 2005). 2005, S. 104-110. (WSEAS Transactions on Information Science and Application Bd. 2, Nr. 2).
5. Berthold, Michael R.; Glen, Robert; Diederichs, Kay; Kohlbacher, Oliver; Fischer, Ingrid (Hrsg.): Computational Life Sciences. (First International Symposium on Computational Life Sciences, CompLife 2005 Konstanz 25.9.2005-27.9.2005) Berlin : Springer, 2005 (Lecture Notes in Bioinformatics Bd. 3695). - 269 Seiten.
6. Borgelt, Christian; Meinl, Thorsten; Berthold, Michael: MoSS: A Program for Molecular Substructure Mining. In: Goethals, Bart; Nijssen, Siegfried; Zaki, Mohammed J. (Hrsg.): Proceedings of OSDM 2005 (Open Source Data Mining Workshop on Frequent Pattern Mining Implementations Chicago, IL, USA 2005-08-21). 2005, S. 6-15.
7. Böhm, Niko; Kókai, Gabriella; Mandl, Stefan: An Evolutionary Approach to Tetris. In: 6th Metaheuristics International Conference (6th Metaheuristics International Conference Wien August 22-26, 2005). 2005, S. CD-ROM.
8. Dormeyer, Ricarda; Fischer, Ingrid; Weber Russell, Sylvia: A Lexicon for Metaphors and Idioms. In: Schnorbusch, Daniel; Langer, Stefan (Hrsg.): Semantik im Lexikon. Tübingen : Gunter Narr Verlag, 2005, (Tübinger Beiträge zur Linguistik Bd. 479), S. 205-224. - ISBN 3-8233-6099-X.

9. Dreweke, Alexander: Implementation and Optimization of the Lattice Boltzmann Method for the Jackal DSM System. Erlangen-Nürnberg, Friedrich-Alexander-Universität, Bac-Arb., 2005. - 69 Seiten.
10. Fischer, Ingrid: Graph Transformations and Neural Networks. In: Wang, John (Hrsg.): Encyclopedia of Data Warehousing and Mining. Hershey, PA, USA : Idea Group, 2005, S. 534-539.
11. Hardung, Bernd; Kollert, Thomas; Kókai, Gabriella; Krüger, Andreas: Optimierung von Steuergerätevarianten unter Berücksichtigung der Kundenbestellungen. In: VDI Verein Deutscher Ingenieure (Veranst.): VDI Berichte Nr. 1907 (12th International Conference and Exhibition Electronic Systems for Vehicles Baden-Baden, Germany 6-7 October 2005). 2005, S. 619-630.
12. Hardung, Bernd; Kollert, Thomas: Optimisation of the Variant Combination of Control Units Considering the Order History. In: -- (Hrsg.): -- (Veranst.): Proceedings of the International Scientific Annual Conference Operations Research 2005 (Operations Research 2005 Bremen 7-9 September 2005).
13. Hörmann, K.; Dittmann, L; Hindel, Bernd; Müller, M: SPICE in der Praxis - Interpretationshilfe für Anwender und Assessoren. Heidelberg : dpunkt.verlag, 2005. - 280 Seiten. ISBN 3-89864-341-7.
14. Kókai, Gabriella; Frühauf, Hans Holm; Feng, Xu: Development of a Hardware Based Genetic Optimizer to Adjust Smart Antenna Receiver. In: International Journal of Embedded Systems Special Issue on Hardware-Software Codesign for Systems-on-Chip 1 (2005), Nr. 1-2, S. 50-65.
15. Meinl, Thorsten; Fischer, Ingrid; Philippsen, Michael: Parallel Mining for Frequent Fragments on a Shared-Memory Multiprocessor -Results and Java-Obstacles-. In: Bauer, Mathias; Kröner, Alexander; Brandherm, Boris (Hrsg.): LWA 2005 - Beiträge zur GI-Workshopwoche Lernen, Wissensentdeckung, Adaptivität (Workshop der GI-Fachgruppe "Maschinelles Lernen, Wissensentdeckung, Data Mining" (FGML) Saarbrücken, Germany 2005-10-10 - 2005-10-12). 2005, S. 196-201.
16. Meinl, Thorsten; Fischer, Ingrid: Subgraph Mining. In: Wang, John (Hrsg.): Encyclopedia of Data Warehousing and Mining. Hershey, PA, USA : Idea Group, 2005, S. 1059-1063.
17. Nijssen, Siegfried; Meinl, Thorsten; Karypis, George (Hrsg.): Proceedings of MGTS 2005. (3rd International Workshop on Mining Graphs, Trees and Sequences Porto, Portugal 07.10.2005) 2005. - 123 Seiten.
18. Schneider, Hans Jürgen: Changing Labels in the Double-Pushout Approach Can Be Treated Categorically. In: Kreowski, Hans Jörg (Hrsg.): Formal Methods in Software and Systems Modeling (Festschrift Ehrig). Berlin-Heidelberg : Springer, 2005, S. 134-149. - ISBN 3-540-24936-2.
19. Veldema, Ronald; Philippsen, Michael: Near Overhead-free Heterogeneous Thread-migration. In: IEEE (Hrsg.): Proceedings of the 2005 IEEE International Conference on Cluster Computing (2005 IEEE International Conference on Cluster Computing Boston, Massachusetts, USA 26.09.2005-30.09.2005). 2005, S. 145-154.
20. Wörlein, Marc; Meinl, Thorsten; Fischer, Ingrid; Philippsen, Michael: A quantitative comparison of the subgraph miners MoFa, gSpan, FFSM, and Gaston. In: Jorge, Alipio; Torgo, Luis; Brazdil, Pavel; Camacho, Rui; Gama, Joao (Hrsg.): Knowledge Discovery

in Database: PKDD 2005 (9th European Conference on Principles and Practices of Knowledge Discovery in Databases Porto, Portugal 2005-10-03 - 2005-10-07). Berlin : Springer, 2005, S. 392-403. (Lecture Notes in Computer Science Bd. 3721) - ISBN 3-540-29244-6.

5 Vorträge

1. Fischer, Ingrid: Beschreiben und Parsen von Graphen. Vortrag: Kolloquium Fachbereich Informatik und Informationswissenschaft, Universität Konstanz, Konstanz, 08.06.2005.

6 Studien- und Diplomarbeiten; Bachelor und Master Thesis

6.1 Studienarbeiten

1. Bieber, Benjamin: Erweiterung der Kommunikationsbibliothek Lizard.
2. von Mammen, Sebastian: Evolvierende Schwärme zur Bildung von 3D-Strukturen (Evolving Swarms that Build 3D Structures).
3. Handl, Johannes: Graphischer Editor für JavaKara.
4. Förster, Manuel: Interaktiver Assistent zur Einstellung der Multiplikatoren von multidimensionalen Zielfunktionen.
5. Bezold, Matthias: JaMP - Implementierung eines OpenMP-Dialektes im DSM-System Jackal.
6. Bauer, Elena: Vergleich des GeLog-Systems mit anderen GLP-Systeme auf der Basis einer statistischen Auswertung.
7. Klöden, Monic: Vergleich von Algorithmen zur Fragmentsuche in Moleküldatenbanken - Teil 2.
8. Wörlein, Marc: Vergleich von Algorithmen zur Fragmentsuche in Moleküldatenbanken.
9. Riedl, Martin: Wortstellungsrestriktionen für Zeichenketten generierenden Hypergraphgrammatiken.
10. Urzova, Olga: Zweidimensionale Visualisierung von Molekülgraphen in Java.

6.2 Diplomarbeiten

1. Meyer, Tobias: Ein Framework für Prozedurale Abstraktion.
2. Neumann, Christoph: Conceptional design of an open framework for optimizing the distribution of hardware and software components in control networks for vehicles.
3. Stoll, Tobias: Extending the use of Programmer supplied Assertions.
4. Korsch, Sven-Dimo: Re-Design eines Frameworks für Zeitplanungsprobleme.
5. Götz, Georg: Re-Design eines Frameworks für Zeitplanungsprobleme.
6. Heller, Bernd: Visualisierung auf einem Grid-System mittels verteiltem Java AWT.
7. Tröger, Frank: Design und Implementierung eines Kommunikationspaketes für Jackal

6.3 Bachelor Thesis

1. Dreweke, Alexander: Implementierung und Optimierung des Lattice-Boltzmann-Verfahrens für das DSM-System Jackal

6.4 Master Thesis

1. Monostori, Adam: Entwicklung eines Modells zur schnellen Abschätzung von Ressourcen von FPGAs und ASICs für die automatisierte IP-Core-Generierung mit redFIR 2.

Professur für Didaktik der Informatik

Anschrift: Martensstr. 3, 91058 Erlangen

Tel.: +49.9131.85.27621

Fax.: +49.9131.85.28809

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

1 Einführung

Die Fachgruppe für "Didaktik der Informatik" wurde im April 2005 mit der Besetzung der Professur durch Prof. Dr. Torsten Brinda neu gegründet und ist organisatorisch dem Lehrstuhl Informatik 2 zugeordnet. Bereits in den Jahren zuvor wurden mehrere zweijährige Nachqualifikationskurse im Rahmen des "Sofortprogramms Informatik am Gymnasium - Nachqualifikation von Lehrkräften (SIGNAL)" unter der Leitung von abgeordneten Informatiklehrern erfolgreich durchgeführt. Der letzte dieser Kurse endet im Studienjahr 2005/06.

- **Arbeitsschwerpunkte der Gruppe** bilden:
- die informatikdidaktische Ausbildung von Informatik-Lehramtsstudierenden sowie
- die informatikdidaktische Forschung.

Bedingt durch die Neugründung und die Besetzung der ersten Mitarbeiterstellen mit abgeordneten Informatiklehrkräften im Frühjahr 2006 befindet sich das Lehr- und Forschungsfeld an der Universität Erlangen-Nürnberg noch im Aufbau.

Lehre: Didaktik der Informatik I, Didaktik der Informatik II, Theoretische Informatik für das Lehramt, E-Learning (in Vorbereitung), Informationssysteme aus fachdidaktischer Sicht, informatikdidaktische Schulpraktika und zugehörige Begleitveranstaltungen

2 Forschungsschwerpunkte

Das übergeordnete Ziel der Forschung der Fachgruppe "Didaktik der Informatik" ist die Weiterentwicklung informatischer Bildung, insbesondere des Informatikunterrichts an Schulen. Aus informatischer Sicht sind hierzu einerseits Konzeptionen für softwarebasierte Lernhilfen für vielfältige Ausbildungsszenarios und andererseits für die Strukturierung von Lehr-Lern-Szenarios der Informatik von Belang. Beides erfordert Wissen über den informatischen Erkenntnisprozess bei Lernenden. Diesen drei Bereichen sind die Forschungsschwerpunkte der Arbeitsgruppe zugeordnet:

I.) Informatischer Erkenntnisprozess

- Entwicklung und theoretische Fundierung von Bildungsstandards für den Informatikunterricht (Aufgaben, Kompetenzmodelle, Testmethoden)
- Analyse und Evaluation von Informatikunterricht

II.) Lernhilfen der Informatik

- Konzeption, Implementierung und Evaluation interaktiver, multimedialer Lehr-Lern-Systeme
- Konzeption, Implementierung und Evaluation lernförderlicher Infrastrukturen

III.) Lehr-Lern-Szenarios der Informatik

- Entwicklung und Evaluation von Blended Learning Konzepten für den Informatikunterricht
- Entwicklung und Evaluation von Konzepten für die Informatiklehrausbildung

Kooperationsbeziehungen

- FU Berlin, Didaktik der Informatik
- Universität Kassel, Software Engineering
- TU München, Didaktik der Informatik
- Uni Münster, Didaktik der Informatik
- Universität Paderborn, Didaktik der Informatik
- Universität Potsdam, Didaktik der Informatik
- Universität Siegen, Didaktik der Informatik und E-Learning

3 Forschungsprojekte

3.1 Bildungsstandards der Informatik

Projektleitung:

Prof. Dr. Torsten Brinda

Stichwörter: Bildungsstandards, Kompetenzmodelle, Aufgabenklassen

Beginn: 1.7.2004

Kontakt:

Prof. Dr. Torsten Brinda

Tel.: +49.9131.85.27922, Fax: +49.9131.85.28809,

E-Mail: brinda@informatik.uni-erlangen.de

Auf der Basis eigener Vorarbeiten zu informatikdidaktischen Systemen werden, motiviert durch die Schlussfolgerungen aus den PISA-Studien zur Verstärkung der Output-Steuerung von Lehr-Lern-Prozessen, (Mindest-)Bildungsstandards für verschiedene Themenschwerpunkte des Informatikunterrichts erforscht. Zentrale Teilziele hierbei sind:

- die Analyse des Zusammenhangs zwischen so genannten Aufgabenklassen und Stufen noch zu entwickelnder Kompetenzmodelle,
- die Identifikation weiterer Aufgabenklassen und Kompetenzstufen durch Literaturstudien sowie
- die Entwicklung und Erprobung von Testitems im PISA-Sinne und geeigneten Software-Werkzeugen zur Analyse der Kompetenzprofile von Lernern und zur iterativen Präzisierung des Modells.

Forschungsmethodisch werden in diesem Projekt die Entwicklung von informatikdidaktischen Konzepten, theoretische Literaturanalysen und eine Kombination aus qualitativen und quantitativen empirischen Methoden miteinander verknüpft.

Publikationen

1. Brinda, Torsten: Integration of new exercise classes into the Informatics education in the field of object-oriented modelling. In: Education and Information Technologies 9 (2004), Nr. 2, S. 117-130.
2. Brinda, Torsten: Preparing educational standards in the field of object-oriented modelling. In: Magenheimer, Johannes; Schubert, Sigrid (Hrsg.): Informatics and student assessment. Concepts of empirical research and standardization of measurement in the area of didactics of informatics (Informatics and student assessment. Concepts of empirical research and standardization of measurement in the area of didactics of informatics Dagstuhl 19.-24.09.2004). 1. Aufl. Bonn : Köllen, 2004, S. 11-22. (Lecture Notes in Informatics / S: Seminars Bd. LNI / S, Nr. 1) - ISBN 3-88579-435-7.
3. Schulte, Carsten; Brinda, Torsten: Beiträge der Objektorientierung zu einem Kompetenzmodell des informatischen Modellierens. In: Friedrich, Steffen (Hrsg.): Unterrichtskonzepte für informatische Bildung (Unterrichtskonzepte für informatische Bildung Dresden 27.-30.09.2005). Bonn : Köllen, 2005, S. 137-148. (Lecture Notes in Informatics Bd. LNI, Nr. P-60).
4. Brinda, Torsten: Teaching object-oriented modelling in secondary education. In: IFIP - International Federation for Information Processing (Veranst.): 8th IFIP World Conference on Computers in Education - Conference CD (WCCE - World Conference on Computers in Education University of Stellenbosch, Cape Town, South Africa 04-07.07.2005). 2005.

3.2 Didaktische Systeme der Informatik

Projektleitung:

Prof. Dr. Torsten Brinda

Stichwörter: Didaktisches System, Aufgabenklassen, Explorationsmodule, Informatikunterricht, Objektorientiertes Modellieren

Beginn: 1.4.2000

Kontakt:

Prof. Dr. Torsten Brinda

Tel.: +49.9131.85.27922, Fax: +49.9131.85.28809,

E-Mail: brinda@informatik.uni-erlangen.de

In den Jahren 2000 bis 2004 wurde ein Konzept zur didaktischen Aufbereitung fachlicher Inhalte des objektorientierten Modellierens für Informatikunterricht der Sekundarstufe II als Didaktisches System für objektorientiertes Modellieren im Informatikunterricht der Sekundarstufe II gestaltet und evaluiert. Dieses Konzept wird nun verfeinert und seine Übertragbarkeit auf andere Teilgebiete der Informatik untersucht.

Publikationen

1. Brinda, Torsten: Didaktische Systeme für objektorientiertes Modellieren (OOM) im Informatikunterricht. In: Gesellschaft für Informatik e.V. (Hrsg.): Informatiktage 2000 (Informatiktage 2000 Bad Schussenried). Leinfelden-Echterdingen : Konradin, 2000, S. 282-285.
2. Brinda, Torsten: Objektorientiertes Modellieren - Sammlung und Strukturierung von Übungsaufgaben im Informatikunterricht. In: LOG IN 20 (2000), Nr. 5, S. 39-49.
3. Brinda, Torsten: Didactic system for object-oriented modelling. Vortrag: eingeladene, vom dänischen Bildungsministerium geförderte, Fortbildung für dänische Informatiklehrende der Sek. II, Bjerringbro / Denmark, 08.11.2001.
4. Brinda, Torsten: Einfluss fachwissenschaftlicher Erkenntnisse zum objektorientierten Modellieren auf die Gestaltung von Konzepten in der Didaktik der Informatik. In: Keil-Slawik, Reinhard; Magenheimer, Johannes (Hrsg.): Informatikunterricht und Medienbildung (GI Informatik und Schule - INFOS 2001 Paderborn 17.-20.09.2001). Bonn : Köllen, 2001, S. 75-86. (Lecture Notes in Informatics Bd. LNI, Nr. 8) ISBN 3-88579-334-2.
5. Brinda, Torsten; Ortman, Tobias: Fallstudien zur unterrichtlichen Einbettung spezieller Aufgabenklassen. In: Schubert, Sigrid; Magenheimer, Johannes; Hubwieser, Peter; Brinda, Torsten (Hrsg.): Forschungsbeiträge zur „Didaktik der Informatik“ - Theorie, Praxis, Evaluation. 1. GI-Workshop DDI'02 (Schwerpunkt: Modellierung in der informatischen Bildung) (DDI'02 Witten-Bommerholz 10.-11.10.2002). 1. Aufl. Bonn : Köllen, 2002, S. 13-22. (Lecture Notes in Informatics Bd. LNI, Nr. 22) - ISBN 3-88579-351-2.
6. Brinda, Torsten; Schubert, Sigrid: Learning aids and learners' activities in the field of object-oriented modelling. In: Passey, Don; Kendall, Mike (Hrsg.): Tele-Learning. The Challenge for the Third Millennium (IFIP World Computer Congress 2002 - WCC2002 Montreal, Canada 25.-30.08.2002). 1. Aufl. Boston : Kluwer, 2002, S. 37-44. - ISBN 1-4020-7219-8.
7. Brinda, Torsten; Schubert, Sigrid: Exploration of object-oriented models in informatics education. In: van Weert, Tom; Munro, Robert (Hrsg.): Informatics and The Digital Society: Social, Ethical and Cognitive Issues (SECIII 2002 Dortmund 22.-26.07.2002). 1. Aufl. Boston : Kluwer, 2003, S. 109-118. - ISBN 1-4020-7363-1.
8. Brinda, Torsten: Student experiments in object-oriented modelling. In: Cassel, Lillian N.; Reis, Ricardo Augusto da Luz (Hrsg.): Informatics curricula and teaching methods (ICTEM 2002 Florianopolis, Brazil 10.-12.07.2002). Boston : Kluwer, 2003, S. 13-20. - ISBN 1-4020-7266-X.
9. Brinda, Torsten: Didaktisches System für objektorientiertes Modellieren im Informatikunterricht der Sek. II. Siegen, Universität Siegen, Diss., 2004. - 267 Seiten.
10. Brinda, Torsten; Hoffmann, Andreas: Entwicklung von Software zur Exploration im Bildungskontext. In: Engels, Gregor; Seehusen, Silke (Hrsg.): DeLFI 2004: Die e-Learning Fachtagung Informatik (DeLFI 2004 Paderborn 05.-08.09.2004). 1. Aufl. Bonn : Köllen, 2004, S. 343-354. (Lecture Notes in Informatics Bd. LNI, Nr. 52) - ISBN 3-88579-381-4.
11. Brinda, Torsten: Integration of new exercise classes into the Informatics education in the field of object-oriented modelling. In: Education and Information Technologies 9 (2004), Nr. 2, S. 117-130.

12. Brinda, Torsten: Teaching object-oriented modelling in secondary education. In: IFIP - International Federation for Information Processing (Veranst.): 8th IFIP World Conference on Computers in Education - Conference CD (WCCE - World Conference on Computers in Education University of Stellenbosch, Cape Town, South Africa 04-07.07.2005). 2005.

3.3 SIGNAL - Sofortprogramm Informatik am Gymnasium - Nachqualifikation von Lehrkräften (ehemals NELLI)

Projektleitung:

Prof. Dr. Michael Philippsen

Beteiligte:

Prof. Dr. Torsten Brinda, Franz Forman, Dr. Klaus Kreisel, Dr. Hermann Puhlmann

Laufzeit: 1.9.2002 - 31.8.2006

Förderer:

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

Kontakt:

Prof. Dr. Michael Philippsen
Tel.: +49.9131.85.27625, Fax: +49.9131.85.28809,
E-Mail: philippsen@informatik.uni-erlangen.de

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. Konsequenterweise führte Bayern im Schuljahr 2004/05 das Pflichtfach Informatik - zunächst als Teil des Faches "Natur und Technik" - an allen Gymnasien ein. Jedes der rund 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 Lehrerbildung in Informatik) gestartet und ein elektronisches Angebot von speziell zugeschnittenen Studien- und Unterrichtsmaterialien entwickelt. Der Lehrstuhl 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). In den Jahren 2002 bis 2004 startete an der Universität Erlangen-Nürnberg jeweils ein zweijähriger Kurs mit rund 25 Teilnehmern.

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. Im April 2005 wurde die Fachgruppe für "Didaktik der Informatik" mit der Besetzung der Professur durch Prof. Dr. Torsten Brinda neu gegründet und ist organisatorisch dem Lehrstuhl Informatik 2 zugeordnet. Bereits in den Jahren zuvor wurden mehrere zwei-jährige Nachqualifikationskurse im Rahmen des "Sofortprogramms Informatik am Gymnasium - Nachqualifikation von Lehrkräften (SIGNAL)" unter der Leitung von abgeordneten Informatiklehrern erfolgreich durchgeführt. Der letzte dieser Kurse endet im Studienjahr 2005/06.