IST-214373 ArtistDesign
Network of Excellence
on Design for Embedded Systems

Activity - Progress Report for Year 1

Timing Analysis

Clusters:
SW Synthesis, Code Generation and Timing Analysis

Activity Leader:

Prof. Björn Lisper (Mälardalen University)
http://www.idt.mdh.se/~blr/

Policy Objective (abstract)

The activity gathers the most prominent groups in the timing analysis area. They have all previously worked together in the ARTIST2 NoE, and therefore have well established links. The theme of the activity, timing analysis of MPSoC systems, is basically a new field scientifically, and also very timely from an application perspective as MPSoC and Multicore architectures rapidly are becoming mainstream. A research effort in this area will thus establish European dominance in a field that rapidly is becoming very important.

ArtistDesign also provides a close to perfect environment for this research due to the relevant competence in other activities and clusters, such as the compiler groups in the local cluster, and the MPSoC cluster.

Versions

number	comment	date
1.0	First version delivered to the reviewers	December 19th 2008

Table of Contents

1. Overview of the Activity

1.1 ArtistDesign Participants and Roles

Prof. Dr. Reinhard Wilhelm – Saarland University (Germany)
Compiler Design, Static Program Analysis, Timing Analysis
Saarland University has developed much of the timing-analysis technology that is further developed and commercialised by the spin-off company AbsInt.

Dr. Iain Bate – University of York (UK)
Research on Timing Analysis

Prof. Dr. Björn Lisper – Mälardalen university (Sweden)
Activity Leader, Timing-AnalysisTtools
Mälardalen University is working on automatic flow analysis, WCET analysis case studies on industrial code, the maintenance of a WCET-benchmark suite, the definition of interface formats for timing analysis, and the use of WCET tools in education. Mälardalen University is coordinating the integration activity.

Dr. Jan Gustafsson – Mälardalen university (Sweden)
Timing Analysis Research and Tools, WCET Analysis Case Studies

Dr. Andreas Ermedahl – Mälardalen university (Sweden)
Timing Analysis Research and Tools, WCET Analysis Case Studies

Prof. Dr. Peter Puschner – TU Vienna (Austria)
Timing-Analysis Tools and Temporally Predictable HW-SW Architectures
Within the Timing-Analysis Activity TU Vienna focuses on measurement-based WCET analysis and on hardware and software architectures that provide time-predictability and composability.

Prof. Dr. Peter Marwedel – TU Dortmund (Germany)
Architecture-Aware Compilation, Low-Power Code Generation, Development of Optimizations for WCET Minimization.

Dr. Claire Burguiere – Saarland University (Germany)

research on Scheduling and Timing Analysis in the presence of interrupts.

Dr. Raimund Kirner – TU Vienna (Austria)
Timing-Analysis Tools and Compilation with Support for Timing Analysis, definition of Annotation Language for WCET Analysis, Measurement-Based Timing Analysis.

Oleg Parshin – Saarland University, Saarbrücken (Germany)
PhD student, *research on the integration of Code Synthesis, Compilers, and Timing Analysis*

Jan Reineke – Saarland University, Saarbrücken (Germany)
PhD student, research on Timing Analysis and Timing Predictability.

Sebastian Altmeyer – Saarland University, Saarbrücken (Germany)
PhD student, research on Timing Analysis and Scheduling.

1.2 Affiliated Participants and Roles

Dr. Christian Ferdinand – AbsInt GmbH (Germany)

Tool Supplier

AbsInt provides advanced WCET analysis tools for a wide variety of targets. The work within ARTIST2 focuses on the advance of WCET analysis techniques by providing and defining interchange formats for the components of WCET tools like AIR (ARTIST2 Intermediate Program Representation for WCET tools).

Gernot Gebhard – AbsInt GmbH (Germany)

Generic specification of processor components for generation and validation of timing analyses.

Dr. Niklas Holsti – Tidorum Ltd. (Finland)

Timing-Analysis Tools

Tidorum Ltd supplies the timing analysis tool Bound-T. Tidorum takes part in the definition of the architecture of the tool platform and in particular in the definition of the interchange representation, AIR. Later, Tidorum will integrate Bound-T with the platform by adding AIR export and import functions.

1.3 Starting Date, and Expected Ending Date

January 1st, 2008 until there is a framework for timing analysis of MPSoC systems.

1.4 Policy Objective

The activity gathers the most prominent groups in the timing analysis area. They have all previously worked together in the ARTIST2 NoE, and therefore have well established links. The theme of the activity, timing analysis of MPSoC systems, is basically a new field scientifically, and also very timely from an application perspective as MPSoC and Multi-core architectures rapidly are becoming mainstream. A research effort in this area will thus establish European dominance in a field that rapidly is becoming very important.

ArtistDesign also provides a close to perfect environment for this research due to the relevant competence in other activities and clusters, such as the compiler groups in the local cluster, and the MPSoC cluster.

1.5 Background

All the partners in this activity have participated in the NoE Artist2. They developed a common tool architecture, and exchanged tool components. They have created a WCET Tool Challenge, executed in the first time in 2006, to evaluate the existing commercial tools and academic prototypes. The Tool Challenge will be executed every second year with improved conditions and more challenging benchmarks.

1.6 Technical Description: Joint Research

Traditional timing analysis has three parts: the flow analysis, which finds constraints on the possible program flows, the low-level analysis, which applies hardware timing models to obtain timing estimates for short execution paths, and the calculation which combines the result of the two previous analyses to obtain an estimate of the WCET for the full code.

Timing analysis on code level has so far dealt almost exclusively with sequential programs running in isolation. For MPSoC and multi-core architectures, these assumptions will no longer

be valid: tasks might be parallel, and different tasks will run in parallel on different sets of cores. Timing analysis of parallel code, running on parallel hardware, is a new research area, and the aim of this activity is to initiate research in this area. Due to the novel nature much of the research, at least for the first 18 months, will have he nature of initial investigations, paving the way for future in-depth research.

Some research problems:

- Flow analysis has to be extended from single-threaded programs to multi-threaded programs with possible synchronization between threads.
- Current low-level analysis is restricted to synchronous processor models: only [Thesing06] has modelled processor periphery. Hardware modelling must be extended to include asynchronous systems including, ultimately, full MPSoC and multicore architectures.
- New WCET calculation methods must be invented, which take into account that several interacting threads may have to complete before a task has completed.
- Methods to handle common resources must be devised. For single-processor systems, interference between tasks through shared resources like caches can be dealt with on the scheduling level, by bounding the number of preemptions and calculating a maximal timing penalty. For parallel processors, common resources can potentially be accessed at any time by totally unrelated activities. This renders traditional scheduling theory useless to estimate costs from interference with other tasks running in parallel.

Some ideas how to tackle the research problems are given below:

For flow analysis, there are several possibilities. One is to consider restricted parallel programming models, like Bulk Synchronous Programming, which have been developed in the parallel programming area in order to ease the task of parallel programming. These programming models have simple cost models, which should translate into more predictable timing models. Another possibility may be to use timing analysis to derive a Timed Automaton modelling the parallel code, and use the TA to analyze its synchronization properties. A third possibility is to use information from a parallelizing compiler. Such compilers sometimes use internal representations describing the computation in an abstract way, like an explicit task graph, or a polyhedral index set for sets of loop body executions, which is allocated and scheduled. The compiler then actually has considerable knowledge about where and when different computations are performed, which could be used to help predict the timing.

For low-level analysis, the necessary hardware modelling should start with a formal specification of the architecture, and be based on sound methods of abstraction, analysis, and transformation. The attainable accuracy of the models will be critically dependent on the hardware architecture: thus, research is necessary to find suitable MPSoC architectures which are amenable to timing analysis.

The calculation methods will depend on the program execution model. Thus, research to find appropriate such methods will be strongly connected to the flow analysis research.

The common resources problem is a matter of both hardware and system design. As for low-level analysis, research into MPSoC architecture and systems is necessary to reduce the interference between tasks. In particular on-chip networks and memories are crucial components which have to be designed to allow predictable timing. A hypothesis is that the ability to dynamically partition the resources, like assigning different parts of the network to different tasks, is helpful in this regard.

In the first 18 months, we foresee the following activities and potential results:

1. Derivation of timing models from MPSoC designs given in a language like Verilog or VHDL.
2. Meetings with researchers in Timed Automata (Modelling & Validation Cluster) to discuss the possible connections between timing analysis on code level and timing analysis on model level. Possible outcome: a report describing one or several combined approaches to the problem of analyzing parallel software with respect to timing properties such as WCET.
3. An investigation whether restricted models for parallel programming can make the problem of WCET analysis easier to solve for programs adhering to these models. Possible outcome: a survey of potentially interesting parallel programming models, with an assessment of their respective amenability to WCET analysis.
4. A joint activity with the MPSoC cluster, where TA expertise is fed back to MPSoC architecture level. Task: to identify features of MPSoC architectures that are critical to the predictability of timing properties, and to suggest possible designs which make the architectures more predictable with respect to these properties. Evident targets are shared resources like on-chip networks and shared memories. Possible outcome: a report describing the problem and some possible solutions, with their respective pros and cons.

1.7 Problem Tackled in Year 1

We studied the problem of timing predictability further. On the level of individual cores, work at Saarland University gave for the first time ever a precise and useful definition of predictability of cache replacement policies, competitiveness and sensitivity. This work can pave the ground for the analysis of cache predictability in the presence of multiple core and of sensitivity to disturbances from outside influences.

Additionally, work on quantifying the influence of preemptive scheduling on cache contents allows to safely bound effects of preemption on WCET analysis. Furthermore, new work aims at guiding developers to select preemption points which minimise the costs of preemption.

Work has started on the elaboration of design principles for architectures with predictable and composable timing behaviour. Various kinds of side effects, interferences that inhibit composability and impair predictability of single- and multi-processor systems have been identified. A small set of mechanisms that rely on pre-planned control to protect the time-relevant state from unforeseen changes has been proposed.

Work has been carried out in the area of measurement-based timing analysis. Such methods can be appropriate when timing behaviour is very complex and hard to model, such as for MPSoC systems, and real-time requirements are soft. Problems tackled include test-case generation, and the use of learning techniques to identify timing model features.

Previous work in parametric WCET analysis has been continued. Parametric WCET analysis calculates a formula for the WCET, in some input parameters, rather than a single number. There are strong relations between the techniques used for this analysis and techniques used in parallelizing compilers, especially the polytope method.

2. Summary of Activity Progress

2.1 Technical Achievements

Timing Analysis and Timing Predictability (USaar and AbsInt)
The notion of predictability of cache architectures has been clarified. A definition of predictability of caches has been given, and the relative competitiveness of four different cache replacement strategies (LRU, PLRU, FIFO, MRU) has been analysed. In particular, by relating the hit and miss rates for different cache analyses, it was shown that sound analyses for FIFO and MRU can be built from analyses for FIFO. This was the first such work to formally define cache predictability and to rigorously compare different replacement policies. Similarly, sensitivity of cache replacement policies to the initial state have been investigated.

The PREDATOR project in the 7th Framework Programme attempts to reconcile performance and predictability. It has identified the PROMPT (PRedictability Of Multi-processor Timing) design rules for predictable multi-processor design. The first principles are to avoid interference on shared resources in the architecture and to allow the application designer the mapping of applications to target architecture without the introduction of new interferences that were not present in the application.

WCET Analysis for Cooperative Task Scheduling (USaar)
We investigated timing-analysis aspects of cooperative scheduling/deferred preemption. Deferring preemption enables a tradeoff between the flexibility of a preemptive schedule and the predictability of a non-preemptive one. A method guiding developers of an embedded system to select optimal preemption points (with respect to minimizing the maximal blocking time and/or preemption costs) is under development. To bound the context switch costs, we can use known approaches, but also incorporate the extra knowledge given by the set of preemption points. A technique to derive the maximum blocking time is already available.

Parametric Timing Analysis (USaar, AbsInt and Mälardalen)
Timing analyses require that information such as bounds on the maximum numbers of loop iterations are known statically, i.e., during design time. Parametric timing analysis softens these requirements: it yields symbolic formulas instead of single numeric values representing the upper bound on the task's execution time. So, some input parameters to the program can remain unknown until the final use of the task. The developed analysis determines the parameters of the program, constructs parametric loop bounds, takes processor behaviour into account and attains a formula automatically.

Timing Analysis and Timing Composability (TU Vienna)

The complexity of hardware and software architectures used in today's embedded systems are not only a problem to timing predictability, but also make a hierarchical, composable timing analysis impossible. We investigated in which way side effects, caused by the interplay of hardware and software mechanisms influence the timing of embedded applications and undermine composability. The lack of composability presents a big obstacle to a clean hierarchical design process for real-time embedded applications. We thus propose a number of measures that eliminate dynamic interferences and pave the ground for a system timing (for single- and multi-processor systems) that is both predictable and compositional.

Using Learning to Support the Development of Embedded Systems (Univ of York)
This project seeks to investigate the application of techniques from artificial intelligence and software testing to build models upon which the analysis of real-time systems, specifically the Worst-Case Execution Times (WCET) of tasks can be based. The first part of this work deals with the model inference part of the WCET problem that allows measurements taken during dynamic analysis to be used to generate safe models of behaviour. This years two papers have been published. The first (Bate and Kazakov) showed how particular hardware features can be correctly identified based on program traces and then appropriate analysis be built. A branch predictor example is shown in the paper. The second paper (Bartlett, Bate and Kazakov) enhanced our previous work on program flow analysis to improve the scalability of the approach.

Test-Case Generation for WCET Analysis (Univ of York, TU Vienna)
Sven Bunte from TU Vienna visited York for one week to plan some cooperative activities between the two groups. Wide ranging discussions took place but one specific activity was chosen. Whilst there has been a great deal of research on dynamic and hybrid analysis for WCET, one fundamental issue has been overlooked. That is, without appropriate test data the quality of the results are questionable. Specifically it is important the coverage of the code is sufficient to give good results with statistical confidence. Here is coverage is considered to include not only the structure of the software but should also allow for hardware and data. In the Spring 2009, a joint activity is to be performed (using resources from other funded projects) that will examine which coverage metrics are important and how this coverage can be achieved. In York there is experience with search-based methods that are considered good for achieving broad coverage of metrics and in Vienna model checking is used that can give tightly focussed effort. The combination should allow the vast majority of the software to be covered quickly and with the minimum of human effort and then the remaining cases to be solved in an effective manner. The work would be both novel and timely.

2.2 Individual Publications Resulting from these Achievements

Usaar

- Jörg Herter, Jan Reineke, and Reinhard Wilhelm. *CAMA: Cache-Aware Memory Allocation for WCET Analysis*. In Marco Caccamo, editor, Proceedings Work-In-Progress Session of the 20th Euromicro Conference on Real-Time Systems, pages 24–27, July 2008.
- Daniel Grund and Jan Reineke. *Estimating the Performance of Cache Replacement Policies*. In MEMOCODE '08: Proceedings of the 6th IEEE/ACM International Conference on Formal Methods and Models for Codesign, pages 101–111, June 2008.
- Jan Reineke and Daniel Grund. *Relative Competitiveness of Cache Replacement Policies.* In SIGMETRICS '08: Proceedings of the 2008 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, pages 431–432, June 2008.
- Jan Reineke and Daniel Grund. *Relative Competitive Analysis of Cache Replacement Policies.* In LCTES '08: Proceedings of the 2008 ACM SIGPLAN-SIGBED conference on Languages, compilers, and tools for embedded systems, pages 51–60, June 2008.
- Reinhard Wilhelm and Björn Wachter. *Abstract Interpretation with Applications to Timing Validation*. In Aarti Gupta and Sharad Malik, editors, CAV, volume 5123 of

Lecture Notes in Computer Science, pages 22–36. Springer Verlag, 2008. Princeton, NJ, USA.

- Jan Reineke. *Caches in WCET Analysis: Predictability, Competitivenes, Sensitivity.* Dissertation, Saarland University, November 2008.

TU Vienna

- Peter Puschner and Martin Schoeberl.*On Composable System Timing, Task Timing, and WCET Analysis*. In Proc. 8th Euromicro Workshop on WCET Analysis, p. 91-101, 2008.
- Raimund Kirner and Peter Puschner. *Obstacles in Worst-Case Execution Time Analysis*. In Proc. 11th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing, p. 333-339, 2008.

York

- Bate and D. Kazakov, New Directions in Worst-Case Execution Time Analysis. In *Proceeding of the 2008 IEEE World Congress on Computational Intelligence*. 2008.
- M. Bartlett, I. Bate and D. Kazakov, Challenges in Relational Learning for Real-Time Systems Applications. In *Proceeding of the 18th International Conference on Inductive Logic Programming*. 2008.

Mälardalen

- Stefan Bygde and Björn Lisper. *Towards an Automatic Parametric WCET Analysis*. In Raimund Kirner (ed.) Proc. 8th International Workshop on Worst-Case Execution Time Analysis, (WCET'2008), pp. 9-17, Prague, Czech Republic, July 2008.

2.3 Interaction and Building Excellence between Partners

Interaction between AbsInt and Saarland:

AbsInt and Saarland are cooperating on the integration of code synthesis, compilers, and timing analysis, on timing predictability, on scheduling, and on analysis of multi-core architectures.

Interaction between Saarland and Mälardalen:

The joint work on parametric WCET analysis has continued through the writing of a joint publication (RTCSA 08; see Section 2.4).

Interaction between Saarland, Mälardalen, Tidorum, AbsInt, York, TU Vienna:

The joint writing of a survey paper on WCET analysis and tools resulted in a publication.

2.4 Joint Publications Resulting from these Achievements

- Sebastian Altmeyer, Christian Hümbert, Björn Lisper, and Reinhard Wilhelm: Parametric timing analysis for complex architectures. Proc. 14th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 08), Kaohsiung, Taiwan, Aug. 2008

- Sebastian Altmeyer and Gernot Gebhard. WCET Analysis for Preemptive Systems. In Raimund Kirner, editor, *Proceedings of the 8th International Workshop on Worst-Case Execution Time (WCET) Analysis*, pages 105–112, Prague, Czech Republic, July 2008. OCG.
- Daniel Kästner, Reinhard Wilhelm, Reinhold Heckmann, Marc Schlickling, Markus Pister, Marek Jersak, Kai Richter, Christian Ferdinand: Timing Validation of Automotive Software. 3rd International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISOLA), Kassandra, Greece, 2008.
- Reinhard Wilhelm, Jakob Engblom, Andreas Ermedahl, Niklas Holsti, Stephan Thesing, David Whalley, Guillem Bernat, Christian Ferdinand, Reinhold Heckmann, Tulika Mitra, Frank Mueller, Isabelle Puaut, Peter Puschner, Jan Staschulat, and Per Stenström. *The Worst-Case Execution-Time Problem -- Overview of Methods and Survey of Tools*. In ACM Transactions on Embedded Computing Systems (TECS), 7(3), April 2008.

2.5 Keynotes, Workshops, Tutorials

Tutorial: Timing Analysis and Timing Predictability
Embedded Networked Systems: Theory and Applications
Heraklion, Crete – July 21–25, 2008

The 2008 Lectures in Computer Science of the Onassis Foundation were dedicated to theory and applications of Embedded Systems. Among the talks by leading researchers was a two-part tutorial by Reinhard Wilhelm on Timing Analysis and Timing Predictability.
http://www.forth.gr/onassis/lectures/2008-07-21/lecturers.html

Tutorial: Abstract Interpretation with Applications to Timing Validation

Princeton*, USA – July 7–14, 2008*

This invited tutorial was given by Reinhard Wilhelm (Saarland University). It explained the technique of abstract interpretation and its application to static timing analysis.
http://www.princeton.edu/cav2008/

Tutorial: Timing Analysis and Timing Predictability

ARTIST2 Summer School 2008 in Autrans (near Grenoble), September 2008

Reinhard Wilhelm gave this tutorial.
http://www.artist-embedded.org/artist/ARTIST2-Summer-School-2008.html

Keynote: Constructing Time-Critical Embedded Systems: Use Your Intelligence Before Runtime
6th Workshop on Intelligent Solutions in Embedded Systems

Regensburg, Germany – July 10-11, 2008

This presentation, given by Peter Puschner, examined the complexity of contemporary hardware and software architectures and demonstrated how the sophisticated mechanisms used lead to difficulties in understanding and analysing the timing of embedded real-time applications. It was argued that a new trend towards simplicity is needed that avoids speculation and minimizes the number of dynamic decisions taken at runtime. Following these principles one can eliminate timing variations, system timing becomes easy to understand, and proofs for temporal correctness turn out to be almost trivial.
http://fbim.fh-regensburg.de/~wises08/index.htm

Workshop : 8th Int'l Workshop on Worst-Case Execution Time Analysis (WCET'08)
Prague, Czech Republic – July 1st, 2008

The 8th International Workshop on Worst-Case Execution Time Analysis (WCET 2008) was held as a satellite event to the 20th Euromicro Conference on Real-Time Systems (ECRTS 2008). The goal of the workshop is to bring together people from academia, tool vendors and users in industry that are interested in all aspects of timing analysis for real-time systems. The workshop fosters a highly interactive format with ample time for in-depth discussions. It provides a relaxed forum to present and discuss new ideas, new research directions, and to review current trends in this area. The presentations are kept short to leave plenty of time for interaction of attendees.

The WCET 2008 event of this workshop has gained on popularity, it had 45 registered participants. This tendency may be interpreted that the real-time community becomes increasingly aware of the importance of WCET analysis. The workshop program included four regular sessions with 13 talks on WCET analysis. Additionally an invited talk on timing analysis at system level was given by Prof. Rolf Ernst and the current results of the WCET Tool Challenge 2008 were presented by the organizer, Niklas Holsti.
http://www.artist-embedded.org/artist/WCET-08.html

3. Milestones, and Future Evolution

3.1 Problem to be Tackled over the next 12 months (Jan 2009 – Dec 2009)

The design of an architecture with predictable behaviour will be pursued in the PREDATOR project in close collaboration of USAAR with ETHZ and Bologna and steered by requirements of the major industrial players in the aeronautics and automotive domains Airbus and Bosch. This architecture will be a multi-core architecture following the principle of de-sharing, i.e. of keeping non-shared parts of applications non-shared when mapped to the target architecture.

Predictability requirements for the scheduling level will be also considered. In particular, the comparison of preemptive and non-preemptive scheduling disciplines will be performed in collaboration with SSSA.

Mechanisms that support the timing analysis in a hierarchical development processes for contemporary embedded system architectures will be explored. The goal of this investigation will be to work out software structures, hardware mechanisms, and appropriate strategies for the use and control of these hardware mechanisms to provide a system timing behaviour that is both predictable and composable.

The problem of analyzing explicitly parallel programs (as opposed to sequential code running on single cores in a multi-core environment) with respect to maximal execution time will be tackled. Initial theory and formal timing models will be built.

3.2 Current and Future Milestones

Year 1: PROMPT design principles

The PROMPT design principles were developed.

1. The PROMPT design principles for predictable architectures were developed, based on the principles of de-sharing.
2. The design principles were developed based on some intuitive assumptions derived from experience with developing timing analyzers for several architectures. These intuitive assumptions were formally defined and proved for the case of cache architectures. For other architectural components we still lack the underlying theory.
3. The PROMPT architectural design principles will be applied to the design of the PREDATOR multi-core architecture resulting in the first version of the design of the PREDATOR architecture.

Year 1: Timing Side Effects

A clean design and analysis process of embedded systems timing is strongly simplified if the temporal planning and the timing analysis can be performed in a hierarchical way, e.g., in a separation of WCET and schedulability analysis, of single-core analysis from the analysis of the whole multi-core system. Unfortunately many features found in contemporary hardware and software cause phenomena that inhibit a hierarchical decomposition of the planning and analysis process. It is our goal to investigate into these phenomena and name the unwanted side effects they have on timing. The results will be used to work out alternative hardware and software structures that provide better support to a hierarchical design and analysis of worst-case timing.

We have identified various types of side effects, ranging from data-dependent instruction execution times and data paths, over processor-state dependencies between different invocations of a single task, to the side effects that the execution of a task has on the timing of other tasks, where other tasks may reside on the same processor (in a single or multi-processor system) or a different processor of a multi-processor system. The findings of this investigation have been published and presented at the Euromicro workshop on worst-case execution-time analysis, WCET2008.

Year 2: Composable Software/Hardware Architectures

Temporal composability is essential for making the factor time an integral part of a meaningful hierarchical development process for embedded real-time applications. Our goal is to work out software structures, hardware mechanisms, and appropriate strategies for the use and control of these hardware mechanisms that support temporal composability.

We have started work towards this goal in year1 and some ideas on how to obtain composability have been sketched in a WCET2008 paper. In following period of ArtistDesign we will explore the topic further. We plan to have a more detailed report on ways to achieve temporal composability by the end of year2.

Year 2: Initial Timing Model for Parallel Programs

We plan to define a formal timing model for explcitly parallel programs. This model will serve as a “reference point”, and the correctness of forthcoming timing analyses for such programs will be possible to verify with respect to this model.

3.3 Main Funding

- **AVACS**
  The AVACS project *(Automatic Verification and Analysis of Complex Systems)* is funded by the DFG and supports Saarland University.
  http://avacs.org
- **PREDATOR**
  The PREDATOR project *(Design for Predictability and Efficiency)* is a research project within the European Commission’s 7th Framework Programme on Research, Technological Development and Demonstration. Bologna, Pisa, ETHZ, Saarland University and Dortmund take part in PREDATOR, along with AbsInt, Airbus and Bosch. The goal of PREDATOR is to reconcile performance and predictability, in particular for future multi-core architectures.
  http://predator-project.eu
- **COSTA**
  The COSTA project (*Compiler-Support for Timing Analysis*) is funded by the Austrian Science Fund (FWF) and supports TU Vienna. The project started in July 2006.
  http://costa.tuwien.ac.at/
- **FORTAS**
  The FORTAS project (Formales Zeitanalyseframework für Echtzeitsysteme) is a cooperative project that brings together the orthogonal expertise of the TU Vienna real-time systems group and the group on formal methods and systems engineering from TU Darmstadt. Within FORTAS TU Vienna is supported by the Austrian Science Fund (FWF). FORTAS started in January 2007.
  http://www.fortastic.net

- Mälardalen has funding from the KK-foundation (grant 2005/0271, duration 2006 – 2008), and the Foundation for Strategic Research (PROGRESS Strategic Research Centre, duration 2006 – 2010). 2009-2011, Mälardalen will additionally have a grant from the Swedish Research Council "Worst-Case Execution Time Analysis of Parallel Systems".

4. Internal Reviewers for this Deliverable

Jan Gustafsson, MDH
Radu Dobrin, MDH