Network of Excellence

IST-004527 ARTIST2:
Embedded Systems Design

Cluster Progress Report for Year 2

Cluster:
Compilers and Timing Analysis

Cluster Leaders:

Prof. Dr. Reinhard Wilhelm (Saarland University)
http://rw4.cs.uni-sb.de/users/wilhelm/wilhelm.html

Prof. Dr. Rainer Leupers (RWTH Aachen)
http://www.iss.rwth-aachen.de

Policy Objective (abstract)

Compilation tools and their associated technologies play a fundamental role for the implementation of a system on a given target platform. For embedded systems, we need tools capable of combining platform independent software and a description of the target platform, to generate an executable code having the desired properties related to use of such resources as memory, power, energy, network bandwidth, and computation time. The main objective of the compilers cluster is to strengthen the European research community in the area of compilers for embedded processors by two major activities that utilize the excellence areas of the cluster partners: (1) a common compiler platform and (2) joint research in selected areas of architecture aware compilation and code optimization. We also need tools to verify that the executable code has these properties, including tools to estimate the execution times of embedded software on a given platform. Further, coupling timing analysis tools with compilation tools enables resource-aware compilation.

Table of Contents

1. Overview

Run-time guarantees play an important role in the area of embedded systems and especially hard real-time systems. These systems are typically subject to stringent timing constraints which often result from the interaction with the surrounding physical environment. It is essential that the computations are completed within their associated time bounds; otherwise severe damages may result. Therefore, a schedulability analysis has to be performed which guarantees that all timing constraints will be met (also called timing validation). All existing techniques for schedulability analysis require the worst case execution time (WCET) of each task in the system to be known before the task is executed. Since in general, the problem of computing WCETs is not decidable, estimations of the WCET have to be calculated. These estimations have to be safe, i.e., they may never underestimate the real execution time. Furthermore, they should be tight, i.e., the overestimation should be as small as possible.

In modern microprocessor architectures caches and pipelines are key features for improving performance. Unfortunately, they make the analysis of the execution behaviour of instructions very difficult since this behaviour now depends on the execution history. Therefore, the classical approaches to worst case execution time prediction are not directly applicable or lead to results exceeding the real execution time by orders of magnitude. This may influence the degree of success of timing validations or may lead to a waste of hardware resources and more expensive hardware. For products which are manufactured in high quantity, e.g., in the automobile or telecommunications markets this possibly results in high expenses.

We cite and comment the main high-level objectives from the last 18-months work program here:

"Provision of advice and extensions to CoSy in its role as a platform technology for the cluster. Over the next year we expect more researchers will be using CoSy and see ACE providing workshops, lectures, fielding more requests for platform extensions and having more interaction within the cluster by way of whiteboard design discussions and assistance with the preparation of academic papers. […] Due to the central role of ACE in the cluster, it has been decided to include ACE as a full ARTIST2 partner after project year 1. Simultaneously, the cluster members will be open to further affiliate members with complementary areas of excellence related to the compiler platform activity."

ACE has been successfully integrated as a core member into the compilers cluster. The CoSy platform is being adopted or used by most cluster partners for R&D and/or teaching activities. One new affiliated partner (Prof. Sabine Glesner from TU Berlin) has joined the cluster to perform platform-based work on a complementary research topic (compiler verification). ACE provides extensive support for the partners using CoSy, e.g. a CoSy Users Group meeting will take place in Oct 2006. Further groups inside (e.g. U Bologna) and outside (e.g. TU Karlsruhe, U Edinburgh, TU Dresden) ARTIST2 have adopted the proposed compiler platform. Links to other communities (e.g. compiler researchers in the HiPEAC NoE) have been strengthened.

"One key objective for the forthcoming ARTIST2 period [in the architecture aware compilation activity] is the continuation of the successful mini-cluster level cooperations. Furthermore, a tighter integration between the architecture aware compilation and compiler platform activities is envisioned […]. Moreover, different activities outside of the compilers cluster will be pursued, e.g. inter-cluster cooperation, dissemination, as well as university-industry cooperations originating from ARTIST2 results."

The cooperation has successfully continued at the mini-cluster level, accompanied by global cluster meetings. Most activities in the architecture aware compilation activity now build on the proposed compiler platform. Inter-cluster cooperation has been intensified (e.g. between Bologna and Aachen), as well as university-industry cooperations (e.g. ACE-Aachen). See

technical descriptions of the two major cluster activities (compilers platform and architecture aware compilation) for further details.

TU Vienna's objective was to establish the necessary infrastructure for specifying program analyses with AbsInt's Program Analysis Generator (PAG) for C++, implement an alias analysis, and generate an external format of the computed aliasing information. This goal has been achieved by integrating PAG into the C++ infrastructure ROSE as component of the compiler platform. In year two all C++ language features (additional to those present in C) have been addressed, only excluding exceptions. By using PAG a shape analysis for C++ has been specified. As external text format the input format to AiSee was chosen which permits visualizing the computed aliasing information. The goal is to build a tool for whole-program analysis, based on this infrastructure. An initial prototype, WHOPA, has been implemented that permits performing high-level analysis of generic C++ applications and evaluating the impact of optimizations.

The group at Dortmund is focusing on the efficiency of embedded software. Most of the efforts are dedicated towards the efficient use of memories, as the speed gap between processors and memories widens and future systems' speed could well be dominated by the speed of memories. Also, memories consume a major percentage of the energy of embedded systems. Traditional memory hierarchies have not been designed for energy efficiency or timing predictability. Therefore, a new look at memory architectures was the key initial objective. This objective was maintained during the first two years of the project, apart from the usual refinement. Optimizers were designed which enable compile-time optimization for scratch pad memories. The first results that were achieved look very promising: scratch pads provide energy efficiency and timing predictability at an unprecedented degree. In a slightly more general context, the group used other source-level transformations for improving the efficiency of embedded software.

This cluster is somewhat artificially composed of the two areas Compilation and Timing Analysis. The reasons for this combination were bureaucratic in nature, i.e. restriction on the number of clusters in the proposal. There are some overlaps in the research areas. The emphasis in the activities is on the two platforms and on one attempt to combine both areas. The contributions in this cluster document are therefore either split into a **Timing-Analysis** and a **Compilation** part or given in the reports about these activities.

1.1 High-Level Objectives

The objectives on the **Compilation** side are

- to provide world-class code-synthesis and compiler tools for the generation of efficient machine code, including the integration of existing compiler-generation approaches such that compilers for new architectures can be built quickly, efficiently and reliably.

The objectives on the **Timing-Analysis** side are:

- to achieve a common representation for an intermediate exchange format for the various timing analysis components from the different partners.
- to achieve an integration of different modules from different partners for timing analysis..

Synergy between Compilation and Timing Analysis: Tools for estimating WCET and compilers have traditionally evolved as independent classes of tools. This results in information lacking in both kinds of tools which, however, is available in the other set of tools. For example, flow facts are computed by compilers but frequently not available to the same extent by timing analysis tools. Also, timing information is available in timing analysis tools but almost never available in compilers. Traditional approaches for adding the missing information result in a duplication of efforts: regeneration of flow facts in timing analysis tools and adding timing as a cost function in compilers. This duplication of efforts is to be avoided.

1.2 Industrial Sectors

Timing Analysis side:

Timing-Analysis tools have recently entered industrial practice and are in routine use in the aeronautics and automotive industry. Precision of the results is good, efficiency is tolerable, and usability needs improvement.

All sectors concerned with Embedded Systems need Compilation Technology, WCET estimations are relevant for all industrial sections using hard real-time systems. Therefore, industrial sectors in this case include avionics, automotive, defence and some areas where control systems are applied. Especially in the automotive and the aeronautical domains, there is a need to have precise knowledge on the worst-case timing behaviour of safety critical software. This need is underlined by the fact that the worst-case timing of large parts of the software used within the new Airbus A380 has been analyzed using AbsInt's aiT.

Both the aeronautics and the automotive industries follow a similar trend to integrated architectures, aeronautics to IMA (Integrated Modular Avionics), automotive to a component architecture developed by the AUTOSAR consortium. This transition at latest will require timing analysis as an integrated component in the development process.

However, only the availability of precise timing analyses does not fulfil industrial needs. Since the code of safety critical applications is typically generated by a compiler, the compiler should also be aware of worst-case timings. Currently, this is not the case leading to the unacceptable situation that, whenever it is detected that an application does not meet its real-time constraints, manual code transformation, recompilation and timing analysis need to be done repeatedly. The burden of timing analysis and optimization will be taken away from the human designer by the approach proposed at Dortmund.

Compilation side:

Mainly: Embedded software, semiconductor and system houses

Specifically: audio processing, video processing and data streaming applications in the TV, Set Top box, DVD player and recorder, mobile, base stations, printer and disk drive markets.

Efficiency of embedded software, in particular the efficiency of memories, is relevant for high-speed embedded systems. This includes multimedia and network applications. It is expected that most mobile devices will provide some kind of multimedia processing. Currently, most of the handheld devices feature processors with on chip memories. However, none of the industrial compilers support the automatic utilization of the on chip scratchpad memories. Moreover, no industrial tool exists for performing architectural level-exploration. Currently, all decisions regarding the optimal size of the on chip memories are taken by the system design through ad-hoc methods. Relevant industrial sectors include consumer electronics and telecommunications.

1.3 Main Research Trends

Standard tool architecture for WCET analysis has evolved. In a first phase, which may itself consist of several subphases, the software is analyzed to determine invariants about the sets of execution states at each instruction. Abstract interpretation is mostly used for this phase. The invariants allow the prediction of conservative execution times for individual instructions and for basic blocks. A second phase determines a worst-case path through the program. This is often done by implicit path enumeration; the control flow is translated into an integer linear program and then solved.

The results are more precise if a strong analysis of the control flow is performed. The compiler that has translated the source program into the executable often has valuable information about the control flow. Making this information available is a promising avenue.
The construction of timing-analysis tools is difficult, tedious, and error-prone. Research is on the way to develop computer support for this task.
For non-hard real time tasks, measurement-based methods are being evaluated and tested.

Embedded system software –if compared to hardware- usually involves a significant overhead in terms of energy consumption and execution time. However, for flexibility reasons, hardware cannot be used in applications with changing requirements. In order to make a software implementation feasible, efficiency of embedded software is a must. Various approaches for achieving this efficiency have been explored.

Due to the efficiency requirements, using power-hungry, high-performance off-the-shelf processors from desktop computers is infeasible for many applications. Therefore, the use of customized processors is becoming more common. These processors are optimized for a certain application domain or a specific application. As a consequence, hundreds of different domain or even application specific programmable processors (ASIPs) have appeared in the semiconductor market, and this trend is expected to continue. Prominent examples include low-cost/low-energy microcontrollers (e.g. for wireless sensor networks), number-crunching digital signal processors (e.g. for audio and video codecs), as well as network processors (e.g. for internet traffic management). Source-to-source level transformations are another approach for improving the efficiency of embedded software. These transformations are applied before any compiler is started. To some extent, these transformations are independent of the final processor architecture. Therefore, the advantage of this approach is that it can be used with almost any compiler. It can also be used in combination with retargetable compilation.

A third approach is the use of sophisticated optimizations within a compiler. Optimizations tuned towards embedded systems have been designed by a number of members of the ARTIST2 compiler cluster.

Due to the increasing importance of the memory interface, various optimizations have been designed that help to maximize the efficiency of the memory interface. These optimizations can be either integration into compilers or used as source-to-source level optimizations.

Requirements of most embedded systems comprise not only optimal resource usage but also high safety and dependability guaranties. To meet safety requirements it is necessary to develop software in higher programming languages and to ensure that the transformation process to machine code preserves the semantics of the program and consequently the software system's behaviour. To achieve very efficient machine code, compilers for embedded systems apply aggressive optimizations. The need to verify the compilation process is increased by the fact that such optimizations are very error-prone, in particular if they change the structure of programs. To make optimizations applicable in safety-critical systems and to ensure that efficient and also correct executable code is produced by the compiler, methods for the verification of optimizing compilers are explored.

The research trends are described in the activities reports in more details.

2. State of the Integration in Europe

The state of integration in the area of **Timing Analysis** is described in the *Timing-Analysis Platform Report*

There is an additional integration going on between Compilation and Timing Analysis, namely in the Activity WCET-aware Compilation. This activity is one of the first worldwide to take hard real-time requirements of systems into account in the compiler.

Within the ARTIST2 project, *RWTH Aachen* focuses on retargetable compilers and code optimization for embedded processors. RWTH Aachen closely cooperates with ACE and CoWare Inc. on the LISATek Compiler Designer product, based on the CoSy compiler system that is also used as the primary compiler platform within ARTIST2.

Dortmund University: memory-aware code optimization and source-level code optimization. Memory-architecture aware compilation techniques have been proposed by a limited set of people. The Architecture and Compilers for Embedded Systems (ACES) group of Prof. Nikil Dutt at University of California, Irvine and the group of Prof. Rajeev Barua at University of Maryland at College Park focus on memory aware compilation and optimization issues. The optimization techniques by both groups consider only data variables for optimizations, whereas the techniques proposed by University of Dortmund consider both variables and program code segments for optimization. Furthermore IMEC, an affiliated partner, has analysed allocation techniques for memory hierarchies intensively. Case studies were a key vehicle for this analysis. The group of Kandemir at Penn State University has published numerous papers on the issue. In addition, some work performed in the high-performance computing can be applied to memory-architecture aware compilation for embedded systems as well.

STMicroelectronics: embedded processor design, tools adaptation

Absint works on timing analysis and stack usage analysis tools, the program analyzer generator PAG, post pass code optimization and compilation (e.g. for Java). Within Artist 2, the focus is on timing analysis and PAG.

IMEC works on many aspects of System-on-Chip (SoC) design technology, including novel architectural templates, design methods and design tools. The target domains are especially focusing on embedded systems like embedded multi-media and communications systems Within the ARTIST2 project, the main focus lies on source-level code optimisation for data-dominated applications, with special attention for the data memory organisation related aspects.

ACE provides and supports the CoSy compiler development system as a platform technology with Aachen, STMicroelectronics, Philips, CoWare, Edinburgh and others being part of the CoSy eco-system. ARTIST2 is generating additional opportunities for cooperation and the sharing of results as ARTIST2 related activities progress with requests for access to CoSy coming from outside the immediate cluster.

TU Vienna works on optimization techniques for using high-level abstractions in programming embedded systems. It is paramount for the next decade that these abstractions can be optimized such that they can be used in embedded systems without a significant performance impact. This effort focuses on building and integrating existing infrastructures such that we can offer platforms for evaluating the impact of the precision of program analyses on today's languages performance that are used for embedded software. Within ARTIST2 the main focus is on integrating tools that allow a high-level specification of program analyses and performing library aware optimizations as source-to-source transformations. The targeted applications are real-world C and C++ applications making use of the full range of all language features.

TU Berlin works on the verification as well as on the development of optimizing compiler transformations and machine code generation. Especially in safety-critical applications in the embedded domain, compiler transformations must be both optimizing and correct. Hence, verification is necessary to ensure that transformations indeed preserve program semantics during compilation. Within ARTIST2, the focus is on the development of automated checkers that, for a particular compiler run with its source and target program, make sure that both programs are indeed semantically equivalent. As a starting point, the verification and development of checkers for loop transformations based on unimodular transformations is investigated.

Besides ARTIST2, a significant European compiler research community is active e.g. in the HiPEAC NoE, including INRIA and U Edinburgh. While the research activities of HiPEAC are partially complementary to ARTIST2 (e.g. HiPEAC is primarily exploring gcc as a compiler platform), many interactions are taking place, e.g. Rainer Leupers from Aachen participated as a lecturer in the 2006 HiPEAC summer school ACACES.

2.1 Other Research Teams

Essentially three more research teams have competed with Europe in the area of Timing Analysis, one at Seoul National University, one at Florida State University, now with some branches in North Carolina etc., and Singapore National University. Seoul has turned to power-aware computing, and flash memory based components research. Singapore and Florida have cooperated with the ARTIST2 partners in writing the survey paper. We hope that these groups will participate in the WCET Tool Challenge with their academic prototypes.

There is a continuous exchange of PhD students and PostDocs with the team of Prof. Sang Lyul Min at the Seoul National University.

Only few groups have working on the integration of worst case execution times and compilers. Smaller, apparently volatile efforts have been reported from Sweden and South Korea. More sustained work was performed in the group of David Whalley at the Computer Science Department of the Florida State University. Similar to the design of a WCET-aware compiler within this cluster, they have integrated WCET analysis and compilation techniques. However, due to the lacking cooperation between research groups, only very simplified timing models and highly predictable processor architectures are considered at Florida

2.2 Interaction of the Cluster with Other Communities

Timing-Analysis activities in the cluster interact closely with the Execution-Platform cluster in the area of increasing the timing-predictability of real-time systems. This issue is of high interest to several industrial sectors. Representative companies from these sectors supported a proposal that was submitted to FET Open. The topic was to reconcile predictability with performance. Both short and full proposal passed all thresholds. However, only 5% of the projects could be funded. Most likely, our proposal will not be among them.

An exchange has been initiated with the group of Prof. Min at *Seoul National University* to utilize the parametric WCET analysis work done at Saarland University in the analysis of embedded systems based on OS controlled flash memory based disk devices. The goal is to make the ARTIST2 results known and to apply them in a practical and finally industrial setting for real devices.

For the researchers at *Dortmund University*, the interaction with the local technology transfer centre ICD is a key for interacting with industry. ICD is headed by Peter Marwedel. It works exclusively on industrial contracts. ICD is used as a channel for transferring research results to industry. For example, compilation techniques for network processors are currently

commercialized at ICD. Furthermore, the group at Dortmund is currently leading a coordination effort on embedded systems involving about a dozen groups at Dortmund University. The effort is expected to lead to harmonized research on mobile systems. In the same area, the group contributes to the MORE project funded by the European Commission. The group is a key organizer of the SCOPES series of workshops on compilation for embedded systems. Finally, the group is actively promoting education in embedded systems through the publication of a text book and through courses at EPFL, at ALARI and at spring or summer schools in New Zealand, China and Germany.

TU Vienna is working on measurement-based timing analysis, with the aim of developing an easily portable WCET analysis framework. One of the central features of this framework is the systematic and automatic generation of test data used to exercise and measure the execution time of code fragments. Model checking is used as the basic technology behind the test data generation. Discussions with members of the theoretical computer science group at TU Munich led to several informal meetings discussing techniques of improving the test data generation engine. As a result, concrete plans of cooperation have been decided. A project probal has been submitted and is currently examined by the funding agencies.

A close cooperation exists with the ARTIST2 Execution Platforms cluster, in particular between Dortmund, Aachen, and Bologna University. RWTH Aachen participates to the HiPEAC network of excellence and is starting up new cooperations related to code optimization, e.g. with Edinburgh University. Furthermore, Aachen maintains tight industry cooperations, e.g. with CoWare, ACE, and Infineon. TU Vienna maintains a close interaction with the Lawrence Livermore National Laboratory, CA, USA, in optimizing high-level abstractions. Also IMEC is tightly integrated in European research networks, including HiPEAC. Moreover, IMEC is the central partner of a Marie Curie Host Fellowship project that involves more than 10 universities across Europe. IMEC also has many industry co-operations including most large European multi-media and communication systems oriented companies. ACE has been working closely with Aachen in this domain for some time. One of the results of this cooperation has been the integration of compiler technology in a start-up company that span out of the university. Cooperation with Imperial and Edinburgh has also started. ACE works closely with ST and with Philips having both a commercial relationship with them as well as being co-members of EU project consortia – in one case along with Verimag.

Within the EmBounded Project (IST-510255) AbsInt is also involved in the development of the Hume compiler. Hume is a domain-specific high-level programming language for real-time embedded systems.

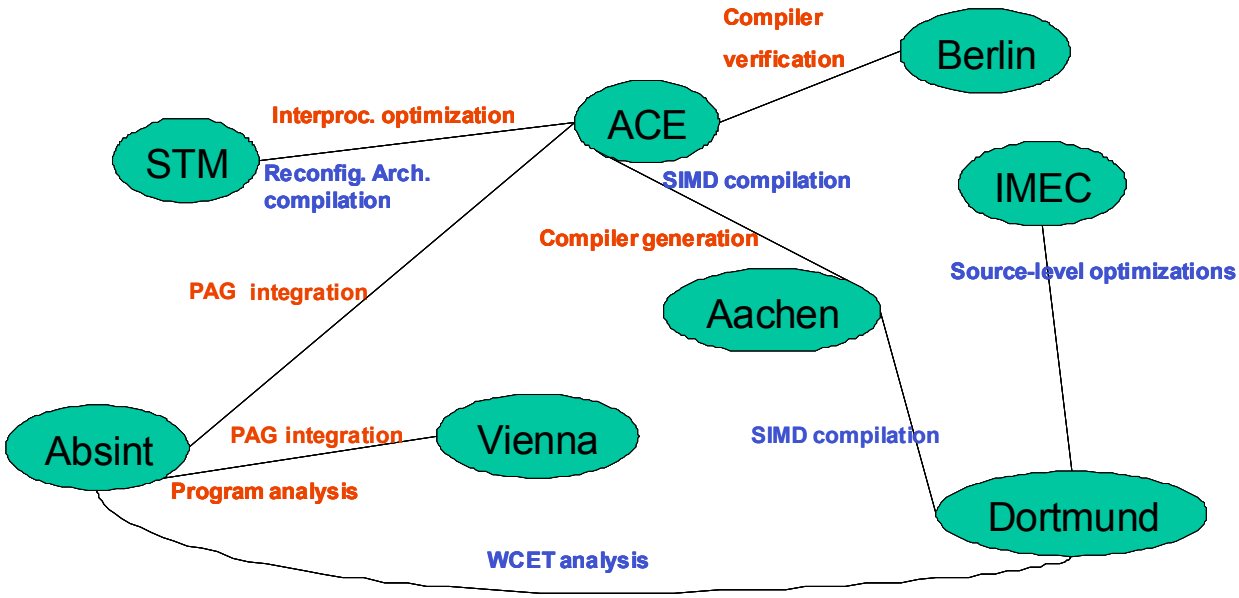
Several cluster members perform common course teaching activities (e.g. Peter Marwedel from Dortmund and Rainer Leupers from Aachen at ALARI, Lugano and EPFL, Lausanne) in cooperation with other ARTIST2 members (e.g. Luca Benini/Bologna and Lothar Thiele/Zürich). Further links of ARTIST2 members exist to the SHAPES Integrated Project and to the HiPEAC Network of Excellence.

2.3 Main Aims for Integration and Building Excellence through Artist2

The Compiler and Timing Analysis Cluster and the Execution Platforms Cluster aim at developing a common methodology to enable resource aware design and compilation. The aim of this integration activity is to increase predictability while retaining a performant system.

In the Compilers and Timing Analysis cluster, the aim is to integrate timing analysis into the compilation chain to make the compiler aware of execution time properties and constraints. For this, Uni Dortmund and AbsInt plan to integrate AbsInt's tool **aiT** into Dortmund's compiler infrastructure.

The main aim of using the ARTIST2 network is to get access to competences, knowledge and tools which are not available locally at each of the institutions. The cooperation provides the required size of research teams, necessary to handle the complexity of today's technology. Furthermore, a major goal is the combination of areas of excellence of the different partners by defining and implementing interfaces between their design flows and tools in order to achieve compilation platforms applicable to a wider problem scope. In the compiler cluster, this is implemented by the formation of several topic-specific "mini-clusters", partially based on a common compiler platform (Cosy). This form of integration works well and will be continued and intensified in the future. See below figure for the current cluster structure and connectivity (red indicates "compiler platform" activities; blue indicates architecture aware compilation activities). TU Berlin has joined as a new affiliate member.



3. Overall Technical View

3.1 Brief State of the Art

Timing Analysis

Hard real-time systems need to satisfy stringent timing constraints, which are derived from the systems they control. A schedulability analysis for the set of tasks making up the system and a given hardware architecture has to be performed in order to guarantee that all the timing constraints of these tasks will be met (timing validation). Schedulability analysis requires knowledge of upper bounds for the execution times of all the system's tasks. These upper bounds (and lower bounds) have to be safe, i.e., they must never underestimate (overestimate) the real execution time. The upper bounds represent the worst-case guarantee the developer can give.

Furthermore, the bounds should be tight, i.e., the overestimation (underestimation) should be as small as possible. Thus, the main criteria for the acceptance of a timing-analysis tool that is used to obtain guarantees are soundness of the methods-do they produce safe results?-and precision-are the derived bounds tight?

The problem of determining upper bounds on execution times for single tasks and for quite complex processor architectures has been solved. Several commercial WCET tools are available. They have experienced positive feedback from extensive industrial use in the automotive and aeronautics industry. This feedback concerned performance of the tools, precision of the results, and usability. In addition, several research prototypes are under development.

Timing-Analysis tools have to cover a rather large space originating from different application domains with their specific requirements, different classes of processor architectures, more general hardware and overall system architectures, and different user expectations. The tools offered by the cluster partners have their strengths in different points of this space.

The existing tools serve some particular and highly relevant points in this space. AbsInt's tool for example has been used in the development of safety-critical systems in the Airbus A380.

On the other hand, they currently do not serve distributed architectures well. Another interesting point in this space is occupied by timing-analysis tools adapted for teaching. This is approached by cooperation between Mälardalen and Tidorum.

The tool development is highly complex connected to the high complexity of modern processor architectures. It takes too much effort and it is error prone. Therefore, computer-supported tool-development efforts are needed.

Compilers

A large number of compiler platforms are available in industry and academia, e.g. GCC (GNU), LANCE (Dortmund/Aachen), OCE (Atair/Mentor), SUIF (Stanford), and CoSy (ACE). Compiler platforms are usually conceived as software systems that allow for quick development of compilers for new target machines and that permit efficient research by means of an open, easily extensible infrastructure.

A key problem, though, is the fact that there is still no "one-size-fits-all" platform. Each of the available platforms has its specific strengths and weaknesses w.r.t. openness, IP rights issues, code quality etc. Furthermore, different platforms serve different research requirements, e.g. some are more suitable for backend modifications while others are better for source level

transformations. Therefore, it is expected that the heterogeneous platform landscape will continue to exist in the future. Nevertheless, the members of the ARTIST2 compiler cluster have decided to largely focus on one specific compiler platform, i.e. the CoSy system by ACE.

With the increasing level of customization of embedded processors it becomes more and more obvious that architecture aware compilation is a must to achieve sufficient code quality. Application of only classical code optimization techniques, largely working at the machine-independent intermediate representation level is not good enough. Therefore, members of the ARTIST2 compiler cluster have designed numerous novel code optimization techniques, e.g. Dortmund and Aachen have extensively worked on optimizations for DSP, VLIW and network processors. This work is being continued in the context of ARTIST2.

Timing problems are expected to become more severe in the future, due to the increasing speed gap between processors and memories. Due to this gap, efforts for improving the performance of systems have been predicted to hit the “memory wall”. This means that memories will be the key limiting constraint for further performance improvements. Memories also consume a major portion of the electrical energy of embedded systems. Both problems can be addressed by introducing memory hierarchies. However, traditional memory hierarchies have been designed for a good average case behaviour. New technologies are needed in order to design fast, energy-efficient and timing predictable memory hierarchies. Some work has been done in the context of caches (loop caches, filter caches etc.). However, these approaches have mostly focussed on hardware approaches for reaching the goals. For these approaches, compilers were considered to be black boxes and untouchable. Only few authors (e.g. Barua, Catthoor, Dutt, Kandemir) have taken a holistic approach, looking at both hardware as well as software issues.

See ARTIST2 Y1 deliverables for a more exhaustive description of the state of the art. No significant change has taken place in the meantime.

3.2 Ongoing Work in the Partner Institutions

<u>Timing Analysis Activity</u>:

Saarland:

Saarland University is developing a transformation tool suite that starts with the formal specification of a processor in VHDL and transforms it in a provably correct way into an abstract timing model of this processor as it is needed by Saarland’s and AbsInt’s approach to timing analysis. This work is financed by the German Research Council (DFG) in the Transregional Collaborative Project AVACS.

The *predictability properties* of several processor components, e.g. caches, virtual memory, are analyzed and measures for improvement are developed. These concern among others partially locked caches and software-controlled caches. This work is financed by the German Research Council (DFG) in the AVACS project.

Synergy between code synthesis from formal specification and timing analysis is exploited in cooperation between Saarland University, AbsInt and several German software houses.

Tidorum

Within the Artist2 Timing-Analysis sub-cluster, Tidorum Ltd is mainly working on the definition of the common interchange representation, AIR. Since AIR is based on CRL2 as created at Saarbrucken and AbsInt, Tidorum’s role is mainly to inject requirements and suggest changes and extensions to CRL2. Outside Artist 2, Tidorum is continuing the development and commercialization of its WCET tool, Bound-T. In 2006 Tidorum has focussed on leading a

project for the European Space Agency to study the verification of the real-time performance of space-based processors with cache memory, in particular the LEON2 processor (a SPARC V8 variant). The partners in this project include Rapita Systems Ltd who supplies the RapiTime tool, the commercial form of the pWCET timing analysis tool from the University of York. Tidorum is also continuing its collaboration with Mälardalen University where Bound-T is used in the teaching of real-time systems development, in particular on the Renesas H8/300 processor. The universities of York and Mälardalen are Artist2 partners and these collaborations are synergistic with Artist2 integration.

Mälardalen

Mälardallen's main focus is automatic flow analysis. The prototype tool SWEET is used to test different methods for this. Ongoing work includes methods to speed up the analysis by program slicing, and methods to find infeasible paths. Within the newly initiated research centre PROGRESS, the group will develop methods for WCET analysis of component-based embedded software. This includes parametric WCET analysis.

The group regularly performs case studies on industrial code, where WCET analysis tools are tested on real production codes to find out where the real issues are. In most of these case studies, the group has used the aiT tool from AbsInt.

The group is an active participant in the cluster work on defining and implementing common formats and interfaces for WCET tools, as well as in other cluster activities such as meetings and workshops. It also maintains a WCET analysis benchmark suite. Finally, there is an ongoing collaboration with TU Wien regarding WCET-oriented programming and analysis of single-path properties.

AbsInt works closely with selected customers from avionics and automotive industry to adapt the WCET technology to industrial needs. The main areas are improved efficiency, i.e. to reduce the effort to produce safe timing bounds, better precision, i.e. less overestimation, and support for new modern architectures.

Besides the industrial work AbsInt works in the Embounded FET-project (IST-510255) towards the timing analysability of high-level functional programming languages and in the INTEREST STREP towards a smoother integration of European embedded systems tools.

Dortmund

Based on the interface language CRL2 of AbsInt's timing analysis tool aiT, a successful integration of timing analysis into the compiler infrastructure of Dortmund University was achieved. This way, precise timing information is available within a compiler for future optimization for the very first time. This approach will be useful in order to perform automated trade-offs between different optimization goals. For example, the influence of the loop nest splitting source code optimization on the worst-case execution time (WCET) was examined. Loop nest splitting minimizes the number of executed if-statements in loop nests of embedded applications. It identifies iterations of a loop nest where all if-statements are satisfied and splits the loop nest such that if-statements are not executed at all for large parts of the loop's iteration space. All together, considerable reductions of WCET were achieved by the source code optimization. No automated tool currently supports tradeoffs involving WCET estimates, leaving the human system designer of industrial safety critical applications alone with the error-prone task of trial-and-error compiler runs. The design of a WCET-aware compiler at Dortmund University will solve this crucial problem.

At **TU Vienna**, the group participating in the Timing Analysis platform and the group participating in the Compilers Platform cooperate on research of extending compilation to support timing analysis. The cooperation is funded by the national research project COSTA, which started in July 2006.

Compilers Activity:

Only a brief summary is provided here. See the technical descriptions of the two major cluster activities (compiler platform and architecture aware compilation) for further details.

After the successful adoption of the CoSy compiler platform, Aachen has shifted attention mainly to the architecture aware compilation activity. Aachen and ACE have been working together on a number of topics together including a SIMD optimisation framework and conditional execution optimisation.

ACE has also provided ongoing support of ST's team in relation to interprocedural optimisation. Furthermore, ACE has been providing extensive CoSy support for all partners using that platform.

The ongoing work with respect to the PAG integration (Absint-Vienna) is described in the activity report for the compiler platform.

Ongoing work in IMEC vzw focuses mainly on source to source optimizations. This means development of methodologies (Data transfer and storage exploration (DTSE)) and tools to alleviate and support the application designer when mapping an application to a multi-processor platform. The result of this design time application mapping exploration is steered with the use of various scenarios according to the run-time situations. Also, dynamic concurrent applications are increasingly important in multimedia and network applications. Several design steps are required to cost-effectively map the data and tasks in such applications to a multi-processor platform. Ongoing work in IMEC vzw within the scope of ARTIST 2, deals with these problems and has provided for this type of applications substantial reductions on cost (memory footprint, energy consumption and processor cycles) of at least a factor of 2. The work is mostly focused on (i) Dynamic data type exploration, (ii) Dynamic memory management refinement, (iii) Task-level data transfer and storage exploration and (iv) Task-level concurrency management for event-driven applications. In these research areas, synchronization is achieved with Dortmund Uni. and tighter collaboration is programmed for the last quarter of 2006.

During the last years, TU Berlin has gained significant experience in the verification of compilers. These research efforts are two-fold: On one hand side, compilation algorithms are verified, i.e., it is shown that they preserve the semantics of programs during transformations. These proofs are also formalized in the interactive theorem prover Isabelle/HOL. One the other side, checkers are developed that make sure that transformations have been applied correctly in practical compilers, i.e. by compiler implementations. This research will be continued in cooperation with ACE by verifying important compiler optimizations (as e.g. loop transformations) and by designing checkers for them.

TU Vienna und AbsInt have established a cooperation that allowed integrating AbsInt's program analysis generator PAG into the C++ infrastructure ROSE as part of the compiler platform. In Y2 all C++ language constructs (only excluding exceptions) were addressed. A shape analysis and a constant propagation were specified to operate on C++ programs.

High-level programming styles were evaluated with respect to the abstraction level and the performance of the generated code. Early results show that generic programming permits writing high-level code and with compiler technology as available in GNU g++ 4.0+ similar machine code is generated as with C programs. This performance and assembly code quality evaluation was combined with a static high-level analysis of C++ code for evaluating the impact of code optimizations in different generic programming styles. In particular, this work shows that library centric development for embedded systems is possible today but requires whole-program optimization including both application and library code.

Due to the large energy consumption of the memory system and the increasing speed gap between processors and memories, the Dortmund group worked on memory allocation

techniques reducing the energy consumption in the memory system and improving the timing predictability. The focus was on memory allocation techniques for scratch pad memories. Such memories are fast, energy efficient and timing predictable. However, they require smart tools for mapping memory objects to such memories. The groups' previous work proposed compile-time or design-time memory allocation approaches to share the scratchpad memory. The current work extends the previous work to dynamic approaches. Advanced dynamic approaches need support by an operating system. The proposed approaches have been integrated into the RTEMS operating system. A memory aware tool-chain supporting uni-processor ARM, multi-processor ARM and M5 DSP based systems was set up. Both the simulation and compilation subsystems are configured from a single memory hierarchy description. In addition, a common energy database is used by the memory optimizers in the compilation subsystem as well as by the memory and multi-processor SoC simulators in the simulation subsystem. The experiments demonstrate that for highly dynamic applications significant energy savings were achieved. The memory aware compilation developed at Dortmund University will elevate the burden of decisions regarding the utilization of the on chip memories. Additionally, it will help the system designer in determining the optimal memory parameters for their application. Relevant industrial sectors include all sectors using high-speed embedded systems (for example consumer electronics and telecommunications). Furthermore, the group worked on bit-true data flow optimizations as a processor specific source-level transformation.
Reconfigurable processors are replacing specialized hardware blocks and can provide significant savings for embedded SOC development, especially in terms of time to market. They are more generic and flexible than hardware blocks and should allow code reuse. Usually structured around a minimal core, reconfigurable processors may accept new extensions. An extension can include new resources like register banks and/or new instructions dedicated to a class of applications. To support such cores, a large effort must be spent to redesign the associated tool chain. All the components (compiler, assembler, linker, debugger, simulator etc) are impacted. To offer a high degree of flexibility, it was decided for the tools to make use of dynamic librarie(s) for the support of the extension(s) connected to the main core. Furthermore, to minimize time to market, ST's final goal is to open/
the definition of the core configuration and extension modelling to various categories of end users, mostly those in charge of the definition of final products: application developers, SOC designers, core architects. This allows faster definition, test, tuning of the extension.

Development tools for reconfigurable cores must:

- allow the user to specify the configuration and extension under need. The tools must fit the level of expertise of the user and offer the appropriate view on the configuration/extension being defined. For instance, it is possible to rely on a default encoding of the extension instructions, although, a more efficient job would probably be carried out by an experimented designer.
- derive the underlying model of the extension in different formats, such that this information flows into the generation of the shared objects to be called by each tool involved in the support the new extensions.
- ensure the consistency of the global architecture and code produced by the tool chain.

Dealing with configurable/extendable processors at compiler level implies new capabilities, especially:

- if an extension can be defined like a reconfigurable technology, the compiler must access to and make use of a software description or a 'machine model' of the extension. Though, this model must be read dynamically at compile time, and may have to deal with more numerous and different features, like non C data types, registers of large size (256, 512 or 1024-bit), or instructions with odd semantics (two results, memory side-effects, etc);

- the compiler and the rest of the toolset must have the ability to be configured to make use of new instructions and resources, with a minimal effort. Ideally, this should be feasible at end-user level. However, this can be done only if one defines what an extension can be. Otherwise, the support through user-defined built-ins can provide a first and light solution.

3.3 Interaction and Building Excellence between Partners

Timing Analysis Activity

Timing Aware Compilation
The University of Dortmund and AbsInt are integrating timing analysis into a compiler. AbsInt's aiT tool has been integrated into the compiler framework of Dortmund so that the compiler can check execution time constraints during compilation and take the WCET bounds into account when applying optimizations or performing code generation. **Parametric WCET Analysis**
Hard real-time systems need absolute bounds on execution times, i.e. expressed in numbers of machine cycles or milliseconds. Many applications, however, need parametric bounds, i.e. bounds that have numeric parameters such as the number of tasks managed or the number of requests served. Saarland University together with Mälardalen University have developed a parametric timing analysis based on the aiT technology of AbsInt.

Industrial Case Studies:
Mälardalen has performed a number of industrial case studies with AbsInt's aiT tool. Valuable insight has been gained.

Framework Integration:
Extended discussions have been done during meetings about a common exchange format for intermediate representations of the participants' tools. A proposal has been agreed on that will be made public for use by the participants. Another proposal is worked on that specifies a way to exchange semantic information about instructions embedded into the common exchange format AIR.

The definition of a common intermediate program representation (AIR) will allow for an easier exchange of tool components and results. This will allow researchers to focus more on their research tasks instead of investing a lot of time in building infrastructure. Furthermore, this will contribute to measure the quality of research results

Tidorum Ltd is collaborating with the universities of York and Mälardalen as described in the preceding section.

Measurement-based WCET Analysis
Adam Betts from the University of York visited Technical University of Vienna, for 3 months, starting October 2005. The main purpose of the visit is to do joint collaborative work on measurement based WCET analysis. As a result of the visit two research papers have been produced that are being submitted to major conferences in the area.

Activities at Dortmund University
Manifold integration activities take place at Dortmund University. The design of a timing-aware compiler within this cluster is an excellent example for the integration activities within the ARTIST2 NoE. In cooperation between AbsInt and Dortmund University, timing analysis techniques are tightly integrated into a compiler tool chain. The WCET analyzer aiT provided by AbsInt has been connected to tools designed at Dortmund. For this purpose, a common information exchange format (CRL2) was selected and now is centrally used by both the timing analyzer aiT and the WCET-aware compiler WCC. Tools from both partners are now tightly

connected and are now used on a daily basis for exploiting the synergies enabled by the work in the first two years of ARTIST2.

Compilers Activity

Interaction between the compiler cluster partners takes place via regular global meetings (usually twice per year) as well as numerous 2 or 3-partner “mini-cluster” meetings. Furthermore, there is an extensive exchange of software components. For instance, ACE is making the CoSy platform available free of charge to interested parties, and specific tools and interfaces are being exchanged for common R&D work. Staff mobility is partially used to make this exchange even more efficient. As a result, ARTIST2 has significantly contributed to the partners´ awareness of each others achievements and technologies.

IMEC is continuing to organize courses that train the participants in the most mature parts of the Data Transfer and Storage Exploration techniques that are developed within research projects like ARTIST2. These courses are attended by participants with an academic or an industrial background.

Prof. Sabine Glesner (TU Berlin) has joined the cluster as a new affiliate member. Her work complements the cluster’s previous activities by research on compiler verification.

TU Vienna und AbsInt have established cooperation by integrating AbsInt's tool PAG into the C++ infrastructure ROSE, which is developed as a cooperation between TU Vienna and the Lawrence Livermore National Laboratory, CA, USA. For visualizing the analysis results AbsInt's tool AiSee has proven useful. A textual representation of the analysis results is generated and passed to AiSee as input for visualization.

The cooperation with University of Dortmund and University of Bologna resulted in a coherent memory aware compilation and simulation tool chain. Dortmund’s memory allocator for memory hierarchies has been integrated into the MPARM SoC simulation environment of Bologna. The two partners have also agreed upon a common XML description of the multi-processor architecture which will be the basis of all future optimizations. The resulting tool chain has been used for an analysis of the potential efficiency gains effected by employing Dortmund’s optimization techniques in multiprocessor environments such as the ones studied by Bologna. Visits paved the way for removing incompatibilities. It was agreed to use the same memory architecture description format for work at both institutions. Furthermore, source-level transformations have been explored in cooperations with IMEC and RWTH Aachen.

4. Overall Assessment and Vision for the Cluster

4.1 Assessment

Timing-Analysis activity:

- The timing-analysis work in the cluster partly suffers from the success of previous research. This research has solved several central problems, in particular the prediction of the architectural behaviour for single tasks on uniprocessors. This problem can be regarded as being essentially solved. However, the development of the corresponding tools is tedious and error-prone. Research to support tool development is still needed. This research, however, is concentrated in Saarbrücken only.
- Program-flow analysis needs to be integrated with path analysis. This is planned for the later phase of the project.
- Several partners consider measurement-based methods and tools. While these can not give guarantees they may give a feel for the distance between predictions and reality and are of interest for applications where QoS is of importance, e.g. multimedia applications.
- Europe is still leading the field. Only light competition exists in the US and in Singapore.
- USaar has started cooperation about design for predictability with partners in the Execution-Platform cluster, ETHZ, Bologna, and Dortmund.
- Mälardalen will componentize its flow-analysis methods.
- Mälardalen and Vienna have been working on programming-support strategies for writing code with good timing predictability.
- Vienna has developed concepts to maintain the control-flow information needed for WCET analysis during the code transformations of optimizing compilers.
- Critical Mass is definitely present in the cluster. Two measurement-based partners and 4 analytical partners are enough to develop synergies.

Scientific discussion within the subcluster is lively. Cooperation is strong between several partners, although only pairwise in some cases.

Dependence on the software components of a commercial tool, i.e. the CRL2 internal representation of AbsInt's aiT tool, while being the only realistic and quite stable choice, introduces conflict potential between the individual commercial and survival interests and the integration goals. AbsInt has several times overspent its budget and cannot be forced to invest more on their own costs. So, the definition of the common interchange representation, AIR, is developing slower than expected. However, progress is being made, and consensus is possible, which is promising.

Compilers Activity

The compilers cluster partially suffers from the presence of relatively fragmented research topics and tool environments. Different partners have different research interests and design tools. Coupling of completely separated tools within one single platform requires huge manpower, which not all partners are willing to invest. This issue has been resolved by the formation of several topic-specific "mini-clusters", each typically comprising 2-3 partners.

Thanks to this fine grained structure, a number of successful new cooperations have been established, and previously existing cooperations have been extended. The cluster participants believe that by the mini-cluster formation a sufficient critical mass has been achieved for day-to-day research activities. Still, global coordination of the compilers cluster takes place via regular cluster-wide meetings.

While the “mini-cluster” structure is mostly static, it is also open to new developments, e.g. Sabine Glesner from TU Berlin was added as an affiliate member to the cluster due to the need to complement existing activities by compiler verification.
Dortmund’s cooperation with AbsInt and the Universities of Bologna and Linköping exceeds expectations. Future opportunities include a commercialization of some of the results, for example through COWARE, ACE, AbsInt or ICD, a technology transfer centre located at Dortmund and headed by Peter Marwedel.

4.2 Vision and Long Term Goals

The cooperation will strengthen the position of European toolmakers in industry. Usability of the tools and precision of the results will be further improved.

The next step is the integration of the single-task-on-uniprocessor methods and tools into tools considering distributed and communication-centric systems. These approaches are represented in the Cluster Execution Platforms.

Timing Analysis: As more and more experience is gained with timing-analysis tools in industrial practice, strengths and weaknesses of the different approaches and necessities for future research and development become apparent. Timing-analysis methods and tools have to be integrated into the design and development process, interacting with schedulability analyses and task allocation for distributed systems. Resource awareness has to be a first-class citizen in the development methodologies for embedded systems. Checking and verifying resource constraints after development is current practice. However, it entails a long and costly design loop; insufficient resources for a complete design and implementation may lead to a radically new design. Resource constraints should be taken into consideration early in the design process.

The problem of unpredictable resource consumption by embedded systems, be it time, space, or energy, is intolerable. A new design discipline, Design for Predictability, is urgently needed. It should cover the development of individual components as well as the interaction of layers in layered systems.

The ultimate vision is a fully integrated development process with resource needs and safely and precisely determined resource consumption communicated between components and layers through resource interfaces.

Improved availability of timing information in compilers is certainly overdue. It can be expected that this will be recognized outside this consortium as well and that timing issues will be given more attention. Partners currently cooperating on this issue could become a seed for work in this direction.

Compilers

The cooperation is strengthening the position of European researchers and toolmakers in industry. Usability of the tools and precision of the results will be further improved. The compilers cluster emphasizes this vision by active participation of key industrial players such as STM and ACE. The cooperations begun in ARTIST2 year one are being intensified, and

new upcoming research challenges are continuously taken up together by the participants in order to exploit synergy effects right from the start. The long-term goal is a stable, self-sustained cluster structure, naturally open to new research teams with excellence in specific new areas.

Research on memory-architecture aware compilation is urgently needed. Due to the obvious impact of the memory-wall problem, embedded systems will become memory-speed limited and all techniques easing the problem can be expected to find a major attention. Again, partners currently cooperating on this issue could have a major impact on technologies dealing with the memory wall problem.

4.3 Future Work and Evolution

4.3.1 Technical Description

Timing Analysis:
Most of the semantics of the common interchange format, AIR, will be defined. The partners will begin to implement (prototype) modules for exporting and importing program models and analysis results in AIR form.

The development of WCET-aware compiler optimizations will be performed by Dortmund University. On the one hand, this will include optimizations exclusively focussing on WCET as objective function, like e.g. exploitation of memory hierarchies for WCET minimization. On the other hand, the mechanisms provided by Dortmund's WCET-aware compiler developed during ARTIST Year2 for multi-objective optimization (e.g. trading off WCET vs. code size) will be used. It is intended to set up a cooperation with more ARTIST2 core partners working on timing analysis platform (e.g. Mälardalen, Vienna) in order to integrate their techniques. As a consequence, excellence will be spread among research groups that currently work in completely different domains.

The predictability issue will be of high concern for several partners. I t will mostly concern the consumption of time, but also of energy.

Compilers:
ACE and Aachen's work on SIMD will continue to improve its ability to translate a wider class of loops into SIMD instructions. The SIMD and conditional execution work has to be integrated into CoSy solving various practical retargeting issues. It is planned to evaluate the new code optimization techniques in a joint master thesis with an industrial semiconductor partner.

Also Berlin's work, in cooperation with ACE, will focus on transformation of loops and their verification. It is planned to develop checkers for general unimodular loop transformations to ensure that, in individual compilations, the source and target programs are semantically equivalent.

AbsInt – TU Vienna Cooperation: Further enhance the ROSE-PAG connection for performing whole-program source-code analysis of C/C++ applications and provide evaluation data on the scalability of an analysis. The goal is to provide enabling technology for library centric development of embedded systems codes.

In the coming months, memory optimizations to exploit memory hierarchies of multi-processor systems will be developed at Dortmund University. The group will also concentrate on developing online allocation approaches for the efficient utilization scratchpad memories by applications composed of periodic and aperiodic tasks. Additionally, our highly configurable

memory hierarchy simulator (MEMSIM) will be integrated with the multi-processor SoC simulator (MPARM) from University of Bologna. This will facilitate the development of advanced memory optimizations for the MPARM setup and will strengthen our cooperation with Bologna.

4.3.2 Current and Future Milestones

Current Milestones:

- **Timing - Analysis**
  Reinhard Wilhelm (Saarland University)
  - *Year2: Standard tool architecture and interfaces*
    The chosen interface language, AIR, is being extended by Saarland University and by AbsInt to suit the needs of other partners.
    Four partners of the team (Vienna, Mälardalen, Tidorum, AbsInt) will continue to work on path description attributes for AIR to arrive at a uniform notation.
  - *Year3: Initial integration of existing components*
    Mälardalen will wrap up its flow analysis into a component with well-defined interfaces, which will be integrated with the aiT tool of AbsInt and the Bound-T tool of Tidorum.
  - *Year4: Version 2 integration of existing components*
- **Compilers**
  Rainer Leupers (RWTH Aachen)
  (See also current 18 months work program)
  - **Year 1**: *Initial definition of common compiler platform*
    This milestone has been achieved by selection of ACE´s CoSy platform as the primary platform for most cluster partners.
  - **Year 2**: *Initial implementation of the platform*
    This milestone has been achieved by installing and adopting the platform at the partners´ sites (partially after some setup meetings and training) for teaching and research purposes (e.g. for projects related to the architecture aware compilation activity). Examples: Aachen is using CoSy presently for development of SIMD and conditional instruction based code optimization in close cooperation with ACE. Likewise, Berlin (new affiliate partner) is using CoSy for research on new compiler verification technologies. TU Vienna has integrated the Program Analyzer Generator (PAG) in the C++ infrastructure ROSE which is the source-to-source component of the compilers platform. The full C++ language has been addressed, in particular virtual methods, templates, constructor/destructor calls, function pointers, etc. - only exceptions are not addressed yet. ROSE permits generating C++ code and lowered C code. The generated code can serve as input to ACE's compiler for generating optimized machine code.
  - For **Year 3** and **Year 4**, the compiler cluster envisions a status where more and more new technologies (e.g. code optimization, verification) have been integrated into the common platform, which would lay a solid basis for self-sustained cooperations after the ARTIST2 funding period.

5. Cluster Participants

5.1 Core Partners

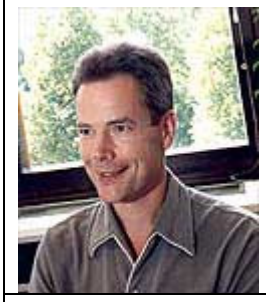
Cluster Leader Activity Leader for "Timing Analysis Platform"	
	Prof. Dr. Reinhard Wilhelm (Saarland University) http://rw4.cs.uni-sb.de/users/wilhelm/wilhelm.html
Technical role(s) within Artist2	Leading the discussion about the different approaches to timing analysis, initiating a process eventually leading to the evaluation of the different approaches and their realization under several criteria, precision, performance, usability. Pushing the idea to exploit synergies between the different phases in the design of hard real-time systems, formal specification, code synthesis, compilation, and timing analysis. Working towards a theory and corresponding practical rules to increase the predictability of embedded systems.
Research interests	Compiler design, static program analysis, embedded systems, timing analysis of hard real-time systems
Role in leading conferences/journals/etc in the area Embedded Systems	Site Coordinator Saarbrücken in the AVACS Project Member of the ACM SIGBED Executive Committee Member at Large of the ACM SIGPLAN Executive Committee Member of the Steering Committee of the International Conference on Embedded Software EMSOFT Member at Large of the Steering Committee of the ACM Conference on Languages, Compilers, and Tools for Embedded Systems LCTES
Notable past projects	DAEDALUS Shared-cost research and technology development (RTD) project IST-1999-20527 of the European IST Programme of the Fifth Framework Programme (FP5) on the « validation of software components embedded in future generation critical concurrent systems by exhaustive semantic-based static analysis and abstract testing methods based on abstract interpretation. http://www.di.ens.fr/~cousot/projects/DAEDALUS/index.shtml COMPARE ESPRIT project, developed a new compiler technology, CoSy, enabling the efficient construction of compilers to suit specific hardware architectures http://cordis.europa.eu/esprit/src/results/pages/infotec/inftec14.htm
Awards / Decorations	ACM Fellow, Alwin-Walther Medal

Cluster Co-Leader **Activity Leader for "Architecture Aware Compilation"**	
	Prof. Dr. Rainer Leupers http://www.iss.rwth-aachen.de
Technical role(s) within Artist2	Compiler platform, code optimization
Research interests	Compilers, ASIP design tools, MPSoC design tools
Role in leading conferences/journals/etc in the area	TPC member of DAC, DATE, ICCAD etc. Co-founder of SCOPES workshop
Notable past projects	HiPEAC NoE, SHAPES IP, several DFG-funded projects Industry-funded projects with Infineon, Philips, Microsoft, CoWare, ACE, Tokyo Electron etc.
Awards / Decorations	Several IEEE/ACM best paper awards
Further Information	Co-founder of LISATek Inc. (acquired by CoWare Inc.)

	Dr. Christian Ferdinand (AbsInt Angewandte Informatik GmbH) http://www.absint.com/
Technical role(s) within Artist2	Christian Ferdinand coordinates the activities of AbsInt within Artist 2
Research interests	Timing analysis, program optimization, compiler construction.

Partner	
	Hans van Someren (ACE, Netherlands) www.ace.nl
Technical role(s) within Artist2	Supporting ARTIST2 partner use of compiler platform (CoSy compiler development system) – ranging from initial training through to project/design reviews. The design and construction of extensions to CoSy required for ARTIST2 projects.
Research interests	The development and exploitation of compilation techniques and development systems in the wider contexts of SoC and EDA supported by descriptions of the system including application and target architectures. Particular interests include MPSoC and highly parallel system.
Role in leading conferences/journals/etc in the area	Programme Committees of SCOPES and DATE.
Notable past projects	COMPARE/PREPARE ESPRIT projects: These projects, particularly COMPARE, were precursors for CoSy. PREPARE focused on retargetable compilation for Fortran 90 and High Performance Fortran using massively parallel MIMD machines. MESA/NEVA (ongoing): Framework IPs addressing the challenges of designing and constructing multi-processor systems.
Further Information	Principal architect of the CoSy. Previously, architect of ACE's shared memory heterogeneous multiprocessor UNIX OS.

	Prof. Dr. Peter Marwedel (University of Dortmund) http://ls12-www.cs.uni-dortmund.de/~marwedel/
Technical role(s) within Artist2	Improved code quality for embedded application is the main goal of the work at Dortmund University. Due to the widening gap between processor and memory speeds, emphasis has been on improving the efficiency of memory accesses, in terms of average and worst case execution time and in terms of the energy consumption.
Research interests	Peter Marwedels Embedded Systems Group focuses on embedded software. Particular emphasis is on compilers for embedded processors. One of the very first publications in this area, the book "Compilers for Embedded Processors", edited by Peter Marwedel and Gert Goossens, was the result of the CHIPS project, funded by the European Commission. The group's current focus is on advanced optimizations for embedded processors (e.g. by using bit-level data flow analysis) and energy-aware compilation techniques. Current research also includes high-level transformations of algorithms as well as WCET-aware code generation.
Role in leading conferences/journals/etc in the area	Member of the EDAA (European Design and Automation Association) Main Board. Editorial Board Member of ACM Transactions on Embedded Computing Systems. Editorial Board Member of the Journal of Embedded Computing. Editorial Board Member of the Microelectronics Journal. Co-Founder and Steering Board Chair of the SCOPES Workshop (Software and Compilers for Embedded Systems) Series.
Notable past projects	MAMS: Multi-Access modular-services framework, national project funded by the German Federal Ministry of Education and Research (BMBF) MORE: Network-centric Middleware for group communications and resource sharing across heterogeneous embedded systems, supported by the European Commission http://www.ist-more.org HiPEAC: European NoE on High-Performance Embedded Architecture and Compilation http://www.hipeac.net
Awards / Decorations	IEEE Senior Member
Further Information	CEO of the Informatik Centrum Dortmund (ICD), a technology transfer centre founded in 1989.

	Prof. Björn Lisper (Mälardalen University) http://www.idt.mdh.se/personal/blr/
Technical role(s) within Artist2	Timing analysis, program analysis.
Research interests	Timing analysis, static program analysis, language design for embedded and real-time systems, program transformations.

	Dr. Niklas Holsti (Tidorum Ltd) http://www.tidorum.fi
Technical role(s) within Artist2	Participate in the definition of the common WCET tool-set architecture, the analysis modules and the interchange representations (languages, file formats). Adapt Tidorum's WCET tool, Bound-T, to integrate with the architecture and interchange formats defined in ARTIST2.
Research interests	Static analysis of the worst-case execution time of embedded programs.

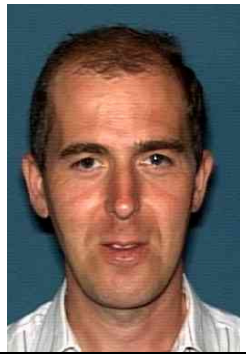
	Prof. Dr. Peter Puschner (TU Vienna) Real-Time Systems Group Institute of Computer >Engineering Vienna University of Technology http://www.vmars.tuwien.ac.at/people/puschner.html
Technical role(s) within Artist2	Peter Puschner and his group are participating in the Timing-Analysis activities of the Compilation and Timing Analysis cluster. Within ARTIST2 the contributions are in the area of path-description languages for static WCET analysis, compilation support for WCET analysis, methods and problems of measurement-based execution-time analysis, and on software and hardware architectures that support time predictability.
Research interests	Peter Puschner's main research interest is on real-time systems. Within this area he focuses on Worst-Case Execution-Time Analysis and Time-Predictable Architectures.
Role in leading conferences/journals/etc in the area	Member of the Euromicro Technical Committee on Real-Time Systems, the steering committee of the Euromicro Conference on Real-Time Systems (ECRTS) Member of the advisory board and organizers committee of the IEEE International Symposium on Object- and Component-Oriented Distributed Computing (ISORC) conference series Chair of the Steering Committee of the Euromicro Workshop on Worst-Case Execution-Time Analysis (WCET) series
Notable past projects	DECOS - *Dependable Embedded Components and Systems* Develop the basic enabling technology to move from a federated distributed architecture to an integrated distributed architecture. http://www.decos.at MoDECS - *Model-Based Development of Distributed Embedded Control Systems* Model-based construction of distributed embedded control systems: shift from a platform-oriented towards a domain-oriented, *platform-independent* development of composable, distributed embedded control systems. http://www.modecs.cc NEXT TTA Enhance the structure, functionality and dependability of the time-triggered architecture (TTA) to meet the cost structure of the automotive industry, while satisfying the rigorous safety requirements of the aerospace industry. http://www.vmars.tuwien.ac.at/projects/nexttta/

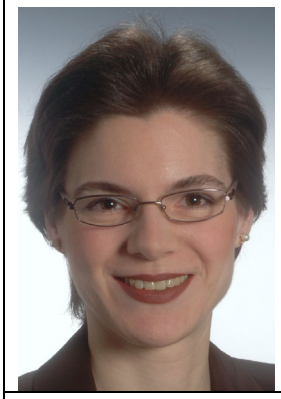
	Dr. Guillem Bernat (University of York) http://www-users.cs.york.ac.uk/~bernat/
Technical role(s) within Artist2	Responsible for the WCET cluster at University of York. Also involved in the flexible scheduling cluster.
Research interests	Worst-case execution time analysis. Especially measurement based methods. Flexible scheduling and real-time systems in general.
Further Information	Also CEO of Rapita Systems Ltd. a spin-off company commercialising RapiTime. A tool for measurement based WCET analysis.

5.2 Affiliated Academic Partners

	Prof. Andreas Krall (TU Vienna) www.complang.tuwien.ac.at/andi/
Technical role(s) within Artist2	Leader of the TU Vienna group working on program analysis and back end optimizations.
Research interests	Implementation of object oriented languages, compiler back ends and computer architecture, implementation of logic programming languages.
Role in leading conferences/journals/etc in the area	PC Member of the International Workshop on Software and Compilers for Embedded Systems
Notable past projects	Christian Doppler Laboratory: Compilation Techniques for Embedded Processors The aim of the CD laboratory Compilation Techniques for Embedded Processors is to develop the necessary compilation and decompilation techniques to make the production of highly optimizing compilers and decompilers for embedded processors feasible. http://www.complang.tuwien.ac.at/cd/

	Dr. Markus Schordan (TU Vienna) www.complang.tuwien.ac.at/markus
Technical role(s) within Artist2	Leader of the TU Vienna project group working on program analysis and optimization tools for high-level languages.
Research interests	Analysis of object-oriented languages, alias analysis, source-to-source infrastructures, high-level optimizations, and parallelization.
Role in leading conferences/journals/etc in the area	PC Member of Joint Modular Languages Conference (JMLC) since 2003. PC Member of International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC) since 2005.
Notable past projects	ROSE (2001-2003) The goal in the project ROSE was to build an infrastructure for C/C++ source-to-source transformation of multi million line applications at Lawrence Livermore National Laboratory, CA, USA. This work has been continued as a cooperation with TU Vienna since 2004. http://www.llnl.gov/casc/rose/ PAOLA (1999-2001) The goal of PAOLA was the research of techniques for context-sensitive analysis of object-oriented languages (with focus on Java programs). The project was a cooperation between University Klagenfurt (Austria) and Friedrich Schiller-Universität Jena (Germany).

	Professor Francky Catthoor (IMEC vzw.)
Technical role(s) within Artist2	Collaboration with Dortmund Uni. on high-level transformations for source code optimization.
Research interests	Francky Catthoor received a Ph.D. in El. Eng. from the K.U.Leuven, Belgium in 1987. Since then, he has headed several research domains in the area of architectural methodologies and system synthesis for embedded multimedia and telecom applications, all within the DESICS division at IMEC, Leuven, Belgium. His current research activities mainly belong to the field of system-level exploration, with emphasis on data storage/transfer and concurrency exploitation, both in customized and programmable (parallel) instruction-set processors.
Role in leading conferences/journals/etc in the area	Francky Catthoor has (co-)authored over 500 papers in international conferences and journals, and has worked on 8 text books in this domain. He was the program chair and organizer of several conferences including ISSS'97 and SIPS'01.
Notable past projects	CATHEDRAL II project Production of a synthesis system for multiprocessor DSPs. DAB-LP project Specification transformations to minimise access to large memories and distant data in DSP systems; applied to a DAB IC. DACMA project Design Methodologies and Advanced Designs for Communication and Multimedia Applications
Awards / Decorations	Francky Catthoor is currently a research fellow within the DESICS division at IMEC and an IEEE fellow.
Further Information	Francky Catthoor is also professor at the K.U.Leuven.

	Prof. Dr. Sabine Glesner (Technical University of Berlin) www.pes.cs.tu-berlin.de
Technical role(s) within Artist2	Activity Leader for Compiler Platform Compiler Verification
Research interests	Compilers, Verification, Embedded Systems and Software, Formal Semantics
Role in leading conferences/journals/etc in the area	PC Member of Compiler Construction 2007 Date'06, Design, Automation and Test in Europe, TPC Member of Topic B9 on Formal Verification Workshop Compiler Optimization meets Compiler Verification COCV, ETAPS Conferences, PC Member in 2005 and 2006, Program Co-Chair in 2007 Workshop Formal Foundations of Embedded Software and Component-Based Software Architectures (FESCA), ETAPS Conferences, PC Member 2005 and 2006 Editorial Board Member of “Informatik – Forschung und Entwicklung” by Springer, starting with Vol. 21, No. 1
Notable past projects	VATES (Verification and Transformation of Embedded Systems), funded by DFG, will start soon Aktionsplan Informatik (Emmy Noether-Program), funded by DFG, support for young researchers to build a research group, with a focus on optimization and verification in the compilation of higher programming languages, from 2004 to 2009 Correct and Optimizing Compilers for Modern Processor Architectures, funded by a postdoc excellence program of Baden-Württemberg, Germany, 2003-2005 Grant in the Wrangell-Habilitation Program of Baden-Württemberg, Germany, 2001-2005
Awards / Decorations	Award of the “Forschungszentrum Informatik” for one of the two best PhD theses of the Faculty for Computer Science, University of Karlsruhe, 1998/99 Member of the „Studienstiftung des deutschen Volkes”, the german national scholarship organization, 1991-1996 Fulbright grant to study at the University of California, Berkeley, 1993-1994 Member of the Siemens Internationaler Studenten / Doktorandenkreis, 1993-1999