1 Chair of Computer Science 2 (Programming Systems)

Anschrift: Martensstr. 3, 91058 Erlangen
Tel.: +49 9131 85-27621
Fax: +49 9131 85-28809
E-Mail: info@i2.informatik.uni-erlangen.de

Head:
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
Secretary:
Agnes Brütting
Waltraud Rück (until June, 30 2009)
Margit Zenk (since September, 1 2009)
Scientific Coworkers:
Dipl.-Inf. Alexander Dreweke, B. Sc. mult.
Dipl.-Ing. (FH) Thorsten Edelhäußer
Dipl.-Inf. Ralf Ellner
Dipl.-Inf. Philipp Janda
Dipl.-Inf. Stefan Kempf (since June, 30 2009)
Dr.-Ing. Norbert Oster
Ronald Veldema, Ph.D.
Dipl.-Inf. Tobias Werth
PD Dr.-Ing. habil. Peter Wilke (until November, 6 2009)
Dipl.-Inf. Marc Wörlein
Guests/Scholarship holders:
Dipl.-Inf. (FH) Josef Adersberger
Dipl.-Inf. Johannes Drexler
Dr.-Ing. Martin Jung
Dr.-Ing. Stephan Otto
Dr.-Ing. Szilvia Zvada
Technical/Administrative Staff:
Dipl.-Inf. Samir Al-Hilank
Dr.-Ing. Klaudia Dussa-Zieger
Dr. Karsten Schmidt

The Chair of Computer Science 2 (programming systems) was founded in 1972 and is headed by Prof. Michael Philippsen (as the successor of Prof. H.-J. Schneider) since April 2002. Closely associated with the programming systems group are the professorship for Didactics of Computer Science and the professorship for Open Source Software.

1.1 Focus of research

The main research topics in the programming systems group are programming of parallel or distributed systems and programming of embedded or mobile systems. Software (and its development) for such systems should ideally be as complex, portable, maintainable and robust as existing software for single core systems and workstations. It is our long-term goal to allow applications to take full advantage of the available computing and network power. A particular focus lies on programming systems for multi-cores because more and more cheap multi-core high-performance parallel hardware (for example graphics cards or FPA-Hardware) is available. This will have an unpredictable impact on the future of the software landscape. Research results of the group are always evaluated by means of prototypes and demonstrators.

Important Research Areas

- **Exploit the available parallelization potential.** In the future the clock rate of multi-core systems will grow only slowly whereas the number of cores will grow. This makes it necessary to exploit the parallelization potential of already older, existing software to allow it to benefit from the new hardware. As a consequence, in most application areas a change to parallel computing is unavoidable. Therefore, the programming systems group develops tools to support the programmer interactively in reengineering existing sequential applications. It also develops architectural patterns for new software projects that scale automatically to support a growing number of cores.

- **Achieve portability in high-performance applications.** Up to the present, application programmers achieve the best possible performance results only if they handle latency issues and communications between different components of the system manually, optimize their code with hardware specific ”tricks” and split their application into multiple sections to outsource them to other hardware (for example graphics cards). To change this situation, the programming systems group researches the performance impact of higher programming abstraction layers that would improve programming productivity and software portability. The improvements are caused by generated code that allows the distribution of the program onto multiple heterogeneous system components to permit parallel execution. The higher abstraction layer makes the communication between the components transparent for the developer. To increase the efficiency of this approach it is necessary to give the programmer the possibility to express available domain knowledge in the programming language. For the higher abstraction layer, the details of the hardware architecture are hidden from the developer (for example by library functions or programming language extensions).

- **Adapt the degree of parallelism dynamically**. High-performance applications are often developed for a fixed number of cores. As requested cluster nodes of a batch system are statically assigned for a fixed time period, inefficient reservation gaps are unavoidable. Similar problems appear in multi-threaded applications on multi-core systems. The programming systems group works on the dynamic adaptation of the extent of parallelism by the means of code transformations (under consideration of the resulting data redistribution) and operating system interactions. As control flow based synchronisation measures interfere with the necessary analyzes, the programming systems group researches new programming constructs that can replace the existing ones and allow to specify the synchronisation in a data-centric way.

- **Develop Testing for Parallelism**. In software engineering, testing has always assumed an important role. Code coverage, test data generation, reliability assessment etc. are tools of the trade. Unfortunately, current research insufficiently covers the indeterminism caused by concurrency. To deal with that issue, the programming systems group develops tools that consider (based on the coverage criteria) interleavings of parallel threads in their test data generation. This topic also includes research on operating systems and schedulers. As concurrency considerably increases the search space of the test generation it is necessary to develop infrastructures that allow the test generation and execution on a cluster.

- **Improve of Software Development Processes**. The current development practice of complex, business or security critical software in global distributed teams (commonly found in the software industry) demands compliance with well-defined software development processes. To support the enforcement of this requirement, appropriate development tools are used. The corresponding research area is covered by the Practical Software Engineering research group that is lead by the honorary professors Dr. Bernd Hindel and Dr. Detlef Kips. Both possess long term experience in industrial software projects as managers of medium size software companies. The goal of the Practical Software Engineering group is the development of a machine executable notation for modelling of software development processes. For that purpose the research group examines the semi-automatic retrieval of traceability information from the artefacts of different tools and notations as well as the model based development, integration and configurations of software components, used in the design of automotive embedded systems.

1.2 Focus of research

The focal points of the **programming systems research group** are programming systems for both parallel and distributed systems as well as for embedded and mobile systems. Software (and its creation) for these platforms should not be more complex than regular desktop software for single processor systems. It should also be as portable, maintainable, and robust. Our long-term aim is to deliver the full performance of the hardware (both computing and communication performance) to the application, in spite of a high level of abstraction that is needed for the engineering of demanding applications.

In the area of **embedded systems** we have mainly worked on reducing the amount of memory required for the code by means of clever compiler optimizations. In the area of **parallel systems** we have continued to improve their programmability in 2008. We have worked on (homogeneous) clusters as well as grids that are composed of heterogeneous nodes. In 2008 we have contributed to programming systems for multicores, especially parallelization and optimization for multicore architectures.

The honorary professors Dr. Bernd Hindel and Dr. Detlef Kips are both general managers of medium size software companies and have a long term experience in industrial software projects. They lead the **Practical Software Engineering research group** that investigates scientific methods to develop complex software systems based on well-defined processes. The research focus is on systematic descriptions, modelling and evaluation of software development processes and their tool based application to real world problems. Adequate meta-models, notations and evaluation methods are analyzed and their practical applicability is assessed.

1.2.1 Compiler-supported parallelization for multi-core architectures

Project manager:
Prof. Dr. Michael Philippsen
Project participants:
Dipl.-Inf. Tobias Werth
Dipl.-Inf. Dominic Schell
Start: 1.3.2007
Contact:
Dipl.-Inf. Tobias Werth

Phone: +49 9131 85-28865
Fax: +49 9131 85-28809
E-Mail: tobias.werth@informatik.uni-erlangen.de

Several issues significantly retard the development of quicker and more efficient computer architectures. Traditional technologies can no longer contribute offer more hardware speed. Basic problems are the divergent ratio of the latencies of memory access and CPU speeds as well as the heat and waste of energy caused by increasing clock rates.

Homogeneous and heterogeneous multi-core architectures were presented as a possible answer and offer enormous performance to the programmer. The decreasing clock rates help avoid most of the above problems, while the multiplied hardware can still deliver high performance since more arithmetic operations can be executed per time unit with less energy. Potentially, performance can increase even further by specialization of some hardware components. For example, often the latency problem is attacked with a multi-tiered memory hierarchy and lots of caches.

But there is no free lunch. It seems to be quite difficult to make multi-core architectures deliver their theoretically available performance into applications. Only with a lot of expertise in both the application domain and the specifics of the multi-core platform at hand and only with enough time to invest into tuning endeavors, one can make multi-core programs run fast.

From the point of view of a programming systems research group, there are - among others - the following open questions: What kind of support can a modern compiler offer to the programmer that develops applications for multi-core architectures? How much context knowledge is necessary in order to make reasonable decisions for parallelization? Which part of the available performance can be used by the programmer with a reasonable amount of effort without detailed knowledge about the features and quirks of the underlying architecture? Which tools are necessary for debugging and for finding bottlenecks in applications that run on multi-core architectures? How can they be designed?

It is the intention of this new research project to answer these questions for a restricted application domain. We have selected the Lattice-Boltzmann-Method (LBM) that is mostly used in computational fluid dynamics as our problem domain. Caused by its lattice structure and its manageable number of data dependencies between the single lattice points, it is comparatively straightforward how to parallelize it. Hence, our compiler research can focus on the above questions.

The heterogeneous CellBE architecture is selected as target architecture due its good performance on a single chip. It consists of a PowerPC core (PPU) and eight Synergistic Processing Units (SPUs), which can do computations in parallel. The SPUs offer only a small local store (256kB) for both program code and data. In order to achieve maximal performance one has to utilize the local store optimally.

The SPUs can access only their small local store (256kB) directly. This space has to be used for code, stack and data. The idea of our program loader is as follows: not the whole program code is executed at the same time in order to execute the program properly.

A dissertation that was finished in 2009 a novel kind of memory management for program code was developed. Therefore, a loader fragments the code dynamically at runtime (from basic blocks to functions) and loads these fragments into a code cache instead of loading the whole program into memory. If memory is low within the code cache, a garbage collector automatically removes currently unused code fragments. Several garbage collector techniques were investigated. Benchmarks on the CellBE architecture showed that the loader can save up to 80

Furthermore, we have investigated a free library for simulations of the lattice boltzmann method (OpenLB). This was done by evaluating several approaches for the port and the parallelization that were implemented in prototypes. For the 2D and the 3D case, we have developed concepts and structures, e.g. to simplify the programming with the vector instruction set as on the CellBE architecture. In matters of usability and performance we could achieve satisfactory results with a first port to the CellBE architecture.

We have also significantly improved the program loader for the CellBE architecture by removing the threshold of 256kb for the program code. Therefore, we have compared different overlaying concepts in order to get the advantages from both the dynamical approach of the program loader and the static overlaying techniques of a overlay manager. Developing a prototypical implementation, it turned out that this method is promising. As part of further research, we have to improve the runtime performance, however.

1.2.2 Graph-based procedural abstraction

Project manager:
Prof. Dr. Michael Philippsen
Project participants:
Dipl.-Inf. Alexander Dreweke, B. Sc. mult.
Dipl.-Inf. Marc Wörlein
Dipl.-Inf. Tobias Werth
Start: 1.4.2006
Contact:
Prof. Dr. Michael Philippsen
Phone: +49 9131 85-27625
Fax: +49 9131 85-28809

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

Embedded systems are even nowadays programmed in a machine-oriented fashion. The high level of abstraction and comfort one is used to in the development of desktop applications (object-orientation, garbage collection, exception handling, parallelism, aspect orientation, etc.) seems to be far away for embedded systems. Thus portability, robustness, and maintenance of the software are substantially impaired. This is also an important economical problem because Europe is not yet dominated by the USA in this field. So the long term goal must be to gradually increase the level of abstraction for programming embedded systems, i.e. optimization techniques are needed that achieve small and energy efficient code in spite of the increased abstraction level.

Apart from the obvious question how the well-known standard compiler optimization techniques affect code size, new specific questions arise for embedded systems. While the RAM consumption of an application plays hardly a role on desktop systems, it is crucial for embedded systems. Object-oriented code and especially libraries offer a substantial and mostly unused potential to reduce code size by means of procedural abstraction. Not only code size, but energy efficiency is a target for code optimization on embedded systems. Possibly in cooperation with the operating system, compiler optimizations are crucial. A challenging problem is how to deal with the non-uniform memory hierarchy, that not only consists of registers, caches, and main memory. But in addition, there is another layer, e.g. Flash memory. Another question for embedded systems is wether they can be programmed with the illusion of a uniform memory hierarchy. Is it possible to extract information about data locality through statical analysis? What kind of optimizations are practicable with this information? Can statical analyzes and runtime mechanisms benefit from each other in this specific area? How can prefetch and post-store commands be generated in the code that help improve latency and energy consumption?

A common method to reduce the size of program code is procedural abstraction (PA): repeated code portions in a program are identified and moved into a single new procedure. Instead of the extracted code portions, calls to the newly created procedure are inserted. This removes redundancy from the program and thus reduces its size. Earlier PA algorithms regarded a program as a sequence of instructions and searched for subsequences that appear frequently. However, if the instruction sequence is altered within a subsequence, they are no longer recognized by sequence matching algorithms. Hence, the result is suboptimal. To solve this problem we transform the instructions of a basic block into a data flow graph (DFG) and use a graph mining tool to search for common fragments in the DFGs of ARM assembly codes that are often used in embedded systems. In cooperation with the project ParSeMiS that is concerned with general optimizations of graph miners, specific properties of the PA domain are exploited. Our research has focused mainly on the analysis optimization of the correct reconstruction of data-flow graphs. The more accurate the data-flow analysis is, the more size reduction can be archived, compared to the traditional sequential approaches. In addition, various extraction mechanisms have been refined. These mechanisms are used to extract code fragments that have been identified to be frequent by ParSeMiS. The designed extraction mechanisms are as size-efficient as possible and standardize semantic equivalent fragments in a way that they can be extracted together.

1.2.3 ParSeMiS - the Parallel and Sequential Mining Suite

Project manager:
Prof. Dr. Michael Philippsen
Project participants:
Dipl.-Inf. Marc Wörlein
Dipl.-Inf. Alexander Dreweke, B. Sc. mult.
Dipl.-Inf. Tobias Werth
Start: 1.5.2006
Contact:
Prof. Dr. Michael Philippsen
Phone: +49 9131 85-27625
Fax: +49 9131 85-28809
E-Mail: philippsen@informatik.uni-erlangen.de

The **ParSeMiS** project (**Par**allel and **Se**quential Graph **Mi**ning **S**uite) searches for frequent, interesting substructures in graph databases. This task is becoming increasingly popular because science and commerce need to detect, store, and process complex relations in huge graph structures.

For huge data that cannot be worked on manually, algorithms are needed that detect interesting correlations. Since in general the problem is NP-hard and requires huge amounts of computation time and memory, parallel or specialized algorithms and heuristics are required that can perform the search within time boundaries and memory limits.

Our target is to provide an efficient and flexible tool for searching in arbitrary graph data, to improve the adaption to new application areas, and to simplify and unify the design of new mining algorithms.

In 2009, the following goals have been attacked:

- Optimizations for embedding-based frequency mining: A more detailed look at the pruning-related properties of the maximum-clique-test resulted in a huge runtime improvement, particularly for low frequencies that are of special interest for applications. This is achieved by an early detection during the NP-complete test that decides for a fragment whether can become frequent at all.

- Improved distribution for clusters of multi-core nodes: Co-location of threads in the same memory speeds up parallel search. First results have been seen in 2009, more are expected in 2010.

1.2.4 JavaParty

Project manager:
Prof. Dr. Michael Philippsen
Project participants:
Dipl.-Inf. Marc Wörlein
Start: 1.4.2007
Contact:
Prof. Dr. Michael Philippsen
Phone: +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 allows easy porting of multi-threaded Java programs to distributed environments such as e.g. clusters. Regular Java already supports parallel applications with threads and synchronization mechanisms. While multi-threaded Java programs are limited to a single address space, JavaParty extends the capabilities of Java to distributed computing environments.

The normal way of porting a parallel application to a distributed environment is the use of a communication library. Java's remote method invocation (RMI) renders the implementation of communication protocols unnecessary, but still leads to increased program complexity. The reasons for increased complexity are the limited RMI capabilities and the additional functionality that has to be implemented for creation and access of remote objects.

The JavaParty approach is different. JavaParty classes can be declared as remote. While regular Java classes are limited to a single virtual machine, remote classes and their instances are visible and accessible anywhere in the distributed JavaParty environment. As far as remote classes are concerned, the JavaParty environment can be viewed as a Java virtual machine that is distributed over several computers. Access and creation of remote classes are syntactically indistinguishable from regular Java classes.

In 2009, the JavaParty compiler was extended by a semantics for inner classes.

1.2.5 Jackal - Cluster and Grid computing made easy

Project manager:
Prof. Dr. Michael Philippsen
Project participants:
Ronald Veldema, Ph.D.
Start: 1.1.2006

Jackal is a project to create a distributed-shared-memory system for Java. This means that you can write a multi-threaded program (that you could run using normal JVMs on single machines as well) and deploy it on a cluster connected by a network. Jackal also sports a nice check-pointer so it can periodically write the program state to disk for fault tolerance.

To make things more interesting, you can write the program also using OpenMP annotations which allows re-parallelization. Combined with checkpointing this allows a program to be restarted on a different number of machines as that it was started with.

OGRE is a sub-project that makes the Jackal-DSM available for Grid-computing. OGRE is responsible for job submission, check-point management, and job migration between clusters.

1.2.6 Tapir

Project manager:
Prof. Dr. Michael Philippsen
Project participants:
Ronald Veldema, Ph.D.
Dr.-Ing. Michael Klemm
Duration: 1.1.2006–31.12.2010
Contact:
Ronald Veldema, Ph.D.
Phone: +49 9131 85-27622
Fax: +49 9131 85-28809
E-Mail: veldema@informatik.uni-erlangen.de

Tapir is a new programming language to ease systems programming.

Systems programming includes networking protocols, operating systems, middlewares, DSM systems, etc. Such systems are critical for the functioning of a system as they supply services that are required by user applications. For example, an operating system supplies an operating environment and abstracts from concrete hardware in doing so.

A DSM system simulates a single shared memory machine by abstracting from the single machines inside a cluster so that a (user-level application on a) cluster can be programmed without having to program explicit message passing.

Compared to application programming, systems programming has a different set of requirements. The programming 'style' is also very different from the styles used in programming user-level applications. Finally, the performance requirements are usually very strict in systems programs as the complete system's performance greatly relies on the underlying layers of systems programs. Bugs in systems code have also great repercussions on a complete system's reliability. Combined, we can directly conclude the following when using high-level languages for systems programming:

- High-level languages, such as C++, C#, and Java 'hide' implementation details from the programmer. A programmer for example no longer needs to know how exactly a method call is implemented. This knowledge is, however, required when doing systems programming.
- High-level languages supply functionality that is neither required nor wanted. For example, when programming an operating system, automatic language driven exception handling or garbage collection are not wanted.
- Systems programs require no high abstraction levels like common high-level programming languages supply. Likewise, the libraries that a given language offers can simply not be supplied in a systems context. Usually this is due to the system itself supplying the functionality that the library is to provide.

The basics of the Tapir language have been created. While Tapir has some similarities to Java, C#, and C++, all unneeded and unwanted functionality of the above have been removed. For example, Tapir has no automatic memory management, no exception handling, and no type-casts. Class and object concepts have been kept, but inheritance has been removed. The resulting Tapir programs can be verified by means of model-checking, even while the programming is still being developed. A prototype of the Tapir compiler and the Tapir model-checker is operational. While the language is still being developed, a prototype DSM protocol has already been implemented in the Tapir language. We have evaluated RDMA-based DSM protocols so that they can be added to the Tapir language. Tapir's semantic checks are implemented by means of model-checking. Model-checking, however, is a very memory intensive analysis. This made us write our own Java Virtual machine, called LVM, which is especially suited for managing large numbers of objects. LVM outperforms standard Java VMs as soon as swapping becomes necessary.

In 2009, the Tapir language was itself improved to allow both easier automatic program verification and to allow more efficient code to be generated. For example, the language has become easier to verify because code pieces that are implementation details can be marked as such and be safely ignored by the verifier. The efficiency of the language has been improved such that selected parts of programs can now be executed in parallel for example also on a graphics card without the possibility of the common programming errors associated with parallelism can occur.

1.2.7 OpenMP/Java

Project manager:
Prof. Dr. Michael Philippsen
Project participants:
Ronald Veldema, Ph.D.
Dr.-Ing. Michael Klemm
Dipl.-Inf. Georg Dotzler
Duration: 1.10.2009–1.10.2015

JaMP is an implementation of the well-known OpenMP standard adapted for Java. JaMP allows one to program, for example, a parallel for loop or a barrier without resorting to low-level thread programming. For example:

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

is valid JaMP code. JaMP currently supports all of OpenMP 2.0 with partial support for 3.0 features, e.g., the collapse clause.

The older (Jackal) JaMP version is part of the Jackal compiler. It allows transparent cluster computing and even allows migration of the application from one cluster of the Grif to another by means of the OGRE framework.

The newer JaMP version is independent of Jackal and generates pure Java 1.5 code that runs on every JVM. It also translates parallel for loops to CUDA-enabled graphics cards for extra speed gains. If a particular loop is not CUDA-able, it is translated to a threaded version that uses the cores of a typical multicore machine.

1.2.8 Model Driven Component Composition

Project manager:
Prof. Dr. Michael Philippsen
Project participants:
Dipl.-Inf. Philipp Janda
Duration: 15.6.2007–14.6.2010
Sponsored by:
AUDI AG

This project is part of the INI.FAU collaboration between AUDI AG and the University of Erlangen-Nuremberg. It examines model-driven ways to integrate vehicle functions on electronic control units (ECUs). Moreover, the project develops supporting methods and tools for this task. The insights gained in the course of this project will be practically validated by integrating a damper control system into an AUTOSAR ECU.

In the automotive industry it is common practice to develop in-car-software on a high level of abstraction and in a model-based way. To eliminate uncertainties concerning resource consumption and runtime it is necessary to test the developed software on the target hardware as early as possible. But due to cost and safety requirements the integration of the software into an ECU is very time-consuming and demands special expertise going beyond that of the function developer. AUTOSAR (AUTomotive Open Systen ARchitecture) is on the way to become a standard for the basic software on ECUs. But due to the novelty of this standard there are neither processes nor tools to support the integration of the developed in-car-software into an ECU.

In 2008, we have examined the modeling expressiveness of AUTOSAR with respect to both its applicability and possible conflicts with existing standards and technologies that are currently in use at Audi. Furthermore, the automatic generation of an AUTOSAR software architecture from a single damper control component has been implemented.

Since 2009, a model-driven approach that supports the integration of software into an ECU is being implemented and integrated into the tool set used at Audi. In particular we are looking at the automatic configuration of the bus communication by means of a bus database and the automatic task scheduling among the application processes. The model-driven development, which in this case is based on the Eclipse Modeling Framework, will enable easier tayloring of the emerging prototype to changing requirements.

1.2.9 Integrated Tool Chain for Meta-model-based Process Modelling and Execution

Project manager:
Hon.-Prof. Dr.-Ing. Detlef Kips

Project participants:
Dipl.-Inf. Ralf Ellner
Prof. Dr. Michael Philippsen
Dr.-Ing. Martin Jung
Dipl.-Inf. Johannes Drexler
Al-Hilank, Samir
Duration: 1.10.2008–30.9.2011
Sponsored by:
BMWi

As demands on the development of complex software systems are continuously increasing, compliance with well-defined software development processes becomes even more important. Especially large and globally distributed software development projects tend to require long-running and dynamically changeable processes spanning multiple organizations. In order to describe and support such processes, there is a strong need for suitable process modeling languages and for powerful support by tools.

The results of a preceding cooperation project show that today's tools markets lack integrated tool chains which actually support the fine-grained and precise modeling of software development processes as well as their computer-aided execution, controlling and monitoring. An ongoing cooperation project intends to bridge this gap.

The objective of this cooperation project is to prototype an integrated tool chain by using a rigorous, metamodel based approach which supports modeling, enactment and execution of industrial software development processes. Bearing the applicability of such a tool in mind, this approach is mainly intended to provide a wide adaptability of process models to different industrial development scenarios, to define a user-friendly concept of process description and to establish an extensive computer-aided process execution support, contributing to the efficiency of development activities. These benefits will be achieved by a high grade of formalism, by an integrated, generic concept of process modeling and process enactment and by using commonly accepted industrial standards (UML, SPEM).

This ongoing cooperation project is carried out together with develop group as an industrial partner and is funded by BMWi. It started in October 2008 and has been scheduled for three researchers over a period of three years.

Within the reporting period, first concrete results of this project could be achieved. SPEM has been analyzed and several problems of the standard have been fixed. This includes an extension of SPEM by certain behavior modeling concepts, which allow for specifying enactable, dynamically executable software process models. Our extended version of SPEM is called eSPEM (enactable SPEM) and is available through the chair's website. In order to evaluate eSPEM, a prototype of a process modeling environment

(PME) has been developed and is actually used to model some development processes which are widely used in industrial development teams. A prototype of a process execution environment, which is able to enact and dynamically execute eSPEM-based process models, is currently under development and will be the main subject of our research in 2010.

1.2.10 Softwareleitstand

Project manager:
Prof. Dr. Michael Philippsen
Project participants:
Dipl.-Inf. (FH) Josef Adersberger
M. Eng. Norbert Tausch, Dipl.-Ing. (FH)
Start: 1.11.2009
Sponsored by:
Bundesministerium für Wirtschaft und Technologie
Contact:
Prof. Dr. Michael Philippsen
Phone: +49 9131 85-27625
Fax: +49 9131 85-28809
E-Mail: philippsen@informatik.uni-erlangen.de

Prototypical implementation of a new tool for quality assurance during software development

Modern software systems are getting increasingly complex with respect to functional, technical and organizational aspects. Thus, the number of requirements per system and the degree of their interconnection constantly rise. Furthermore the technical parameters, e.g., for distribution and reliability are getting more complex and software is developed by teams that are not only spread around the globe and also suffer from increasing time pressure. Due to this, the functional, technical, and organizational control of software development projects is getting more difficult.

The "Softwareleitstand" is a tool for managing project roles like the project leader, the software architect, the requirements engineer, or the head of development. Its purpose is to make all aspects of the development process transparent and thus to allow for better project control. To achieve transparence, the tool distills and gathers properties from all artifacts and correlations between them. It presents/visualizes this information in a way suitable for the individual needs of the users.

The Softwareleitstand unifies the access to relations between artifacts (traceability) and to their properties (metrics) within software development projects. Thus, their efficiency can be significantly increased. The artifacts, their relations, and related metrics are gathered and integrated in a central data store. This data can be analyzed and visualized, metrics can be computed, and rules can be checked.

1.2.11 Wireless Localization

Project manager:
apl. Prof. Dr.-Ing. Gabriella Kókai
Project participants:
Dipl.-Ing. (FH) Thorsten Edelhäußer
Duration: 1.5.2008–30.4.2011
Sponsored by:
Fraunhofer Institut für Integrierte Schaltungen
Participating institutions:
Fraunhofer Institute für Integrierte Schaltungen

In 2009, we continued the development of our algorithms to estimate the receiving antenna's position (pose) of location systems. The algorithms estimate measuring points which allow a fast and accurate measurement pose. We applied an robot to execute the measurement automatically. The developed algorithm considers obstacles and the receiving characteristics of the location system and can sort out errors contained in the measurement data (e.g. multipath measurements).

Furthermore, algorithms for dynamic moving models are developed. Learning methods are applied to adapt models at run-time. A description language was developed to describe trajectories.

1.2.12 Evolutionary agents

Project manager:
Dipl.-Inf. Stephan Otto
Duration: 1.4.2008–31.5.2009
Contact:
Dipl.-Inf. Stephan Otto
Phone: +49 9131 85-27830
Fax: +49 9131 85-28809
E-Mail: Stephan.Otto@informatik.uni-erlangen.de

There is a trend towards complex and distributed systems, complicating the design process of self-adaptive and complex software systems. Examples are the internet, grid and peer-to-peer based systems and subsequent developments like multi agent systems. The growth of the World Wide Web and the rapid rise of eCommerce have led to significant efforts to develop distributed and complex software and technologies to support and enable the engineering of systems involving distributed computation. These effects originated from inherent distributed problems e.g. global supply chains or collaborative internet based applications. As a consequence this development has led to more decentrality which has a number of practical advantages over centralized approaches:

- Central elements are limited by how much they can communicate, store and process,
- Centralized solving may be infeasible due to privacy and data integration problems,
- Dynamic environment: by the time we manage to centralize the problem, it has already changed.

In this context of dynamic, inaccessible and distributed structures the adaptation and optimization of systems and processes still poses problems. Current adaptation or optimization techniques rely on central processing. There exist a number of approaches to tackle problems bottom-up but they require cooperative and predefined behaviour of involved actors and such methods work mostly on a limited set of problems. Therefore in current adaptation and optimization methods exist a number of problem areas which need more investigation: distribution, heterogeneity and dynamic environments.

This project focuses on a tolerant approach towards the previously mentioned aspects of investigation. Whilst the top-down approach tackles problems by programming intelligence from above, the bottom-up approach tries to build structures as a result of specifying a set of simple rules, a set of simple elements, called agents, which adhere to those rules. A main characteristic of complex systems is their bottom-up approach on adaptation and self-organization without following central directions. This project aims to develop an approach that provides adapting and self-optimizing decentralized applications using a multi-agent system (MAS) based on Evolutionary Computation (EC). Our approach uses fully decentralized operators for reproduction like mutation, recombination and selection, regulated by market mechanisms. This tackles the existing bottleneck of selection and fitness comparing in current distributed evolutionary approaches. The novelty of this approach lies in the decentralized bottom-up adaptation and optimization method for decentralized systems and is applied to various scenarios. Our proposed method is based on a formal model explaining how adaptation occurs in the number and strategies of agents and thus of the emergent adaptation and optimization itself.

The key contribution of the present project is summarized as follows:

- Development of a new distributed evolutionary algorithm. Central selection and fitness comparison are omitted,
- Development of endogenous fitness and their effect on the quality of solutions,
- Development and comparison of local selection methods,
- To put the results on a solid basis, investigations on the takeover time on weakly connected graphs have been conducted to evince appropriate information flow in weakly connected, distributed and dynamic structures,
- A new measure of decentrality has been developed to help classify distributed approaches,
- Several scenarios were successfully tested using this approach.

The performance of the approach developed in this project has been shown as part of a dissertation. The dissertation was submitted in spring 2009 and successfully defended in September.

1.2.13 Optimization of FIR filter structures

Project manager:
apl. Prof. Dr.-Ing. Gabriella Kókai
Project participants:
Dr.-Ing. Szilvia Zvada
Dipl.-Ing. Hans Holm Frühauf
Duration: 1.1.2006–30.9.2009
Contact:
apl. Prof. Dr.-Ing. Gabriella Kókai
Phone: +49 9131 85-28996
Fax: +49 9131 85-28809
E-Mail: kokai@informatik.uni-erlangen.de

Due to the boom of electronic systems in everyday life, VLSI (very large scale integration) circuits quickly became the hot-spot of intensive research. The main issue in this area is to design chips that are small, fast, and consume little power. The advancement of modern chip fabrication technology and the increasing packing density have made it possible that today's VLSI chips contain a few million transistors. From a chip designer's point of view this unfolds a vast number of possible filter structures while looking for an optimal or near-optimal chip. Thus, the automation of the design process becomes increasingly important.

In case of digital filters much attention has been paid to the design of the FIR (finite impulse response) filters. These filters are widely used to transform digital data sequences according to some specific linear function, e.g. the linearization of power amplifiers or the calibration of audio or video receivers. If, however, a non-linear transformation of the data sequences is required, a great part of the design process must be done manually.

The evolFIR system developed in this project fills this gap by providing a sophisticated tool that can optimize the logic design of polynomial FIR filter structures. At this stage of the circuit design, functional block elements (adders and multipliers) and logical primitives (shifts and delays) are composed in order to fulfill the required functionality of the target filter. This functionality is specified by the filter's transfer function. During the evolutionary process, we firstly have to ensure that the individuals always describe the desired transfer function. Secondly, we have to assure that the composed topology fulfills certain hardware-related requirements known from the subsequent levels of chip design, such as the limited number of input channels or the restricted word-width of the dedicated block element.

The main objective of evolFIR is to obtain a redundancy-free filter design with as few block elements as possible. We achieve this by employing AGGP (attribute grammar based genetic programming) as follows:

- The individuals of the evolutionary process are special derivation trees that represent possible topologies of the available functional elements. Since we allow the usage of multipliers in the compositions, polynomial transfer functions can also be dealt with.

- By means of attributes and the smart random tree generator of AGGP, exactly those derivation trees are surveyed during the optimization that represent filter descriptions of the given transfer function.

- Redundant components of the filter descriptions are eliminated by using a unique representation form of the derivation trees: the abstract-linked derivation trees. Hence, only the redundancy-free filters are included in the search space.

- In addition, the random tree generator is enhanced to take the hardware-related constraints into consideration: it only creates such derivation trees that fulfill these constraints.

Altogether, the evolFIR method operates on an inhomogeneous search space due to these various restrictions. The inhomogeneity of the search space and our particular representation form imply that the fitness function is non-continuous. Therefore, the evolutionary engine requires careful parameterization in order to avoid premature convergence of the evolutionary process.

The performance of the approach developed in this project has been shown as part of a dissertation. The dissertation was submitted in winter 2008 and successfully defended in June.

1.2.14 Graphs and Graph Transformations

Project manager:
Prof. em. Dr. Hans Jürgen Schneider
Duration: 1.10.2004–30.9.2010
Contact:
Prof. em. Dr. Hans Jürgen Schneider
Phone: +49 9131 85-27620
Fax: +49 9131 85-28809
E-Mail: schneider@informatik.uni-erlangen.de

Graphs are often used as an intuitive aid for the clarification of complex matters. Outside the field of computer science this is for example true in biology or chemistry, where molecules are modeled in a graphical way. In computer science data or control flow charts are often used as well as entity relationship charts or Petri-nets to visualize software or hardware architectures. Graph grammars and graph transformations combine ideas from the fields of graph theory, algebra, logic, and category theory, to formally describe changes in graphs.

The underlying theory is an attractive tool for the description of extremely different structures in a uniform way, e.g., the different models for asynchronous processes: Petri-Nets are based on standard labeled graphs, state charts use hierarchical graphs, parallel logic programming can be interpreted in a graph-theoretical way using so-called jungles, and the actor systems can be visualized as graphs, whose labeling alphabet is a set of term graphs.

In 2009, we have concentrated our attention on a theoretical concept as well as on an implementation aspect:

- In the last years, we have generalized the concept of independent derivation steps in such a way that it can be applied to any category of interest, especially to the category of structurally labeled graphs. We have now completed the discussion of transforming derivation sequences by considering parallel productions, i.e., productions that combine the effects of two or more productions in one step. It is easy to define a parallel production as the coproduct of several productions, but in the case of graphs, this leads to derivation steps with non-injective handles. Contrary to some other known approaches, our discussion is not restricted to injective handles, and the construction is sufficient. On the basis of parallel derivation steps, one can also define canonical derivations. In the theory of Chomsky-languages, this definition takes advantage of the left-to-right order of the symbols, which is not available in the case of graphs. In 1976, Kreowski has proposed an alternative: Independent derivation steps are made parallel and the components are shifted to the first possible position. A canonical derivation is one that does no longer allow any shift. Kreowski's definition can be immediately applied to our general setting that is not restricted to graphs. The details can be found in the draft of our textbook: http://www2.informatik.uni-erlangen.de/Personen/schneide/gtbook/chapter5.pdf

- The categorical approach to graph transformations is highly generic: All the proofs and constructions are valid for various types of graphs. Only the basic operations must be described for each application in detail, the categorical properties on top of these are then defined automatically. Since modern programming languages support generic concepts, it is a promising idea to implement the categorical approach to graph transformation in languages such as Java or Haskell. Java uses classes of objects, but does not really support multiple inheritance. In our first version, we define interfaces Cat, CatWithColimits, etc. which must be implemented by classes such as CatFinSet, CatFinGraph, etc. The basic colimits (coproduct, coequalizer, initial object) are introduced as abstract classes the details of which must be implemented by the special categories. Since these colimits allow to construct all other colimits, it would be a straightforward solution to provide CatWithColimits with a method constructing pushouts, but CatWithColimits is an interface. Therefore, we introduce a class PushoutCreator describing this categorical construction independent of a special category. Each category with colimits must import this factory class. A summary of this implementation is available: http://www2.informatik.uni-erlangen.de/Personen/schneide/gtbook/appendix-b.pdf

1.2.15 Time-Tabling Algorithms

Project manager:
PD Dr.-Ing. habil. Peter Wilke
Project participants:
Dipl.-Inf. Johannes Ostler
Duration: 1.1.2004–1.10.2009
Contact:

PD Dr.-Ing. habil. Peter Wilke
Phone: +49 9131 85 27825
Fax: +49 9131 303811
E-Mail: wilke@informatik.uni-erlangen.de

Time tables have to be generated in quite different areas, e.g. human ressources, school time tabling etc. Because scheduling is a time consuming task, especially if the context is complex, most time tables are generated by computer programs. We have developed a sophisticated software which enables us to generate optimised time tables using different optimisation algorithms in recent years. The current software is a complete new implementation as the former version of the software showed some design flaws which persuaded us to re-design the architecture.

Erlangen Advanced Time Tabling Software EATTS is the innovative development and production environment to generate optimized time tables.

Resources

Time tabling problems are quite common and come in different versions, among them rosters, schedules and school time tables. They have in common that given resources have to be used as efficient as possible and that this requires planning with respect to the given constraints to obtain a decent plan. When looking at a school time table the events are lessons, to which the resources like teachers, classes ans rooms have to be assigned. All resources are typed. Each type has as many attributes created by the user as required.

Planning a time table usually begins with compiling the resources, either by reading in a file or typing in them manually. The screen shot shows the input dialogue to enter the attribute values for a class. All resources of the type ”Class” have to be assigned values for the user defined attributes ”name”, ”size”, ”grade” and ”room”.

Results

Time tables are the output of the planning algorithms and can be stored in different file formats and views. The screen shot shows the view of a student, e.g. he sees his personal time table consisting of the lessons he has to attend to. Other views can be created instantly, for a teacher, a headmaster or a caretaker. All users access the EATTS through a browser providing an interface according to the privileges of the user.

A common view indicates which constraints are currently not satisfied naming the events and resources. Based on this information the administrator can decide if he would like to edit the resources, events or constraints or use a different algorithm. The screen shot show time slot clashes and therefore the algorithm should be given more time to find a solution a another algorithm should be given a shot.

Constraints

The specification of constraints is usually more complex than describing resources or events.

On one hand a precise specification is required and one the other hand the current setting should be presented clearly to find gaps and/or inconsistencies, which can't be avoided automatically.

Constraints come in different flavours, therefore a flexible way to specify them is necessary. EATTAS allows to refer to resources, their classes and all attributes. Depending on the type of the attributes, among them are integer, double and string, arithmetic and logical operators can be used to specify the constraint. In addition the parameters of the cost function are set, to compute the correct penalty point if the constraint is violated.

A unique property of EATTS is the option to specify a constraint not only as ”has to be fulfilled (hard)” or ”should be fulfilled (soft)” but also as ”can be violated in exceptional case (soft hard)”. This allows to violate constraints when it is acceptable, e.g. a room isn't available due to a broken water pipe, a teacher isn't available due to a traffic jam. In these cases a time table should be created which is similar to the one currently in effect but violates some constraints to minimize changes.

Running Experiments

The algorithms can be executed on a dedicated server or can be distributed over a TCP/IP network on additional computers. The screen shot shows the dialogue where the user can select the experiments and start their execution. The browser contacts the server regularly and updates the status information, including the costs of the best plan found so far and an estimated remaining computation time. At the end of the computation results are stored and the data required for the views are generated. Now the user decided whether the results are satisfactory or additional computations are required.

Results

Time tables are the result of the application of algorithms to the specifies resources, events and constraints. Time tables can be stored and displayed in various formats, enabling the display of different views of the time table.

Users typically access the EATTS via a browser and views are created depending on the users privileges.

Summary

The software is implemented in Java and available for numerous platforms, among them Windows and Linux operating systems.

To run EATTS the following free-ware software products are required:

- Java Runtime Environment (JRE), v5.0 are above,

- computers connected via TCP/IP, if additional computing power is required (optional).

In 2008 the structure of the algorithms was optimized to enhance their parallel computation. We will look into the use of multi core processor in the near future.

As attempts to install the software by users have shown that this might be too complex a downsized version which does not need a database but uses XML documents to store and exchange data was implemented. In addition a server was set up running on a computer at the University of Erlangen allowing the users to run their experiments on this mashine.

The user interface was reimplemented and is now a web based application.

At CeBIT 2009 the new version of the software has been presented. And it was named EATTS Erlangen Advanced Time tabling System.

1.2.16 International Collegiate Programming Contest at the FAU

Project manager:
Prof. Dr. Michael Philippsen
Project participants:
Dipl.-Inf. Tobias Werth
Dipl.-Inf. Marc Wörlein
Dipl.-Inf. Alexander Dreweke, B. Sc. mult.
Start: 1.11.2002
Contact:
Dipl.-Inf. Tobias Werth
Phone: +49 9131 85-28865
Fax: +49 9131 85-28809
E-Mail: tobias.werth@informatik.uni-erlangen.de

The Association for Computing Machinery (ACM) has been hosting the International Collegiate Programming Contest (ICPC) for many years. Teams of three students try to solve nine to ten programming problems within five hours. What makes this task even harder, is that there is only one computer available per team. The problems demand for solid knowledge of algorithms from all areas of computer science and mathematics, e.g. graphs, combinatorics, strings, algebra and geometry.

The ICPC consists of three rounds. First, each participating university hosts a local contest to find the three teams who afterwards compete in one of the various regional contests. Erlangen lies in the catchment area of the Northwestern European Regional Contest (NWERC) where also teams from e.g. Great Britain, Benelux and Scandinavia participate. The winners of all regionals in the world (and some second place holders) advance to the world finals in spring of the following year.

In 2009 two local contests took place in Erlangen. During the winter semester a team contest was conducted with teams consisting of at most three students. The main goal of this contest was to interest new students for the contests. Also, some teams from TU Munich and University of Konstanz competed with our teams online.

As in previous years in the summer term the seminar ”Hello World - Programming for Advancers” served to prepare students for the contests. In the contest of the summer semester the representatives of the FAU Erlangen-Nuremberg for the NWERC 2009 in Nuremberg were chosen. 25 students of computer science, computational engineering, mathematics as well as information and communication technology took part in the contest. Ten students were selected for the NWERC forming three teams and one stand-by. At the NWERC in Nuremberg (hosted by the FAU!), our best team won clearly against the competitors from Northwestern Europe. Therefore, with the ticket to the World Finals that take place in Harbin/China, the training-camp was also in 2009 a great success.

The regional contest (NWERC) was hosted by the FAU with 64 teams in Nuremberg. It was a great success and a fun weekend for all the teams from different countries and universities.

1.2.17 Embedded Systems Institute

Project manager:
Prof. Dr. Michael Philippsen
Project participants:
Dipl.-Inf. Philipp Janda
Dipl.-Inf. Stefan Kempf
Dipl.-Inf. Georg Dotzler
Start: 1.9.2007

In September 2007 the ESI – Embedded System Institute – was founded as an interdisciplinary center at the Friedrich-Alexander-University (FAU) with the goal to coordinate and organize research, teaching, and further education in the field of embedded systems.

ESI brings together existing skills within the university and interests, activities, and goals of large and medium size companies in the field of embedded systems.

Companies obtain access to latest research results and the opportunity to develop common projects, to establish ties, and to find co-operation partners. The ESI concentrates the skills of the chairs of computer sciences and makes them usable for co-operation projects. Hence, the latest research results can be transfered into products in a speedy way. Finally, the ESI may serve as a platform for recruiting excellent students and highly qualified young academics at an early stage.

The chair of computer science department 2 (Prof. Philippsen) is one of the active founders of the ESI and carries out research projects within the ESI.

1.3 Projektunabhängige Publications

- 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), No. 14, pp 1789-1803

- Dreweke, Alexander ; Fischer, Ingrid ; Werth, Tobias ; Wörlein, Marc: Text Mining in Program Code . In: Song, Min ; Wu, Yi-Fang Brook (Ed.) : Handbook of Research on Text and Web Mining Technologies. Hershey : Idea Group Publishing, 2009, pp 626-645. - ISBN 1599049902

- Edelhäußer, Thorsten ; Kókai, Gabriella: Autonomous Configuration Method for Real-Time Location Systems . In: Suess, Martin ; Arslan, Tughrul (Ed.) : Procedings 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, pp 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), No. 2, pp 130-140

- Grimm, Robert ; Edelhäußer, Thorsten ; Kókai, Gabriella: An Efficient Configuration Method for RTLSs . In: Jacak, Witold ; Affenzeller, Michael (Ed.) : Proceeding 2nd International Symposium on Logistics and Industrial Informatics (2nd International Symposium on Logistics and Industrial Informatics Linz Austria September 10 and 11). 2009, pp 148-153.

- Kempf, Stefan: A Language Independent JIT compiler library . Erlangen-Nürnberg, Friedrich-Alexander-Universität, Dipl. thesis, 2009. - 116 pages.

- Klemm, Michael: Reparallelization and Migration of OpenMP Applications in Grid Environments . Aachen : Shaker Verlag, 2009. Zugl.: Erlangen, Friedrich-Alexander-Universität Erlangen-Nürnberg, Ph.D. thesis, 2009. - 272 pages. 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), No. 3, pp 281-299

- Otto, Stephan: Ein agentenbasierter evolutionärer Adaptions- und Optimierungsansatz für verteilte Systeme . Erlangen, Friedrich-Alexander-Universität Erlangen-Nürnberg, Ph.D. thesis, 2009. - 209 pages.

- Schell, Dominic: Dynamische Programm-Code-Verwaltung und -Optimierung für eingebettete Systeme . Erlangen, Friedrich-Alexander-Universität Erlangen-Nürnberg, Ph.D. thesis, 2009. - 189 pages.

- Schmidt, Karsten ; Janda, Philipp: Erfahrungen bei der modellbasierten Entwicklungvon 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 (Ed.) : Informatik 2009 - Im Focus Leben (ATPS 2009 - 4. Arbeitstagung Programmiersprachen Lübeck 01.10.2009). Bonn : GI Gesellschaft für Informatik, 2009, pp 364. (Lecture Notes Informatics Vol. 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. (Ed.) : Data Mining for Business Applications. Berlin/Heidelberg : Springer-Verlag, 2009, pp 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 (Ed.) : Proc. ICSE Workshop on Multicore Software Engineering (IWMSE'09 Vancouver, Canada 18.05.2009). New York, NY : IEEE, 2009, pp 64-72. - ISBN 978-1-4244-3718-4

1.4 Exam theses (german only)

- 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

- 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: 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

- 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: Entwurf und Implementierung eines Konzepts für die Formulierung von Beschränkungen. Bearbeiter: Eugen Kremer (beendet am 15.11.2009); Betreuer: PD Dr.-Ing. habil. Peter Wilke