

# 1 Lehrstuhl für Informatik 2 (Programmiersysteme)

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

**Tel.:** +49 9131 85-27621

**Fax:** +49 9131 85-28809

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

## **Ordinarius:**

Prof. Dr. Michael Philippsen

## **Honorar- und Außerplanmäßige Professoren:**

Hon.-Prof. Dr.-Ing. Bernd Hindel

Hon.-Prof. Dr.-Ing. Detlef Kips

apl. Prof. Dr.-Ing. Gabriella Kókai

## **Emeritus:**

Prof. em. Dr. Hans Jürgen Schneider

## **Sekretariat:**

Agnes Brütting

Waltraud Rück (bis 30.06.2009)

Margit Zenk (ab 01.09.2009)

## **Wiss. Mitarbeiter:**

Dipl.-Inf. Alexander Dreweke, B. Sc. mult.

Dipl.-Ing. (FH) Thorsten Edelhäuser

Dipl.-Inf. Ralf Ellner

Dipl.-Inf. Philipp Janda

Dipl.-Inf. Stefan Kempf (ab 01.06.2009)

Dr.-Ing. Norbert Oster

Ronald Veldema, Ph.D.

Dipl.-Inf. Tobias Werth

PD Dr.-Ing. habil. Peter Wilke (bis 06.11.2009)

Dipl.-Inf. Marc Wörlein

## **Gäste/Stipendiaten:**

Dipl.-Inf. (FH) Josef Adersberger

Dipl.-Inf. Johannes Drexler

Dr.-Ing. Martin Jung

Dr.-Ing. Stephan Otto

Dr.-Ing. Szilvia Zvada

## **Externes Lehrpersonal:**

Dipl.-Inf. Samir Al-Hilank

Dr.-Ing. Klaudia Dussa-Zieger

Dr. Karsten Schmidt

Der 1972 gegründete Lehrstuhl Informatik 2 (Programmiersysteme) wird seit April 2002 von Prof. Dr. Michael Philippsen (als Nachfolger von Prof. Dr. H.-J. Schneider) geleitet. Eng mit dem Lehrstuhl assoziiert sind die beiden Professuren für Didaktik der Informatik und Open Source Software, deren Forschungsarbeiten separat dargestellt sind.

## 1.1 Forschungsschwerpunkte

Im Mittelpunkt der **Programmiersystemforschung** 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 schon für Einprozessorsysteme und Arbeitsplatzrechner. Langfristiges Ziel ist es, den Anwendungen die verfügbare Rechen- und Kommunikationsleistung möglichst ungebremst zur Verfügung zu stellen bzw. aus begrenzten Systemen ein Maximum herauszuholen. Ein besonderer Arbeitsschwerpunkt sind Programmiersysteme für Multicore-Rechner, da deren unausweichliche Verbreitung ebenso wie die preisgünstige Verfügbarkeit von sehr leistungsstarker paralleler Spezialhardware im Massenmarkt (z.B. Grafik-Karten oder FPA-Hardware) kaum abschätzbare Auswirkungen auf die Software-Landschaft haben werden. Forschungsergebnisse werden stets an eigenen Prototypen und Demonstratoren praktisch evaluiert.

### Wichtige Forschungsfelder

- **Vorhandenes Parallelisierungspotential ausschöpfen.** Da die Taktraten von Mehrkernrechnern kaum noch steigen, dafür aber deren Kernanzahl anwachsen wird, muss das Parallelisierungspotential existierender Software erkannt und ausgeschöpft werden, um an den Leistungssteigerungen der Hardware teilzuhaben. Außer vielleicht in Nischen führt kein Weg an einem Umstieg auf Parallelverarbeitung vorbei. Deshalb entwickelt der Lehrstuhl Werkzeuge, die dem Programmierer interaktiv beim Re-Engineering bestehender Anwendungen helfen, und erarbeitet Architekturmuster für neu zu entwickelnde Software-Projekte, die eine Skalierbarkeit für und damit eine Toleranz gegen wachsende Kernanzahlen aufweisen.
- **Hochleistungsanwendungen portabel machen.** Anwendungsprogrammierer erreichen nur dann bestmögliche Laufzeiten, wenn sie die Überwindung der Latenzzeiten und die explizite Kommunikation zwischen unterschiedlichen Komponenten der Systemarchitektur manuell angehen, ihren Code mit Hardware-nahen

”Tricks” spezifisch für eine Architektur optimieren und ihre Applikation per Hand in mehrere Teile zerlegen, von denen manche z.B. auf die Grafikkarte ausgelagert werden. Der Lehrstuhl erforscht, wie durch Anhebung des Abstraktionsniveaus der Programmierung die Produktivität gesteigert und die Portabilität verbessert werden kann, indem Code so übersetzt wird, dass verschiedene Teile auf heterogenen Systemkomponenten nebenläufig ausgeführt werden und dass die Datenkommunikation dazwischen transparent für den Entwickler erfolgt. Eine wichtige Frage dabei ist es, wie der Programmierer sein Wissen über bestehende Lokalitätsbeziehungen programmiersprachlich ausdrücken kann, damit es effizienzsteigernd nutzbar ist bzw. damit Lokalität schon im Design sichtbar ist. Auf dem Weg zu diesem Ziel werden im Rahmen von Re-Engineering-Projekten Details der Hardware-Architektur vor dem Anwendungsentwickler verborgen, z.B. hinter Bibliotheksschnittstellen oder in domänenspezifischen Programmierspracherweiterungen.

- **Parallelitätsgrad dynamisieren.** Hochleistungsanwendungen werden oft für eine bestimmte Prozessorzahl entwickelt. Da die benötigten Rechnerbündelknoten vom Batch-System statisch für eine feste Zeitspanne zugeteilt werden, entstehen zwangsläufig unwirtschaftliche Reservierungslöcher. Analoge Probleme treten bei vielfädigen Anwendungen auf Multicore-Rechnern auf. Der Lehrstuhl arbeitet daher an der dynamischen Anpassung des Parallelitätsgrads durch Code-Transformationen, die die Kosten der resultierenden Datenumverteilungen berücksichtigen, sowie durch Interaktion und Zusammenarbeit mit dem Betriebssystem. Die in Programmen vorgefundenen expliziten, kontrollflussorientierten Synchronisationsmaßnahmen behindern die erforderlichen Analysen, weshalb der Lehrstuhl neue und bessere Programmkonstrukte erforscht, die die bisherigen adäquat ersetzen können und die Synchronisationserfordernisse deklarativ und an den Daten orientiert ausdrücken.
- **Parallelität testbar machen.** Im Software-Engineering nimmt Testen von jeher eine wichtige Stellung ein. Stichworte wie Testüberdeckung, Testdatengenerierung, Zuverlässigkeitsbewertung etc. gehören zum Handwerk. Leider berücksichtigt die Forschung in diesem Bereich den durch Nebenläufigkeit entstehenden Indeterminismus nur unzureichend. Der Lehrstuhl arbeitet daher an Werkzeugen zur Testdatengenerierung, die bei den zugrundeliegenden Überdeckungskriterien auch Verschränkungen nebenläufiger Programmäste berücksichtigen. Dies schließt Arbeiten an Betriebssystemschnittstellen und am Ablaufplaner ebenfalls ein. Weil ferner durch die Nebenläufigkeit der Suchraum in erheblichem Maße wächst, müssen Infrastrukturen erarbeitet werden, die es ermöglichen, Massentests auf großen Rechnerbündeln auszuführen.

- **Software-Entwicklungsprozesse verbessern.** Die heute in der Industrie praktizierte Entwicklung von komplexer, geschäfts- oder sicherheitskritischer Software in global verteilten Teams verlangt nach der Einhaltung von wohldefinierten Software-Entwicklungsprozessen mit entsprechender Werkzeugunterstützung. Im Bereich der **praktischen Softwaretechnik** arbeitet der Lehrstuhl zusammen mit den **Honorar-Professoren Dr. Bernd Hindel und Dr. Detlef Kips**, die als Geschäftsführer zweier mittelständischer Software-Beratungsunternehmen über langjährige Praxiserfahrung in industriellen Software-Entwicklungsprojekten verfügen, daher an einer maschinen-ausführbaren Notation für die Modellierung von Software-Entwicklungsprozessen, wobei wegen der zum Einsatz kommenden vielfältigen Werkzeuge und Notationen, sowohl die (teil-) automatisierte Rückgewinnung von Nachverfolgbarkeitsinformation aus den Artefakten eines Entwicklungsprojekts, als auch die modellbasierte Entwicklung, Integration und Konfiguration von Softwarekomponenten betrachtet werden, wie sie insbesondere beim Entwurf eingebetteter Systeme für den Automobilbau üblich sind.

## 1.2 Forschungsprojekte

### 1.2.1 Übersetzerunterstützte Parallelisierung für Mehrkern-Architekturen

**Projektleitung:**

Prof. Dr. Michael Philippsen

**Beteiligte:**

Dipl.-Inf. Tobias Werth

Dipl.-Inf. Dominic Schell

**Beginn:** 1.3.2007

**Kontakt:**

Dipl.-Inf. Tobias Werth

Tel.: +49 9131 85-28865

Fax: +49 9131 85-28809

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

Die Entwicklung von schnelleren und immer effizienteren Rechnerarchitekturen ist in den letzten Jahren an verschiedene Grenzen gestoßen. Althergebrachte Techniken trugen nicht mehr oder nur noch wenig zur Beschleunigung der Hardware bei. Grundprobleme sind dabei das auseinanderdriftende Verhältnis der Latenzen von Speicher und CPU und die Abwärme bei steigenden Taktfrequenzen.

Als Lösung drängen sich homogene und heterogene Mehrkern-Architekturen auf, die dem Programmierer enorme Leistung zur Verfügung stellen. Durch verringerte Taktfrequenzen tritt ein Großteil der genannten Problematik nicht auf, die hohe Leistung

wird durch Vervielfältigung der Ressourcen erreicht. Somit sind zum Beispiel bei niedrigerem Energieverbrauch mehr Rechenoperationen pro Zeiteinheit möglich. Unter Umständen wird mittels Spezialisierung einzelner Komponenten die Rechenleistung weiter erhöht. Durch eine mehrschichtige Speicherhierarchie mit vielen Zwischenspeichern soll zum Beispiel das Problem der Latenz verkleinert werden.

Aus Mehrkern-Architekturen die volle Leistung abzurufen stellt sich als große Schwierigkeit für den Programmierer heraus. Die hohe Rechenkapazität kann er nur dann erreichen, wenn er Expertenwissen sowohl in der Domäne der Anwendung, als auch für die konkrete Architektur besitzt.

Gegenstand der Forschung sind dabei unter anderem die folgenden Fragestellungen: Welche Unterstützung kann der Übersetzer dem Programmierer beim Entwickeln von Anwendungen für verschiedenen Mehrkern-Architekturen bieten? Wie viel Kontextwissen ist notwendig, damit der Übersetzer sinnvolle Entscheidungen bei der Parallelisierung auf die einzelnen Kerne trifft? Welchen Anteil der zur Verfügung stehenden Rechenkapazität kann der Programmierer mit vertretbarem Aufwand erreichen, ohne Detailwissen über die Eigenheiten der einzelnen Architekturen besitzen zu müssen? Wie müssen geeignete Werkzeuge zum Auffinden von Fehlern und Flaschenhälsen in der Anwendung auf Mehrkern-Architekturen aussehen?

Ziel dieses Projektes ist es, diese Fragen anhand einer eingeschränkten Anwendungsdomäne zu beantworten und mögliche Lösungswege aufzuzeigen. Als Domäne wird das Lattice-Boltzmann-Verfahren herangezogen, das vor allem in der Strömungssimulation angewandt wird. Durch seine Gitterstruktur und eine überschaubare Anzahl an Datenabhängigkeiten zwischen den einzelnen Zellen lässt sich das Verfahren relativ einfach parallelisieren, so dass sich die Forschung auf die oben genannten Fragestellungen konzentrieren kann.

Die heterogene CellBE-Architektur bietet sich aufgrund ihrer enormen Leistung auf einem Chip als Zielarchitektur an. Sie besteht aus einem PowerPC-Kern (PPU) und acht sogenannten Synergistic Processing Units (SPUs), die Berechnungen parallel ausführen können. Auf den SPUs steht nur ein begrenzter Speicherbereich (256kb) zur Verfügung, der optimal genutzt werden muss, um die angebotene Leistung ausnutzen zu können.

Auf den speziellen Kernen der CellBE-Architektur steht nur wenig Speicher (256kB) zur Verfügung. Dieser Platz muss auf Code und Daten aufgeteilt werden. Die gegenwärtig verfolgte Idee beruht darauf, dass nicht der gesamte Programmcode einer Anwendung komplett im Arbeitsspeicher liegen muss, um das Programm auszuführen.

Im Rahmen einer im Jahr 2009 fertiggestellten Dissertation wurde eine automatische Speicherverwaltung für Programmcode entwickelt, die den Code in Fragmenten (von Basisblöcken bis hin zu Funktionen) in einen Code-Cache lädt. Wenn es zu einem Speicherengpass kommt, wird nicht mehr benötigter Code automatisch erkannt und durch

einen speziellen Speicherbereiniger aus dem Speicher entfernt. Dazu wurden mehrere Speicherbereiniger untersucht. Die damit verbundene Speichereinsparung reduziert den Platzbedarf, die Kosten und den Energiekonsum, z.B. in eingebetteten Systemen. In Benchmarks auf der CellBE-Architektur zeigt sich, dass der Programmlader gegenüber der nativen Ausführung eine deutliche Einsparung (bis zu 80%) an Instruktionen bei relativ geringen Laufzeit-Aufwand (plus 3%) aufweist. Im Rahmen der Dissertation wurden weitere Prototypen für andere Architekturen entwickelt, die ähnliche Resultate liefern.

Desweiteren wurde eine freie Bibliothek zur Simulation nach dem Lattice-Boltzmann-Verfahren (OpenLB) untersucht. Dabei wurden verschiedene Ansätze für die Parallelisierung des Verfahrens auf heterogene Architekturen prototypisch implementiert und evaluiert. Für den 2D- und 3D-Fall wurden Konzepte und Strukturen entwickelt, die z.B. das Programmieren mit dem Vektorbefehlssatz der CellBE-Architektur vereinfachen sollen. Sowohl im Hinblick auf Bedienbarkeit als auch die erzielte Leistung konnten bei der Portierung auf die CellBE-Architektur zufriedenstellende Ergebnisse erzielt werden.

Außerdem wurde der Programmlader für die CellBE-Architektur weiter verbessert, um auch die größte Beschränkung (maximal 256kB Code) umgehen zu können. Dazu wurden vorhandene Overlay-Techniken untersucht, um den dynamischen Ansatz des Programmladers mit den Vorteilen von statischen Overlays und eines Overlay-Managers zu verknüpfen. Eine prototypische Implementierung zeigt, dass dieses Vorgehen erfolgsversprechend ist. Im Rahmen weiterer Forschungen muss hier allerdings noch die Ausführungsgeschwindigkeit optimiert werden.

## **1.2.2 Graphbasierte Prozedurale Abstraktion**

### **Projektleitung:**

Prof. Dr. Michael Philippsen

### **Beteiligte:**

Dipl.-Inf. Alexander Dreweke, B. Sc. mult.

Dipl.-Inf. Marc Wörlein

Dipl.-Inf. Tobias Werth

**Beginn:** 1.4.2006

### **Kontakt:**

Prof. Dr. Michael Philippsen

Tel.: +49 9131 85-27625

Fax: +49 9131 85-28809

E-Mail: philippsen@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 maschinennah 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. 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. 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?

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 der neu erzeugten 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 ARM Assembler-Code, der auf eingebetteten Systemen weit verbreitet ist. In Kooperation mit dem Projekt ParSeMiS, das sich mit der Optimierung von Graph-Minern befasst, werden auch die für PA spezifischen Probleme beim Graph-Minen angegangen. Im Berichtszeitraum wurden die Analysen zur korrekten Rekonstruktion des Datenflussgraphen verfeinert. Eine möglichst genaue Rekonstruktion erhöht das Einsparungspotential im Vergleich zu den herkömmlichen sequentiellen Verfahren. Des Weiteren wurden verschiedene Auslagerungsmethoden optimiert. Diese dienen dazu, die von ParSeMiS als häufig eingestuft Code-Fragmente herauszuziehen und damit das Programm zu verkleinern. Die optimierten Auslagerungsmethoden zeichnen sich vor allem dadurch aus, dass sie möglichst kosteneffizient semantisch gleiche Fragmente vereinheitlichen.

### 1.2.3 ParSeMiS - die Parallele und Sequenzielle Mining Suite

**Projektleitung:**

Prof. Dr. Michael Philippsen

**Beteiligte:**

Dipl.-Inf. Marc Wörlein

Dipl.-Inf. Alexander Dreweke, B. Sc. mult.

Dipl.-Inf. Tobias Werth

**Beginn:** 1.5.2006

**Kontakt:**

Prof. Dr. Michael Philippsen

Tel.: +49 9131 85-27625

Fax: +49 9131 85-28809

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

Die Arbeitsgruppe **ParSeMiS (Parallele und Sequenzielle Graph Mining Suite)** beschäftigt sich mit der Suche nach häufigen interessanten Strukturen in Graphdatenbanken; ein Forschungsgebiet, das in den letzten Jahren sehr reges Interesse findet. Da viele Forschungs- und Wirtschaftsdaten in strukturierter Form erfasst werden können, bietet sich die Speicherung komplexer Zusammenhänge in Form von allgemeinen oder speziellen Graphen an.

Diese meist unüberschaubaren Datenmengen sind nur schwer mit Hand und Auge zu erfassen, so dass Algorithmen zur Entdeckung interessanter Korrelationen unabdingbar

sind. Da deren Entdeckung in Graphen im Allgemeinen aufwändig ist (NP-vollständig), ist die Suche nach parallelen und spezialisierten Algorithmen und Heuristiken notwendig, die den benötigten Rechenzeit- und Speicheranforderungen auch bei immer größer werdenden Datenmengen gewachsen sind.

Das Ziel dieses Projektes ist es, ein effizientes und flexibles Werkzeug zur Suche in beliebigen Graphdaten bereitzustellen, um sowohl die Einbindung in neue Anwendungsgebiete als auch die Entwicklung neuer Suchverfahren zu beschleunigen und zu vereinfachen.

Im Jahr 2009 wurden folgende Ziele angegangen:

- Beschleunigung der Suche mit einbettungs-basierter Häufigkeit: Indem auf die zum Pruning benötigten Eigenschaften des Maximalen-Cliquen-Test intensiver eingegangen wurde, konnte gerade in dem für Anwendungen interessanten Bereich von niedrigen Häufigkeiten eine hohe Beschleunigung erreicht werden. Dies wird durch eine frühzeitige Erkennung im NP-vollständigen Test ermöglicht, die feststellt, ob ein überhaupt noch Fragment noch häufig vorhanden sein kann.
- Verbesserte Verteilung für Mehrkern-Cluster-Architekturen: Hier wurde und wird untersucht, wie sich die gemeinsame Positionierung verschiedener Threads im selben realen Speicher zur Beschleunigung der parallelen Suche ausnutzen lässt.

#### 1.2.4 JavaParty

**Projektleitung:**

Prof. Dr. Michael Philippsen

**Beteiligte:**

Dipl.-Inf. Marc Wörlein

**Beginn:** 1.4.2007

**Kontakt:**

Prof. Dr. Michael Philippsen

Tel.: +49 9131 85-27625

Fax: +49 9131 85-28809

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

[JavaParty]<http://svn.ipd.uni-karlsruhe.de/trac/javaparty/wiki/JavaParty> erlaubt eine einfache Portierung von parallelen Java-Programmen mit mehreren Threads auf eine verteilte Umgebung wie z.B. einem Cluster. Das Standard-Java unterstützt parallele Programme durch Threads und Synchronisationsmechanismen. Während Mehrprozess-Java-Programme auf einen einzelnen Speicheraddressbereich beschränkt sind, dehnt JavaParty die Möglichkeiten von Java auf verteilte Systeme aus.

Die übliche Art, parallele Anwendungen auf eine verteilte Umgebung zu portieren, ist die Verwendung von Kommunikationsbibliotheken. Java's entfernter Methodenaufruf (RMI) macht die Verwendung expliziter Kommunikationsprotokolle unnötig, führt aber immer noch zu einer erhöhten Programmkomplexität. Gründe dafür sind die beschränkten Möglichkeiten des RMIs und die benötigte zusätzliche Funktionalität für die Erzeugung und den Zugriff auf entfernte Objekte.

Der Ansatz von JavaParty ist anders. JavaParty-Klassen können direkt als entfernt (remote) deklariert werden. Während normale Java Klassen auf eine einzelne Virtuelle Maschine beschränkt sind, sind entfernte Klassen und deren Instanzen in der gesamten verteilten JavaParty-Umgebung sichtbar und erreichbar. Soweit man nur entfernte Klassen betrachtet, kann die JavaParty-Umgebung als eine Virtuelle Maschine angesehen werden, die sich über verschiedene Computer verteilt. Der Zugriff und die Erzeugung von entfernten Klassen sind syntaktisch nicht von dem regulärer Java-Klassen zu unterscheiden.

Im Jahr 2009 wurde der JavaParty-Übersetzer um eine Semantik für innere Klassen erweitert.

### **1.2.5 Jackal - Cluster and Grid computing made easy**

**Projektleitung:**

Prof. Dr. Michael Philippsen

**Beteiligte:**

Ronald Veldema, Ph.D.

**Beginn:** 1.1.2006

Im Jackal Projekt wird ein Distributed Shared Memory Systems (DSM) für Java entwickelt. Das System erlaubt es, ein Programm mit mehreren Threads auf einem Rechnerbündel auszuführen. Gleichzeitig bleibt das Programm auf normalen JVMs (die nur einen Rechner unterstützen) lauffähig. Zur besseren Fehlertoleranz beinhaltet Jackal ein Checkpoint System, das in periodischen Abständen den Programmzustand auf die Festplatte speichern kann. Es ist dann möglich, diesen gespeicherten Zustand zu einem späteren Zeitpunkt zu laden und die Ausführung fortzusetzen. Außer normalen nebenläufigen Programmen lassen sich mit Jackal Anwendungen, die OpenMP Direktiven beinhalten, verwenden. Durch eine Kombination aus dem Checkpoint System und den OpenMP Direktiven kann man Anwendungen (ab dem gespeicherten Checkpoint) mit einer beliebigen Rechnerzahl wiederaufnehmen. Man ist dadurch nicht mehr an die Anzahl der Rechner gebunden, die beim Starten des Programms festgelegt wurde.

OGRE ist ein Unterprojekt, das das Jackal-DSM für den Einsatz im Grid-Computing erweitert. OGRE übernimmt die Erstellung von Arbeitsaufträgen, die Verwaltung von

Checkpoints und die Verteilung von Arbeitsaufträgen zwischen Rechnerbündeln.

### 1.2.6 Tapir

**Projektleitung:**

Prof. Dr. Michael Philippsen

**Beteiligte:**

Ronald Veldema, Ph.D.

Dr.-Ing. Michael Klemm

**Laufzeit:** 1.1.2006–31.12.2010

**Kontakt:**

Ronald Veldema, Ph.D.

Tel.: +49 9131 85-27622

Fax: +49 9131 85-28809

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

Tapir ist eine neue Programmiersprache zur Vereinfachung der Systemprogrammierung.

Systemprogrammierung bezeichnet unter anderem die Programmierung von Netzwerkprotokollen, Betriebssystemen, Middleware- und DSM-Systemen. Diese Komponenten stellen essentielle Teile eines Systems dar, da sie Dienste bereitstellen, auf denen andere Applikationen aufbauen. Das Betriebssystem stellt einer Anwendung z.B. eine Ausführungsumgebung bereit, die von der konkreten Hardware abstrahiert, so dass die Applikation eine robuste, Hardware-unabhängige Schnittstelle nutzen kann. Ein weiteres Beispiel für Systemprogrammierung ist ein DSM-System. Es simuliert in einem Rechnerbündel mit verteiltem Speicher einen gemeinsamen Adressraum und verbirgt die Kommunikation im Rechnerbündel vor der Anwendung.

Im Vergleich zu Anwendungssoftware stellt systemnahe Software völlig andere Anforderungen an die Programmierung. Die Leistung der einzelnen Systemkomponenten kann sich auf alle Anwendungen und somit das gesamte System auswirken. Deshalb muss der erzeugte Maschinen-Code besonders effizient ausführbar sein. Fehler auf der Systemebene beeinflussen nicht nur einzelne Anwendungen sondern können ebenfalls das gesamte System beeinträchtigen. Systemsoftware sollte aus diesem Grund möglichst (beweisbar) fehlerfrei sein. All diese Anforderungen haben direkte Auswirkungen auf die verwendbaren Programmiersprachen und den verwendeten Programmierstil:

- Hochsprachen wie C++, C# und Java verstecken Implementierungsdetails vor dem Programmierer. Der Programmierer benötigt z.B. kein Wissen darüber, wie

ein Methodenaufufr konkret durchgeführt wird. Dieses Wissen ist jedoch bei der Entwicklung von Systemsoftware erforderlich.

- Hochsprachen stellen Funktionen bereit, die für Systemsoftware in der Regel nicht benötigt werden oder sogar unerwünscht sind. Beispielsweise wird innerhalb eines Betriebssystems keine automatische Speicherbereinigung oder Ausnahmebehandlung verwendet.
- Systemprogramme erfordern kein so hohes Abstraktionsniveau, wie es meist von Hochsprachen gefordert wird. Ebenso verzichtet man bei der Erstellung von Systemsoftware zumeist auf die Benutzung externer Bibliotheken, da viele Bibliotheken auf die Systemsoftware als Basis angewiesen sind und somit nicht zur Verfügung stehen.

Tapir ist an existierende Hochsprachen wie C++ und Java angelehnt, verzichtet aber auf Eigenschaften und Funktionen die ohne Bedeutung für eine Sprache zur Systemprogrammierung sind. Beispielsweise besitzt Tapir keine Speicherbereinigung, Ausnahmebehandlung oder Typumwandlung. Klassen und Objekte können zwar definiert werden, eine Vererbungsbeziehung zwischen Klassen ist aber nicht erlaubt. Das mit Tapir spezifizierte Systemprogramm kann mit Model Checking-Techniken bereits während der Entwicklung auf Fehler überprüft werden. Ein Übersetzerprototyp und ein Werkzeug zum Model Checking sind bereits implementiert. Während sich Tapir noch in der Entwicklung befindet, wurde es bereits verwendet, um eine Spezifikation für das DSM-Protokoll von Jackal zu erarbeiten. Zur Erweiterung des Tapir-Sprachentwurfes wurden RDMA-basierte DSM-Protokolle evaluiert. Die semantische Analyse von Tapir-Programmen ist sehr speicherintensiv, da sie auf Model Checking beruht. Um die Analyse zu beschleunigen, war es erforderlich, eine eigene, auf sehr große Objektmengen spezialisierte, Virtuelle Maschine für Java zu entwickeln. Die neu entwickelte, LVM genannte virtuelle Maschine zeigt wesentlich bessere Laufzeiteigenschaften als übliche Java Virtual Machine Implementierungen, sobald der verfügbare Hauptspeicher nicht mehr ausreicht und das Auslagern auf den Hintergrundspeicher notwendig wird.

Im Berichtszeitraum 2009 wurde die Sprachspezifikation von Tapir verbessert. Die neue Sprachspezifikation erleichtert die automatische Verifizierung und erlaubt die Generierung von effizienterem Maschinencode. Um die Sprache leichter verifizieren zu können, ist es nun möglich, Programmteile als Implementierungsdetails zu markieren. Diese markierten Teile können danach problemlos bei der Verifizierung ignoriert werden. Die höhere Effizienz der Sprache wurde durch die Unterstützung von Nebenläufigkeit erreicht. Programme können damit (z.B. auf Grafikkarten) parallel ausgeführt werden, ohne dass es zu Verklemmungen oder ähnlichen Fehlern kommt, die üblicherweise bei paralleler Programmierung auftreten.

## 1.2.7 OpenMP/Java

### **Projektleitung:**

Prof. Dr. Michael Philippsen

### **Beteiligte:**

Ronald Veldema, Ph.D.

Dr.-Ing. Michael Klemm

Dipl.-Inf. Georg Dotzler

**Laufzeit:** 1.10.2009–1.10.2015

JaMP ist eine Implementierung des bekannten OpenMP Standards für Java. JaMP erlaubt es (unter anderem), Schleifen zu parallelisieren, ohne sich mit der low-level Thread API von Java befassen zu müssen. Eine parallele Schleife hätte in JaMP folgende Form:

```
class Test {  
...void foo() {  
.....//#omp parallel for  
.....for (int i=0;i<N;i++) {  
.....a[i]= b[i]+ c[i]  
.....}  
...}  
}
```

JaMP implementiert im Moment die Funktionalität von OpenMP 2.0 und Teile der Spezifikation 3.0 (z.B. die collapse clause).

Eine ältere Version von JaMP ist in Jackal integriert und erlaubt es, Java Programme auf Rechnerbündeln einzusetzen. Es besteht sogar die Möglichkeit, die Programme mit dem OGRE Framework innerhalb des Grids auf andere Rechnerbündel zu verschieben.

Die aktuelle JaMP-Version, die unabhängig von Jackal ist und reinen Java 1.5 Code erzeugt, der auf jeder JVM lauffähig ist, kann sogar CUDA-fähige Grafik-Hardware zur parallelen Abarbeitung von Schleifen verwenden. Falls sich für einen Schleifenrumpf eine Transformation nach CUDA verbietet, wird stattdessen nebenläufiger Code für gängige Multicore-Prozessoren erzeugt.

## 1.2.8 Modellgetriebene Komponentenkombination

### **Projektleitung:**

Prof. Dr. Michael Philippsen

**Beteiligte:**

Dipl.-Inf. Philipp Janda

**Laufzeit:** 15.6.2007–14.6.2010

**Förderer:**

AUDI AG

Dieses 2007 im Rahmen der INI.FAU-Kooperation gestartete Projekt soll die Integration von Fahrzeugfunktionen auf Steuergeräte analysieren und modellgetriebene Unterstützungsmöglichkeiten entwickeln. Die gewonnen Erkenntnisse sollen exemplarisch anhand der Integration aller Komponenten eines Fahrdynamikregelsystems auf einem AUTOSAR-Steuergerät überprüft werden.

In der Automobilindustrie ist es schon lange üblich, Fahrzeugfunktionen auf hohem Abstraktionsniveau modellbasiert zu entwickeln. Um frühzeitig Fehleinschätzungen bezüglich Laufzeit- und Ressourcenbedarf auszuschließen, ist es nötig, die entwickelte Software nicht nur zu simulieren sondern auch auf der Zielhardware testen zu können. Aufgrund von Kosten- und Sicherheitsanforderungen ist die Integration auf ein Steuergerät aber sehr zeitaufwändig und erfordert Expertenwissen, das einem Funktionsentwickler normalerweise nicht zur Verfügung steht. AUTOSAR (AUTomotive Open System ARchitecture) scheint sich als Standard für die Basissoftware auf Steuergeräten zu etablieren, doch durch die Neuheit dieses Standards gibt es noch keine Verfahren und Werkzeuge, um die Integration von Funktionen auf einem Steuergerät zu unterstützen.

Im Jahr 2008 wurden die Modellierungsmöglichkeiten in AUTOSAR im Bezug auf ihre Eignung bei Audi und auf mögliche Konflikte mit bestehenden Standards sowie mit bei Audi eingesetzten Technologien untersucht. Des Weiteren wurde die automatische Vervollständigung einer Reglerkomponente zu einer AUTOSAR-Softwarearchitektur prototypisch realisiert. Als zukünftige Unterstützungsmöglichkeiten bei der Integration kommen die automatische Konfiguration der Buskommunikation und das Scheduling der auszuführenden Prozesse in Frage.

### **1.2.9 Integrierte Werkzeug-Kette zur metamodellbasierten Modellierung und Ausführung von Software-Entwicklungsprozessen**

**Projektleitung:**

Hon.-Prof. Dr.-Ing. Detlef Kips

**Beteiligte:**

Dipl.-Inf. Ralf Ellner

Prof. Dr. Michael Philippsen

Dr.-Ing. Martin Jung

Dipl.-Inf. Johannes Drexler

Al-Hilank, Samir

**Laufzeit:** 1.10.2008–30.9.2011

**Förderer:**

BMWi

Aufgrund ständig wachsender Anforderungen, die an die Entwicklung komplexer Softwaresysteme gestellt werden, gewinnt die Einhaltung wohldefinierter Software-Entwicklungsprozesse (SWEPe) immer mehr an Bedeutung. Im Kontext umfangreicher, global verteilter Entwicklungsprojekte ist dabei insbesondere ein Trend zu organisationsübergreifenden, langlaufenden und dabei dynamisch veränderbaren Prozessen erkennbar. Zur effektiven Beschreibung und Unterstützung solcher Entwicklungsprozesse sind speziell geeignete Prozessmodellierungssprachen und eine mächtige Werkzeugunterstützung unverzichtbar.

Im Rahmen einer vom BMWi geförderten Kooperation mit dem Industriepartner develop group wurden im Berichtszeitraum verschiedene existierende Prozessmodellierungssprachen - darunter insbesondere das Software and Systems Process Engineering Metamodel (SPEM) der Object Management Group (OMG) - sowie diverse auf dem Markt verfügbare SWEP-Management-Werkzeuge untersucht und bewertet. Einige wichtige Resultate dieser Untersuchung wurden auf der Konferenz Software Engineering 2008 in München vorgestellt.

Die Ergebnisse der Untersuchung machten deutlich, dass der Markt für SWEP-Beschreibungs- und -Ausführungsumgebungen derzeit noch keine Lösungen bietet, die eine hinreichend präzise und flexible Modellierung von Entwicklungsprozessen sowie deren automatisierte Ausführung, Steuerung und Überwachung ermöglichen. Diese Lücke soll im Rahmen eines weiteren, umfangreicheren Kooperationsprojektes geschlossen werden, das vor kurzem angelaufen ist.

Ziel dieses Projektes ist es, auf Grundlage eines durchgängigen, metamodellbasierten Ansatzes eine integrierte Werkzeugkette für die Modellierung und Ausführung industrieller Software-Entwicklungsprozesse prototypisch zu realisieren. Im Hinblick auf die Praxistauglichkeit der Lösung liegt das Hauptaugenmerk dabei auf der Anpassbarkeit der Prozessmodelle an verschiedene industriellen Entwicklungsszenarien, auf der Anwenderfreundlichkeit der Prozessbeschreibung und auf einer weitgehenden Automatisierung der Prozessausführung, die zur Effizienzsteigerung in der Entwicklung entscheidend beiträgt. Diese charakteristischen Vorzüge sollen durch einen relativ hohen Formalisierungsgrad der Prozessmodellierung, durch eine weitgehende Generizität der Modellierungs- und Prozessausführungswerkzeuge sowie durch die Verwendung verbreiteter und akzeptierter Industriestandards (UML, SPEM) erreicht werden.

Dieses Projekt wird ebenfalls in Zusammenarbeit mit der develop group als Industriepartner durchgeführt und mit Mitteln des BMWi gefördert. Es wurde im Oktober 2008

mit drei wissenschaftlichen Mitarbeitern gestartet und ist auf insgesamt drei Jahre ausgelegt.

### **1.2.10 Softwareleitstand**

**Projektleitung:**

Prof. Dr. Michael Philippsen

**Beteiligte:**

Dipl.-Inf. (FH) Josef Adersberger

Norbert Tausch

**Beginn:** 1.11.2009

**Förderer:**

Bundesministerium für Wirtschaft und Technologie

**Mitwirkende Institutionen:**

QAware GmbH

**Kontakt:**

Prof. Dr. Michael Philippsen

Tel.: +49 9131 85-27625

Fax: +49 9131 85-28809

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

### **Prototypische Entwicklung eines neuartigen Werkzeugs zur Qualitätsabsicherung bei der Softwareentwicklung.**

Moderne Softwaresysteme werden sowohl fachlich, technisch als auch organisatorisch zunehmend komplexer: So steigt die Anzahl und der Vernetzungsgrad der zu realisierenden Anforderungen pro System stetig, die technischen Vorgaben z.B. an den Verteilungsgrad und die Zuverlässigkeit der Systeme werden komplexer und die Softwareentwicklung selbst findet zunehmend in global verteilten Teams und mit wachsendem Zeitdruck statt. Aus diesen Gründen wird es auch zunehmend schwieriger, Softwareentwicklungsprojekt fachlich, technisch und organisatorisch zu steuern.

Als Softwareleitstand bezeichnen wir ein Werkzeug, das leitenden Projektrollen wie dem Projektleiter, dem Softwarearchitekten, dem Anforderungsarchitekten und dem Entwicklungsleiter eine hohe Transparenz und damit verbesserte Steuerbarkeit von Softwareentwicklungsprojekten ermöglicht.

Transparenz herrscht dann, wenn sowohl Zusammenhänge zwischen den vielerlei Erzeugnissen eines Softwareentwicklungsprojekts als auch deren Eigenschaften schnell und gesamtheitlich zugänglich sind und entsprechend dem individuellen Informationsbedarf eines Projektbeteiligten aufbereitet sind.

Der Softwareleitstand ist ein Werkzeug, das den Zugang zu den Zusammenhängen (Traceability) und den Eigenschaften (Metriken) der Erzeugnisse von Softwareentwicklungsprojekten vereinheitlicht. Damit kann die Effizienz von Softwareentwicklungsprojekten maßgeblich gesteigert werden. Es sollen Erzeugnisse des Softwareentwicklungsprojekts (Artefakte) und ihre Zusammenhänge (Relationen), sowie zu den Artefakten zuordenbare Metriken zentral erfasst, integriert und analysiert werden können. Die entsprechenden Analysen werden in Form von Visualisierungen des Artefaktgraphen mit- samt den zugeordneten Metriken und Regelprüfungen durchgeführt.

### **1.2.11 Funkortung von Antennenpositionen**

**Projektleitung:**

apl. Prof. Dr.-Ing. Gabriella Kókai

**Beteiligte:**

Dipl.-Ing. (FH) Thorsten Edelhäuser

**Laufzeit:** 1.5.2008–30.4.2011

**Förderer:**

Fraunhofer Institut für Integrierte Schaltungen

**Mitwirkende Institutionen:**

Fraunhofer Institute für Integrierte Schaltungen

Im Jahr 2009 wurden Algorithmen zur Berechnung der Position und Orientierung der Empfangsantennen eines Funkortungssystems weiterentwickelt. Die Algorithmen berechnen selbstständig Messpunktkoordinaten, die eine schnelle und genaue Einmessung ermöglichen. Zur automatisierten Einmessung wurden hierzu Roboter verwendet. Unser Algorithmus berücksichtigt Hindernisse und die Empfangseigenschaften des Ortungssystems und kann Fehlmessungen wie Mehrwegeempfang in der Berechnung der Position und Orientierung der Empfangsantenne aussortieren.

Weiter werden Algorithmen entwickelt, um dynamische Bewegungsmodelle zu ermöglichen. Hier werden lernende Verfahren eingesetzt, um Modelle während der Laufzeit anzupassen. Innerhalb des Forschungsbereichs wurde eine beschreibende Darstellung von Trajektorien entwickelt.

### **1.2.12 Evolutionäre Agenten**

**Projektleitung:**

Dipl.-Inf. Stephan Otto

**Laufzeit:** 1.4.2008–31.5.2009

**Kontakt:**

Dipl.-Inf. Stephan Otto  
Tel.: +49 9131 85-27830  
Fax: +49 9131 85-28809  
E-Mail: Stephan.Otto@informatik.uni-erlangen.de

Die starke Vernetzung von Computersystemen hat in den letzten Jahren zu enormen Veränderungen in nahezu allen Bereichen geführt. Beispiele hierfür sind das Internet, Grid-Computing, Peer-to-Peer Netzwerke und darauf aufbauende Verfahren wie z.B. Agentensysteme. Diese Entwicklung ist partiell das Resultat verteilter Systeme und den damit entstandenen inhärent verteilten Problemen. Eine Konsequenz ist zunehmende Dezentralität beim Lösen verteilter Probleme, da zentrale Verfahren unzureichend dafür geeignet sind:

- Zentrale Ressourcen sind begrenzt in ihrer Fähigkeit, (alle notwendigen) Daten zu speichern, zu übertragen und zu verarbeiten,
- In unternehmensübergreifenden Geschäftsumgebungen existiert eine kommunikationseinschränkende Informationsasymmetrie,
- Dynamik: während zentral eine Lösung erstellt wird, hat sich das Problem bereits verändert.

In diesem Umfeld dynamischer, nicht zugreifbarer und verteilten Strukturen stellt die Adaption und Optimierung von Systemen und Geschäftsprozessen ein nach wie vor nur unzureichend gelöstes Problem dar. Im Allgemeinen sind Adaptions- bzw. Optimierungsverfahren so ausgelegt, dass Informationen für inhärent verteilte Probleme zentral gesammelt und bearbeitet werden, um ein möglichst gutes Ergebnis zu erzielen. Es existieren eine Reihe von speziellen Ansätzen zur verteilten Optimierung, deren Funktionsweise auf ein Problem bzw. eine eingeschränkte Anzahl von Problemen zugeschnitten ist. Häufig müssen die Akteure kooperativ zusammenarbeiten, um eine Lösung zu erreichen. Bei vorhandenen Optimierungsverfahren finden sich wesentliche, noch nicht ausreichend untersuchte Problembereiche der Verteilung (Daten, Ressourcen), Heterogenität und dynamische Umwelt. Hierbei liegt der Fokus insbesondere auf der Toleranz gegenüber dynamischen, verteilten und heterogenen Basisressourcen.

Dieses Projekt fokussiert ein gegenüber den genannten Problembereichen tolerantes Optimierungsverfahren. Während sogenannte Top-down Verfahren ausgehend von einem zentralen Ansatz arbeiten, wird in diesem Projekt ein sogenannter Bottom-up Ansatz verfolgt. Ebenso wie in natürlichen komplexen Systemen entstehen komplexe Softwaresysteme aus dem Zusammenspiel sogenannter Agenten, indem jeder Agent einfachen lokalen Verhaltensmustern folgt. Selbstorganisation und damit verbunden Adaption ist ein Hauptmerkmal komplexer Systeme. Im Rahmen dieses Projektes wird ein neues

generisches Konzept verteilter Optimierung mittels evolutionärer Agenten verfolgt. Es werden dezentrale Operatoren für die Selektion und Rekombination verwendet, die auf ökonomischen Marktmechanismen basieren. Damit kann der Flaschenhals zentraler Selektion aufgrund berechneter Fitnesswerte umgangen werden. Ein dezentrales bottom-up Adaption- und Optimierungsverfahren kann somit erforscht und in unterschiedlichen Szenarien erprobt werden. Die Methode basiert auf einem formalen Modell, welches den Adaptionmechanismus für die Anzahl und Strategie der einzelnen Agenten erklärt und damit die entstehende emergente Optimierung offenlegt.

Die Beiträge dieses Projektes sind im Wesentlichen wie folgt zusammenzufassen:

- Entwicklung eines neuartigen verteilten Evolutionären Algorithmus, um den üblicherweise zentral ablaufende Fitnessvergleich und die zentrale Selektion zu vermeiden
- Entwicklung von endogener Fitness und ihrer Auswirkung auf die Ergebnisqualität
- Es wurden neue lokale Selektionsverfahren entwickelt und miteinander verglichen.
- Um die grundlegende Fähigkeit des Verfahrens zu zeigen, wurden empirische Studien zur Takeover-time durchgeführt. Hiermit wurde gezeigt dass der Selektionsdruck vergleichbar mit klassischen evolutionären Verfahren ist.
- Ein neues Dezentralitätsmaß wurde entwickelt, um verteilte Ansätze hinsichtlich ihres Grades an Dezentralität einordnen zu können.
- Das entwickelte Konzept wurde an unterschiedliche Anwendungsfälle adaptiert und damit seine Leistungsfähigkeit unter Beweis gestellt.

Die Leistungsfähigkeit des entwickelten Ansatzes wurde im Rahmen einer Dissertation nachgewiesen, die im Frühjahr 2009 abgegeben und im September erfolgreich verteidigt wurde.

### **1.2.13 Optimierung von FIR-Filterstrukturen**

**Projektleitung:**

apl. Prof. Dr.-Ing. Gabriella Kókai

**Beteiligte:**

Dr.-Ing. Szilvia Zvada

Dipl.-Ing. Hans Holm Frühauf

**Laufzeit:** 1.1.2006–30.9.2009

**Kontakt:**

apl. Prof. Dr.-Ing. Gabriella Kókai

Tel.: +49 9131 85-28996

Fax: +49 9131 85-28809

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

Dank der rapiden Verbreitung elektronischer Systeme im Alltag rückten VLSI-Chips (very large scale integration) schnell in den Fokus der Forschung. Das Hauptziel in diesem Bereich ist der Entwurf von kleinen und schnellen Chips bei gleichzeitig niedrigem Energieverbrauch. Der Fortschritt der modernen Chipherstellungstechnologie und die zunehmende Packungsdichte haben es ermöglicht, dass heutige VLSI-Chips einige Millionen Transistoren enthalten. Aus der Sicht eines Chipdesigners bedeutet dies eine beträchtliche Zahl potenzieller Chipstrukturen bei der Suche nach einem optimalen oder annähernd optimalen Chip. So wird die Automatisierung des Designprozesses in zunehmendem Maße wichtig.

Im Fall digitaler Filter liegt die Aufmerksamkeit vor allem auf dem Design von FIR-Filtern (finite impulse response). Diese Filter werden allgemein verwendet, um digitale Datenströme gemäß einer linearen Funktion umzuwandeln, wie z.B. bei der Linearisierung durch Endverstärker oder bei der Kalibrierung von Audio- oder Videoempfängern. Wenn jedoch die Aufgabe eine nicht-lineare Transformation der Datenströme ist, muss ein manueller und daher zeitraubender Entwurf solcher Filter durchgeführt werden.

Das in diesem Projekt entwickelte evolFIR System schließt diese Lücke, indem es ein neuartiges Entwurfswerkzeug zur Verfügung stellt, das das Logikdesign der polynomischen FIR-Filter-Strukturen optimieren kann. Auf dieser Ebene des Chipdesigns sind Funktionselemente wie Addierer oder Multiplizierer und logische Primitive wie Verschiebe- und Verzögerungselemente miteinander kombiniert, um die benötigte Funktionalität der gewünschten Filter zu gewährleisten. Diese Funktionalität ist durch die Übertragungsfunktion des Filters definiert. Während des evolutionären Prozesses müssen wir einerseits sicherstellen, dass die Individuen stets exakt die vorgegebene Übertragungsfunktion beschreiben. Andererseits müssen die betrachteten Topologien bestimmte hardware-spezifische Anforderungen erfüllen, wie z.B. die begrenzte Anzahl an Eingängen der jeweiligen Blockelemente.

Die zentrale Aufgabe für den Evolutionsprozess in evolFIR ist es, eine kleine (möglichst wenige Blockelemente enthaltende), redundanzfreie Filterstruktur zu finden. Dies erreichen wir durch die Anwendung des AGGP-Verfahrens (Genetische Programmierung basierend auf attributierten Grammatiken) auf folgenden Weise:

- Die Individuen des evolutionären Prozesses sind spezielle Ableitungsbäume, die die mögliche Topologie der Funktionselemente und der logischen Primitive dar-

stellen. Da evolFIR die Anwendung von Multiplizierern in Filterkompositionen unterstützt, können auch polynomielle Übertragungsfunktionen optimiert werden.

- Mithilfe der Attribute und des speziellen Zufallsbaumgenerators von AGGP ist sichergestellt, dass ausschließlich solche Ableitungsbäume während der Optimierung erzeugt werden, die genau die Ziel-Übertragungsfunktion repräsentieren.
- Dieser spezielle Zufallsbaumgenerator berücksichtigt außerdem auch variierende Einschränkungen, die für den späteren Hardware-Synthese-Prozess relevant sind.
- Durch die Verwendung einer speziellen Darstellungsform (abstrakt-verkettete Ableitungsbäume), werden redundante Teile der erzeugten Filterstrukturen nicht nur reduziert, sondern komplett eliminiert.

Die obigen Einschränkungen und unsere besondere Darstellungsform zusammen führen dazu, dass der Lösungsraum des evolFIRs inhomogen und die Fitnessfunktion diskontinuierlich sind. Aus diesem Grund erfordert die Parametrisierung des evolutionären Kerns des evolFIRs besondere Aufmerksamkeit, um eine verfrühte Konvergenz des evolutionären Prozesses vorzubeugen.

Die Leistungsfähigkeit des entwickelten Ansatzes wurde im Rahmen einer Dissertation nachgewiesen, die im Dezember 2008 abgegeben und im Juni erfolgreich verteidigt wurde.

#### **1.2.14 Graphen und Graphtransformationen**

##### **Projektleitung:**

Prof. em. Dr. Hans Jürgen Schneider

**Laufzeit:** 1.10.2004–30.9.2010

##### **Kontakt:**

Prof. em. Dr. Hans Jürgen Schneider

Tel.: +49 9131 85-27620

Fax: +49 9131 85-28809

E-Mail: [schneider@informatik.uni-erlangen.de](mailto: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.

Im Jahre 2009 haben wir sowohl ein theoretisches Konzept untersucht als auch einen Implementierungsaspekt betrachtet:

- In den letzten Jahren haben wir das Konzept der unabhängigen Ableitungsschritte so verallgemeinert, dass es in jeder interessanten Kategorie, insbesondere in der Kategorie strukturiert markierter Graphen, angewandt werden kann. Wir haben nun die Untersuchungen zur Transformation von Ableitungssequenzen mit der Betrachtung paralleler Produktionen abgeschlossen, das sind Produktionen, die die Wirkung zweier oder mehrerer Produktionen in einem Schritt kombinieren. Es ist einfach, parallele Produktionen als Coprodukt mehrerer Produktionen zu definieren, aber im Fall der Graphen führt dies zu Ableitungsschritten, deren Einbettung nicht injektiv ist. Im Gegensatz zu anderen bekannten Ansätzen, sind unsere Untersuchungen nicht auf injektive Einbettungen beschränkt, und die Konstruktion ist somit ausreichend. Auf der Basis paralleler Ableitungsschritte kann man kanonische Ableitungen definieren. In der Theorie der Chomsky-Sprachen, benutzt diese Definition die Anordnung der Symbole von links nach rechts, die es bei Graphen nicht gibt. Kreowski hat 1976 eine Alternative vorgeschlagen: Unabhängige Ableitungsschritte werden zu parallelen zusammengefasst und die Komponenten auf die frühest mögliche Position verschoben. Eine kanonische Ableitung ist eine, in der keine Shift-Operationen mehr möglich sind. Kreowskis Definition kann unmittelbar auf unsere allgemeine, nicht auf Graphen beschränkte Darstellung übertragen werden. Die Details finden sich im Entwurf zu unserem Lehrbuch: <http://www2.informatik.uni-erlangen.de/Personen/schneide/gtbook/chapter5.pdf>
- Die kategorielle Behandlung der Graphtransformationen ist hochgradig generisch: Alle Beweise und Konstruktionen gelten für verschiedene Graphtypen. Nur die grundlegenden Operationen müssen für jede Anwendung detailliert beschrieben werden, die darauf aufbauenden kategoriellen Eigenschaften sind dann automatisch definiert. Da moderne Programmiersprachen generische Konzepte unterstützen, sieht es vielversprechend aus, den kategoriellen Ansatz zur

Beschreibung von Graphtransformationssystemen in Sprachen wie Java oder Haskell zu implementieren. Java benutzt Klassen von Objekten, aber es unterstützt die Mehrfachvererbung nicht wirklich. In unserer ersten Version definieren wir Schnittstellen (Interfaces) Cat, CatWithColimits, usw. Die grundlegenden Colimites (Coproduct, Coequalisator, initiales Objekt) werden als abstrakte Klassen eingeführt, deren Details von den speziellen Kategorien implementiert werden müssen. Da diese Colimites die Konstruktion aller anderen Colimites ermöglichen, wäre es sinnvoll, CatWithColimits mit einer Methode auszustatten, die Pushouts konstruiert, aber CatWithColimits ist ein Interface. Daher führen wir eine Klasse PushoutCreator ein, die die kategorielle Konstruktion unabhängig von einer speziellen Kategorie beschreibt. Jede Kategorie mit Colimites muss diese "Factory class" importieren. Eine zusammenfassende Darstellung dieser Version ist verfügbar: <http://www2.informatik.uni-erlangen.de/Personen/schneide/gtbook/appendix-b.pdf>

### 1.2.15 Zeitplanungsalgorithmen

**Projektleitung:**

PD Dr.-Ing. habil. Peter Wilke

**Beteiligte:**

Dipl.-Inf. Johannes Ostler

**Laufzeit:** 1.1.2004–1.10.2009

**Kontakt:**

PD Dr.-Ing. habil. Peter Wilke

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.

**Erlangen Advanced Time Tabling Software EATTS** ist die innovative Entwicklungs- und Produktionsumgebung zur Erstellung optimierter Zeitplanungen.

**Ressourcen**

Zeitplanungsprobleme treten in der Praxis in verschiedenen Formen auf: Schichtpläne, Fertigungspläne, Stundenpläne u.v.a. Allen gemeinsam ist, dass bestimmte Ereignis-

se unter Berücksichtigung von Randbedingungen möglichst optimal geplant werden müssen. Das Ergebnis der Planung ist dann ein Zeitplan. Im Beispiel der Schulplanerstellung wären die Ereignisse Schulstunden, denen Ressourcen wie Lehrer, Klassen und Räume zugeordnet werden müssen. Die Ressourcen werden in Typen unterteilt. Für jeden dieser Typen können beliebig viele Attribute vom Benutzer definiert werden.

Eine Zeitplanerstellung beginnt typischerweise mit der Erfassung der einzuplanenden Ressourcen. Diese kann durch Import eines Datenbestandes oder manuelle Erfassung geschehen.

### **Ergebnisse**

Als Ergebnisse der Planungsalgorithmen werden Zeitpläne erstellt. Diese können in verschiedenen Formaten gespeichert und angezeigt werden. So ist es z. B. möglich, verschiedene Sichten auf einen Plan zu erzeugen.

Typisch ist die Anbindung über einen Browser, d.h. den einzelnen Benutzern werden entsprechend ihren Privilegien die Sichten und Funktionen zur Verfügung gestellt.

### **Randbedingungen**

Die Beschreibung von Randbedingungen ist meist viel komplexer als die von Ressourcen und Ereignissen.

Zum Einen müssen die Randbedingungen exakt formuliert werden, zum Anderen darf die Übersichtlichkeit nicht verloren gehen, um z. B. Widersprüche oder Lücken entdecken zu können, die ja leider nicht automatisch gefunden werden können. Randbedingungen kommen in vielen Varianten vor, weshalb eine flexible Spezifikation notwendig ist. In der Spezifikation kann auf Ressourcen und/oder deren Attribute, die ja vom Benutzer definiert werden, zugegriffen werden. Abhängig vom Typ dieser Variablen, unter anderem Integer, Gleitkomma und Zeichenketten, stehen Verknüpfungs- und Vergleichsoperatoren zur Verfügung, um die Bedingungen zu formulieren. Zusätzlich werden die Parameter der Kostenfunktion gewählt, um bei einer Verletzung der Randbedingung die entsprechenden Strafpunkte zu berechnen.

Eine Besonderheit unserer Software ist, dass Randbedingungen nicht nur als "unbedingt einzuhalten (hard)" oder "nach Möglichkeit einzuhalten (soft)" klassifiziert werden können, sondern auch als "darf im Ausnahmefall verletzt werden (soft hard)". Somit kann die Verletzung bestimmter Randbedingungen im Ausnahmefall erlaubt werden. So kann beispielsweise flexibel auf den Ausfall von Ressourcen reagiert werden, indem ein neuer Zeitplan erstellt wird, der möglichst wenig Abweichungen vom bisherigen Plan hat, z. B. muss ja nicht der gesamte Stundenplan aller Schüler neu erstellt werden, nur weil ein Lehrer krank geworden ist, oder ein Klassenraum wegen eines Rohrbruchs nicht benutzbar ist. In diesen Fällen soll nur ein Vertretungsplan erstellt werden.

### **Algorithmen**

Herzstück der Planung sind die verwendeten Algorithmen. Abhängig von der Natur der Randbedingungen und den gewünschten Eigenschaften kann aus einer Vielzahl von bereits implementierten Algorithmen ausgewählt werden: Genetische Algorithmen - Evolutionäre Algorithmen - Branch-and-Bound - Tabu Search - Simulated Annealing - Graphenfärbung - Soft Computing - Schwarm Intelligenz.

Für den Einstieg stehen vorkonfigurierte Algorithmen zur Verfügung, der fortgeschrittene Benutzer kann aber die Parameter der Algorithmen an seine Bedürfnisse anpassen oder neue Algorithmen implementieren. Alle diese Algorithmen können in Experimenten beliebig zu Berechnungssequenzen kombiniert werden. Die Konfiguration eines Experiments kann abgespeichert werden und z. B. als Vorlage für ein neues Experiment dienen oder nochmals ausgeführt werden.

### **Ausführung von Experimenten**

Die Algorithmen werden entweder auf einem dedizierten Server ausgeführt und bei Bedarf über das TCP/IP-Protokoll auf weitere Rechner verteilt. Die Abbildung zeigt den Dialog zur Auswahl und zum Start der Experimente und die Übersicht der laufenden Experimente. Der Browser verbindet sich in regelmäßigen Abständen automatisch mit dem Server und erhält von diesem den aktuellen Stand der Berechnung. Dieser Statusinformationen beinhalten unter anderem die Kosten des bisher besten gefundenen Plans sowie eine Abschätzung für die verbleibende Berechnungszeit. Nach Beendigung der Berechnung werden die Ergebnisse gespeichert und die Dateien, die zur Visualisierung der Pläne nötig sind erstellt. Der Planer kann nun entscheiden, ob die Qualität der gefundenen Lösung ausreichend ist, oder ob er auf ihrer Basis weitere Optimierungsläufe starten will.

### **Ergebnisse**

Als Ergebnisse der Planungsalgorithmen werden Zeitpläne erstellt. Diese können in verschiedenen Formaten gespeichert und angezeigt werden. So ist es z.B. möglich verschiedene Sichten auf den Plan zu erzeugen.

Typisch ist die Anbindung über einen Browser, d.h. den einzelnen Benutzern werden entsprechend ihren Privilegien die Sichten und Funktionen zur Verfügung gestellt.

### **Zusammenfassung**

Die Software ist in Java implementiert und damit plattform-übergreifend verfügbar, insbesondere für die Betriebssysteme Windows und Linux.

Für den Betrieb von EATTS werden folgende frei verfügbare kostenlose Software-Produkte benötigt:

- ein JavaScript-fähiger Browser zur Anzeige der Bedienoberfläche

Optional kann ein dedizierter EATTS-Server konfiguriert werden. Dazu wird benötigt:

- Java Laufzeitumgebung (JRE Java Runtime Environment) (min v5.0),
- über TCP/IP Netzwerk erreichbare Rechner zur verteilten Berechnung (optional).

Im Jahr 2008 wurde die Struktur der Algorithmen optimiert um die nebenläufige Berechnung zu beschleunigen. Dies soll in Zukunft auf Rechner mit Multi-Core-Prozessoren ausgedehnt werden.

Da es sich die Installation der Software durch die potentiellen Nutzer als zu komplex herausgestellt hat, wurde eine abgespeckte Version implementiert, die keine Datenbank mehr benötigt, sondern deren Datenhaltung und Austausch auf XML-Dokumenten basiert. Zusätzlich wird eine Variante angeboten, bei der die Nutzer ihre Experimente auf einem an der Universität Erlangen installierten Server rechnen lassen können.

Die Oberfläche der Software wurde komplett als web-basierte Anwendung reimplementiert.

Auf der CeBIT 2009 wurde die neue Version der Software vorgestellt, die jetzt EATTS Erlangen Advanced Time tabling System heißt.

### **1.2.16 International Collegiate Programming Contest an der FAU**

**Projektleitung:**

Prof. Dr. Michael Philippsen

**Beteiligte:**

Dipl.-Inf. Tobias Werth

Dipl.-Inf. Marc Wörlein

Dipl.-Inf. Alexander Dreweke, B. Sc. mult.

**Beginn:** 1.11.2002

**Kontakt:**

Dipl.-Inf. Tobias Werth

Tel.: +49 9131 85-28865

Fax: +49 9131 85-28809

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

Die Association for Computing Machinery (ACM) richtet seit Jahrzehnten den International Collegiate Programming Contest (ICPC) aus. Dabei sollen Teams aus je drei Studenten in fünf Stunden neun bis zehn 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 und Mathematik, wie z.B. Graphen, Kombinatorik, Zeichenketten, Algebra 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 seit dem Jahr 2009 im Einzugsbereich des Northwestern European Regional Contest (NWERC), an dem u.a. auch Teams aus der Großbritannien, den Benelux-Staaten und Skandinavien teilnehmen.

Die Sieger aller regionalen Wettbewerbe der Welt (und einige Zweitplatzierte) erreichen die World Finals, die im Frühjahr des jeweils darauffolgenden Jahres (2010 in Harbin) stattfinden. Im Jahr 2009 fanden zwei lokale Wettbewerbe an der FAU statt. Der im Wintersemester 2009 mit dem Ziel ausgetragene Mannschaftswettbewerb, neue Studierende für die Wettbewerbe zu begeistern, zog eine Rekordzahl von 24 Teams an. Jedes Team bestand aus maximal drei Studenten. Außerdem nahmen noch Teams der TU München sowie der Universität Konstanz online am Wettbewerb teil.

Im Sommersemester fand zum wiederholten Mal das Hauptseminar "Hallo Welt! - Programmieren für Fortgeschrittene" statt, um Studierende verschiedener Fachrichtungen in Algorithmen und Wettbewerbsaufgaben zu schulen. Der Wettbewerb im Sommersemester diente der Auswahl der studentischen Vertreter der FAU für den NWERC 2009, der dieses Jahr von der FAU in Nürnberg ausgerichtet wurde. Insgesamt nahmen am lokalen Ausscheidungskampf 25 Studierende der verschiedensten Fachrichtungen teil. Die besten neun bildeten Dreiermannschaften (der Zehntplatzierte wurde als Ersatzmann ausgewählt). Das beste Team gewann den nordwesteuropäischen Wettbewerb NWERC souverän und setzte sich gegen die internationale Konkurrenz durch. Die beiden anderen Teams landeten im Mittelfeld der 64 Teams auf den Plätzen 27 und 36. Auch 2009 zeigte das Trainingslager somit den gewünschten Erfolg, da das beste Team zu den Weltmeisterschaften in Harbin/China im Frühjahr 2010 fahren und dort als einziges deutsches Team teilnehmen darf.

Der Regionalauscheid für Nordwesteuropa war ein besonderer Erfolg. Dabei kamen 64 Teams verschiedener Universitäten nach Nürnberg, um die Programmierkrone für Nordwesteuropa zu erkämpfen. 2010 wird der Wettbewerb in Bremen stattfinden.

### **1.2.17 Embedded Systems Institute**

#### **Projektleitung:**

Prof. Dr. Michael Philippsen

#### **Beteiligte:**

Dipl.-Inf. Philipp Janda

Dipl.-Inf. Stefan Kempf

Dipl.-Inf. Georg Dotzler

**Beginn:** 1.9.2007

Das im September 2007 als Interdisziplinäres Zentrum an der Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) gegründete "ESI - Embedded Systems Institute" hat sich die fächerübergreifende Koordination und Organisation der Forschung, Lehre und Weiterbildung im Bereich Eingebetteter Systeme zum Ziel gesetzt.

Über das ESI werden an der Universität vorhandene Kompetenzen mit den Interessen, Aktivitäten und Zielen der einschlägigen Großindustrie und des Mittelstands auf dem Gebiet des Entwurfs Eingebetteter Systeme vernetzt.

Unternehmen erhalten durch das ESI Zugriff auf neueste Forschungsergebnisse sowie die Möglichkeit, gemeinsam Entwicklungsprojekte durchzuführen, Kontakte zu knüpfen und Kooperationspartner zu finden. Das ESI bündelt die Kompetenzen der Lehrstühle und macht sie für Kooperationsprojekte nutzbar. Aktuelle Forschung lässt sich damit schneller in Produkte umsetzen. Beschleunigt wird auch der Aufbau gemeinsamer Forschung. Schließlich dient das ESI auch als Schnittstelle zum frühzeitigen Zugriff auf Studierende und fachlich qualifizierten Nachwuchs.

Der Lehrstuhl für Informatik 2 (Prof. Dr. Michael Philippsen) gehört zu den aktiv beteiligten Gründungsmitgliedern des ESI und führt in diesem Rahmen Forschungsprojekte durch.

Weitere Informationen finden Sie auch unter: <http://www.esi.uni-erlangen.de>

### **1.3 Projektunabhängige Publikationen**

- Beyler, Jean Christophe ; Klemm, Michael ; Clauss, Philippe ; Philippsen, Michael: A meta-predictor framework for prefetching in object-based DSMs . In: Concurrency and Computation: Practice and Experience 21 (2009), Nr. 14, S. 1789-1803
- Dreweke, Alexander ; Fischer, Ingrid ; Werth, Tobias ; Wörlein, Marc: Text Mining in Program Code . In: Song, Min ; Wu, Yi-Fang Brook (Hrsg.) : Handbook of Research on Text and Web Mining Technologies. Hershey : Idea Group Publishing, 2009, S. 626-645. - ISBN 1599049902
- Edelhäuser, Thorsten ; Kókai, Gabriella: Autonomous Configuration Method for Real-Time Location Systems . In: Suess, Martin ; Arslan, Tughrul (Hrsg.) : Proceedings NASA/ESA Conference on Adaptive Hardware and Systems (AHS-2009) (NASA/ESA Conference on Adaptive Hardware and Systems (AHS-2009) NASA/ESA Conference on Adaptive Hardware and Systems (AHS-2009) Co-located with Design Automation Conference (DAC-2009) July 29 – August 1, 2009 Moscone Convention Center San Francisco California, USA July 29-August 1, 2009). 2009, S. 265-272.

- Gnezdilov Alexandr ; Wittmann, Stefan ; Helwig, Sabine ; Kókai, Gabriella: Acceleration of a Relative Positioning Framework . In: International Journal of Computational Intelligence Research 5 (2009), Nr. 2, S. 130-140
- Grimm, Robert ; Edelhäuser, Thorsten ; Kókai, Gabriella: An Efficient Configuration Method for RTLs . In: Jacak, Witold ; Affenzeller, Michael (Hrsg.) : Proceeding 2nd International Symposium on Logistics and Industrial Informatics (2nd International Symposium on Logistics and Industrial Informatics Linz Austria September 10 and 11). 2009, S. 148-153.
- Kempf, Stefan: A Language Independent JIT compiler library . Erlangen-Nürnberg, Friedrich-Alexander-Universität, Dipl-Arb., 2009. - 116 Seiten.
- Klemm, Michael: Reparallelization and Migration of OpenMP Applications in Grid Environments . Aachen : Shaker Verlag, 2009. Zugl.: Erlangen, Friedrich-Alexander-Universität Erlangen-Nürnberg, Diss., 2009. - 272 Seiten. ISBN 978-3-8322-7973-8
- Klemm, Michael ; Bezold, Matthias ; Gabriel, Stefan ; Veldema, Ronald ; Philippsen, Michael: Reparallelization Techniques for Migrating OpenMP Codes in Computational Grids . In: Concurrency and Computation: Practice and Experience 21 (2009), Nr. 3, S. 281-299
- Otto, Stephan: Ein agentenbasierter evolutionärer Adaptions- und Optimierungsansatz für verteilte Systeme . Erlangen, Friedrich-Alexander-Universität Erlangen-Nürnberg, Diss., 2009. - 209 Seiten.
- Schell, Dominic: Dynamische Programm-Code-Verwaltung und -Optimierung für eingebettete Systeme . Erlangen, Friedrich-Alexander-Universität Erlangen-Nürnberg, Diss., 2009. - 189 Seiten.
- Schmidt, Karsten ; Janda, Philipp: Erfahrungen bei der modellbasierten Entwicklung von Fahrwerksregelfunktionen im AUTOSAR Umfeld und notwendige Entwicklungsschritte . In: Giese, Holger ; Huhn, Michaela ; Nickel, Ulrich ; Schätz, Bernhard (Hrsg.) : Tagungsband des Dagstuhl-Workshop MBEES: Modellbasierte Entwicklung eingebetteter Systeme (MBEES: Modellbasierte Entwicklung eingebetteter Systeme, Dagstuhl-Wadern, 22.-24.04.2009), 2009, S. 67-74.
- Veldema, Ronald ; Philippsen, Michael: Tapir: Language Support to Reduce the State Space in Model-Checking . In: Fischer, Stefan ; Maehle, Erik ; Reischuck, Rüdiger (Hrsg.) : Informatik 2009 - Im Focus Leben (ATPS 2009 - 4. Arbeitstagung Programmiersprachen Lübeck 01.10.2009). Bonn : GI Gesellschaft für Informatik, 2009, S. 364. (Lecture Notes Informatics Bd. 2860-74) - ISBN 978-3-88579-248-2

- Werth, Tobias ; Dreweke, Alexander ; Wörlein, Marc ; Fischer, Ingrid ; Philippsen, Michael: DAG Mining in Procedural Abstraction . In: Cao, L. ; Yu, P. S. ; Zhang, C. ; Zhang, H. (Hrsg.) : Data Mining for Business Applications. Berlin/Heidelberg : Springer-Verlag, 2009, S. 209-224. - ISBN 978-0-387-79419-8
- Werth, Tobias ; Floßmann, Tobias ; Klemm, Michael ; Schell, Dominic ; Weigand, Ulrich ; Philippsen, Michael: Dynamic Code Footprint Optimization for the IBM Cell Broadband Engine . In: Porter, Adam ; Votta, Larry ; Pankratius, Victor (Hrsg.) : Proc. ICSE Workshop on Multicore Software Engineering (IWMSE'09 Vancouver, Canada 18.05.2009). New York, NY : IEEE, 2009, S. 64-72. - ISBN 978-1-4244-3718-4

## 1.4 Studien- und Abschlussarbeiten

- Studienarbeit: Vergleich von Programmiermodellen für die CellBE-Architektur. Bearbeiter: Bernd Schoebel (beendet am 12.01.2009); Betreuer: Dipl.-Inf. Tobias Werth; Prof. Dr. Michael Philippsen
- Diplomarbeit: Entwurf und Implementierung einer Cilk-Erweiterung für den Cell-Prozessor. Bearbeiter: Silvia Schreier (beendet am 31.01.2009); Betreuer: Dipl.-Inf. Tobias Werth; Prof. Dr. Michael Philippsen
- Diplomarbeit: Konzeption und Implementierung eines Verfahrens zur strukturellen Äquivalenzklassen- und Grenzwerttestüberdeckung. Bearbeiter: Stephan Fritz (beendet am 02.02.2009); Betreuer: Dr.-Ing. Norbert Oster; Prof. Dr. Michael Philippsen
- Studienarbeit: Entwurf und Implementierung einer browserbasierten Bedienoberfläche für eine Zeitplanungssoftware. Bearbeiter: Eugen Kremer (beendet am 11.2.2009); Betreuer: PD Dr.-Ing. habil. Peter Wilke
- Studienarbeit: Automatische Generierung von AUTOSAR Software Component Descriptions. Bearbeiter: Christopher Mutschler (beendet am 15.4.2009); Betreuer: apl. Prof. Dr.-Ing. Gabriella Kókai
- Diplomarbeit: A Language Independent JIT compiler library. Bearbeiter: Stefan Kempf (beendet am 20.4.2009); Betreuer: Ronald Veldema, Ph.D.; Prof. Dr. Michael Philippsen
- Diplomarbeit: Verfolgbarkeit bei der modellbasierten Softwareentwicklung auf Basis von Geschäftsprozessmodellen. Bearbeiter: Ninh Nguyen Duy (beendet am 30.06.2009); Betreuer: Dipl.-Inf. (FH) Josef Adersberger; Prof. Dr. Michael Philippsen

- Diplomarbeit: Konzeptionierung und Pilotierung eines automatisierten Modultests. Bearbeiter: Soufiane ben Abdelfatteh Jarraya (beendet am 11.08.2009); Betreuer: Dr.-Ing. Norbert Oster; Prof. Dr. Michael Philippsen
- Diplomarbeit: Laufzeitparallelisierung von OpenMP/Java-Programmen für die Ausführung auf GPUs. Bearbeiter: Georg Dotzler (beendet am 01.09.2009); Betreuer: Ronald Veldema, Ph.D.; Dr.-Ing. Michael Klemm; Prof. Dr. Michael Philippsen
- Diplomarbeit: Diskbased Backup Data Deduplication for Large Scale Virtual Machine Farms. Bearbeiter: Thomas Glanzmann (beendet am 15.09.2009); Betreuer: PD Dr.-Ing. habil. Peter Wilke
- Diplomarbeit: Entwurf und Implementierung von Algorithmen für Zeitplanungsprobleme. Bearbeiter: Helmut Killer (beendet am 30.09.2009); Betreuer: PD Dr.-Ing. habil. Peter Wilke
- Studienarbeit: Untersuchung der Traceability von Merkmalen auf Programmelemente. Bearbeiter: Miao Tang (beendet am 08.10.2009); Betreuer: Dipl.-Inf. (FH) Josef Adersberger; Prof. Dr. Michael Philippsen
- Diplomarbeit: Aufbau einer adaptiv konfigurierbaren Routenführung für einen Forschungs-Navigationskern: Entwicklung und Test eines vertikalen Prototyps. Bearbeiter: Bernd Stürzenhofecker (beendet am 14.10.2009); Betreuer: PD Dr.-Ing. habil. Peter Wilke; apl. Prof. Dr.-Ing. Gabriella Kókai
- Studienarbeit: Untersuchung von Optimierverfahren zur Kalibrierung von Phasenschiebern einer Multiphasenantenne. Bearbeiter: Roman Koch (beendet am 14.10.2009); Betreuer: apl. Prof. Dr.-Ing. Gabriella Kókai
- Studienarbeit: Analyse des Lattice-Boltzmann-Verfahrens auf der CellBE-Architektur. Bearbeiter: Dimitrij Kotrev (beendet am 19.10.2009); Betreuer: Dipl.-Inf. Tobias Werth; Prof. Dr. Michael Philippsen
- Diplomarbeit: Dynamische Overlays für die CellBE-Architektur. Bearbeiter: Tobias Floßmann (beendet am 23.10.2009); Betreuer: Dipl.-Inf. Tobias Werth; Prof. Dr. Michael Philippsen
- Diplomarbeit: Traceability von Komponentenmodellen zu Code über in UML integrierte Reflexion-Modelle. Bearbeiter: Christian Neuhaus (beendet am 04.11.2009); Betreuer: Dipl.-Inf. (FH) Josef Adersberger; Prof. Dr. Michael Philippsen

- Studienarbeit: Untersuchung der Traceability-Vorgaben in ausgewählten Prozess- und Reifegradmodellen der Softwareentwicklung. Bearbeiter: Chuanru Ma (beendet am 02.12.2009); Betreuer: Dipl.-Inf. (FH) Josef Adersberger; Prof. Dr. Michael Philippsen