

# 7th International Workshop on Worst-Case Execution Time Analysis

WCET 2007, July 3, 2007, Pisa, Italy

Edited by

Christine Rochange



*Editor*

Christine Rochange  
TRACE Group  
IRIT, Université Paul Sabatier  
118 Route de Narbonne  
31062 Toulouse Cedex 9, France rochange@irit.fr

*ACM Classification 1998*

C.4 Performance of Systems, D.2.4 Software/Program Verification

**ISBN 978-3-939897-05-7**

*Published online and open access by*

Schloss Dagstuhl – Leibniz-Center for Informatics GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany.

*Publication date*

November, 2007.

*Bibliographic information published by the Deutsche Nationalbibliothek*

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at <http://dnb.d-nb.de>.

*License*

This work is licensed under a Creative Commons Attribution-Noncommercial-No Derivative Works license: <http://creativecommons.org/licenses/by-nc-nd/3.0/legalcode>.



In brief, this license authorizes each and everybody to share (to copy, distribute and transmit) the work under the following conditions, without impairing or restricting the author's moral rights:

- Attribution: The work must be attributed to its authors.
- Noncommercial: The work may not be used for commercial purposes.
- No derivation: It is not allowed to alter or transform this work.

The copyright is retained by the corresponding authors.

Digital Object Identifier: 10.4230/OASlcs.WCET.2007.i

ISBN 978-3-939897-05-7

ISSN 2190-6807

<http://www.dagstuhl.de/oasics>

## OASlcs – OpenAccess Series in Informatics

OASlcs aims at a suitable publication venue to publish peer-reviewed collections of papers emerging from a scientific event. OASlcs volumes are published according to the principle of Open Access, i.e., they are available online and free of charge.

**ISSN 2190-6807**

**[www.dagstuhl.de/oasics](http://www.dagstuhl.de/oasics)**

**2007 WCET Abstracts Collection**  
**7th Intl. Workshop on Worst-Case Execution**  
**Time (WCET) Analysis**

Christine Rochange

Université Paul Sabatier, F  
rochange@irit.fr

**Abstract.** The workshop on Worst-Case Execution Time Analysis is a satellite event to the annual Euromicro Conference on Real-Time Systems. It brings together people that are interested in all aspects of timing analysis for real-time systems. In the 2007 edition, 13 papers were presented, organized into four sessions: methods for WCET computation, low-level analysis, system-level analysis and flow-analysis. The workshop was also the opportunity to report from the 2006 WCET tool challenge.

**Keywords.** Worst-case execution time, real-time systems, timing analysis

**2007 WCET Preface – Proceedings of the 7th Intl.  
Workshop on Worst-Case Execution Time Analysis  
(WCET’07)**

*Jan Gustafsson*

The purpose of the WCET Tool Challenge is to be able to study, compare and discuss the properties of different WCET tools and approaches, to define common metrics, and to enhance the existing benchmarks. The WCET Tool Challenge has been designed to find a good balance between openness for a wide range of analysis approaches, and specific participation guidelines to provide a level playing field. This should make results transparent and facilitate friendly competition among the participants. This short report presents conclusions from the WCET Tool Challenge 2006 as well as some ideas for the WCET Tool Challenge 2006.

*Keywords:* WCET’07, workshop proceedings, abstracts collection

*Full Paper:* <http://drops.dagstuhl.de/opus/volltexte/2007/1198>

## Automatic Amortised Worst-Case Execution Time Analysis

*Christoph A. Herrmann; Armelle Bonenfant; Kevin Hammond; Steffen Jost; Hans-Wolfgang Loidl; Robert Pointon*

Our research focuses on formally bounded WCET analysis, where we aim to provide absolute guarantees on execution time bounds. In this paper, we describe how amortisation can be used to improve the quality of the results that are obtained from a fully-automatic and formally guaranteed WCET analysis, by delivering analysis results that are parameterised on specific input patterns and which take account of relations between these patterns. We have implemented our approach to give a tool that is capable of predicting execution costs for a typical embedded system development platform, a Renesas board with a Renesas M32C/85U processor. We show that not only is the amortised approach applicable in theory, but that it can be applied automatically to yield good WCET results.

*Keywords:* Amortisation, functional programming, performance measurement, static analysis, type and effect systems, worst-case execution time

*Full Paper:* <http://drops.dagstuhl.de/opus/volltexte/2007/1187>

## Clustering Worst-Case Execution Times for Software Components

*Johan Fredriksson; Thomas Nolte; Andreas Ermedahl; Mikael Nolin*

For component-based systems, classical techniques for Worst-Case Execution Time (WCET) estimation produce unacceptable overestimations of a components WCET. This is because software components more general behavior, required in order to facilitate reuse. Existing tools and methods in the context of Component-Based Software Engineering (CBSE) do not yet adequately consider reusable analyses. We present a method that allows different WCETs to be associated with subsets of a components behavior by clustering WCETs with respect to behavior. The method is intended to be used for enabling reusable WCET analysis for reusable software components. We illustrate our technique and demonstrate its potential in achieving tight WCET-estimates for components with rich behavior.

*Keywords:* Worst-case execution time, Software components, Reuse, Analysis

*Full Paper:* <http://drops.dagstuhl.de/opus/volltexte/2007/1185>

## Measurements or Static Analysis or Both?

*Stefan M. Petters; Patryk Zadarnowski; Gernot Heiser*

To date, measurement-based WCET analysis and static analysis have largely been seen as being at odds with each other. We argue that instead they should be considered complementary, and that the combination of both represents a promising approach that provides benefits over either individual approach. In this paper we discuss in some detail how we aim to improve on our probabilistic measurement-based technique by adding static cache analysis. Specifically we are planning to make use of recent advances within the functional languages research community. The objective of this paper is not to present finished or almost finished work. Instead we hope to trigger discussion and solicit feedback from the community in order to avoid pitfalls experienced by others and to help focus our research.

*Keywords:* Measurement based Approach, Static Analysis, Cache Analysis, Proof, Overestimation

*Full Paper:* <http://drops.dagstuhl.de/opus/volltexte/2007/1188>

## A Framework for Static Analysis of VHDL Code

*Marc Schlickling; Markus Pister*

Software in real time systems underlies strict timing constraints. These are among others hard deadlines regarding the worst-case execution time (WCET) of the application. Thus, the computation of a safe and precise WCET is a key issue for validating the behavior of safety-critical systems, e.g. the flight control system in avionics or the airbag control software in the automotive industry. Saarland University and AbsInt Angewandte Informatik GmbH have developed a successful approach for computing the WCET of a task. The resulting tool, called aiT, is based on the abstract interpretation [3, 4] of timing models of the processor and its periphery. Such timing models are hand-crafted and therefore error-prone. Additionally the modeling requires a hard engineering effort, so that the development process is very time consuming. Because modern processors are synthesized from a formal hardware specification, e.g., in VHDL or VERILOG, the hand-crafted timing model can be developed by manually analyzing the processor specification. Due to the complexity of this step, there is a need for support tools that ease the creation of analyzes on such specifications. This paper introduces the primer work on a framework for static analyzes on VHDL.

*Keywords:* Timing Analysis, Worst-Case Execution Time, VHDL, Static Analysis

*Full Paper:* <http://drops.dagstuhl.de/opus/volltexte/2007/1189>

## Towards Symbolic State Traversal for Efficient WCET Analysis of Abstract Pipeline and Cache Models

*Stephan Wilhelm; Björn Wachter*

Static program analysis is a proven approach for obtaining safe and tight upper bounds on the worst-case execution time (WCET) of program tasks. It requires an analysis on the microarchitectural level, most notably pipeline and cache analysis. In our approach, the integrated pipeline and cache analysis operates on sets of possible abstract hardware states. Due to the growth of CPU complexity and the existence of timing anomalies, the analysis must handle an increasing number of possible abstract states for each program point. Symbolic methods have been proposed as a way to reduce memory consumption and improve runtime in order to keep pace with the growing hardware complexity. This paper presents the advances made since the original proposal and discusses a compact representation of abstract caches for integration with symbolic pipeline analysis.

*Keywords:* WCET, worst-case execution time, hard real-time, embedded systems, abstract interpretation, pipeline analysis, cache analysis, symbolic state traversal BDD

*Full Paper:* <http://drops.dagstuhl.de/opus/volltexte/2007/1190>

## Finding DU-Paths for Testing of Multi-Tasking Real-Time Systems using WCET Analysis

*Daniel Sundmark; Anders Petterson; Christer Sandberg; Andreas Ermedahl; Henrik Thane*

Memory corruption is one of the most common software failures. For sequential software and multi-tasking software with synchronized data accesses, it has been shown that program faults causing memory corruption can be detected by analyzing the relations between defines and uses of variables (DU-based testing). However, such methods are insufficient in preemptive systems, since they lack the ability to detect inter-task shared variable dependencies. In this paper, we propose the use of a system level shared variable DU analysis of preemptive multi-tasking real-time software. By deriving temporal attributes of each access to shared data using WCET analysis, and combining this information with the real-time schedule information, our method also detects inter-task shared variable dependencies. The paper also describes how we extended the SWEET tool to derive these temporal attributes.

*Keywords:* Testing, Real-time systems, WCET analysis, data flow

*Full Paper:* <http://drops.dagstuhl.de/opus/volltexte/2007/1191>

## Timing Analysis of Body Area Network Applications

*Liang Yun; Abhik Roychoudhury; Tulika Mitra*

Body area network (BAN) applications have stringent timing requirements. The timing behavior of a BAN application is determined not only by the software complexity, inputs, and architecture, but also by the timing behavior of the peripherals. This paper presents systematic timing analysis of such applications, deployed for health-care monitoring of patients staying at home. This monitoring is used to achieve prompt notification of the hospital when a patient shows abnormal vital signs. Due to the safetycritical nature of these applications, worst-case execution time (WCET) analysis is extremely important.

*Keywords:* WCET analysis of Peripherals, Body Area Network applications

*Full Paper:* <http://drops.dagstuhl.de/opus/volltexte/2007/1192>

## Data-Flow Based Detection of Loop Bounds

*Christoph Cullmann; Florian Martin*

To calculate the WCET of a program, safe upper bounds on the number of loop iterations for all loops in the program are needed. As the manual annotation of all loops with such bounds is difficult and time consuming, the WCET analyzer aiT originally developed by Saarland University and AbsInt GmbH uses static analysis to determine the needed bounds as far as possible. This paper describes a novel data-flow based analysis for aiT to calculate the needed loop bounds on the assembler level. The new method is compared with a pattern based loop analysis already in use by this tool.

*Keywords:* WCET analysis, loop bound detection, flow analysis

*Full Paper:* <http://drops.dagstuhl.de/opus/volltexte/2007/1193>

## Loop Bound Analysis based on a Combination of Program Slicing, Abstract Interpretation, and Invariant Analysis

*Andreas, Ermedahl; Christer Sandberg; Jan Gustafsson; Stefan Bygde; Björn Lisper*

Static Worst-Case Execution Time (WCET) analysis is a technique to derive upper bounds for the execution times of programs. Such bounds are crucial when designing and verifying real-time systems. A key component for static derivation of precise WCET estimates is upper bounds on the number of times different loops can be iterated. In this paper we present an approach for deriving upper loop bounds based on a combination of standard program analysis techniques.



The idea is to bound the number of different states in the loop which can influence the exit conditions. Given that the loop terminates, this number provides an upper loop bound. An algorithm based on the approach has been implemented in our WCET analysis tool SWEET. We evaluate the algorithm on a number of standard WCET benchmarks, giving evidence that it is capable to derive valid bounds for many types of loops.

*Keywords:* WCET analysis, loop-bound analysis, program slicing, abstract interpretation, invariant analysis

*Full Paper:* <http://drops.dagstuhl.de/opus/volltexte/2007/1194>

## **Analysing Switch-Case Tables by Partial Evaluation**

*Niklas Holsti*

This paper describes ongoing work aimed at the construction of formal cost models and analyses to yield verifiable guarantees of resource usage in the context of real-time embedded systems. Our work is conducted in terms of the domain-specific language Hume, a language that combines functional programming for computations with finitestate automata for specifying reactive systems. We outline an approach in which high-level information derived from source-code analysis can be combined with worst-case execution time information obtained from high quality abstract interpretation of low-level binary code.

*Keywords:* WCET, switch-case, partial evaluation

*Full Paper:* <http://drops.dagstuhl.de/opus/volltexte/2007/1195>

## **Analysis of path exclusion at the machine code level**

*Ingmar Stein; Florian Martin*

We present a method to find static path exclusions in a control flow graph in order to refine the WCET analysis. Using this information, some infeasible paths can be discarded during the ILP-based longest path analysis which helps to improve precision. The new analysis works at the assembly level and uses the Omega library to evaluate Presburger formulas.

*Keywords:* Flow-constraint, control flow graph, path exclusion

*Full Paper:* <http://drops.dagstuhl.de/opus/volltexte/2007/1196>

## WCET Analysis: The Annotation Language Challenge

*Raimund Kirner; Jens Knoop; Adrian Prantl; Markus Schordan; Ingomar Wenzel*

Worst-case execution time (WCET) analysis is indispensable for the successful design and development of systems, which, in addition to their functional constraints, have to satisfy hard real-time constraints. The expressiveness and usability of annotation languages, which are used by algorithms and tools for WCET analysis in order to separate feasible from infeasible program paths, have a crucial impact on the precision and performance of these algorithms and tools. In this paper, we thus propose to complement the WCET tool challenge, which has recently successfully been launched, by a second closely related challenge: the WCET annotation language challenge. We believe that contributions towards mastering this challenge will be essential for the next major step of advancing the field of WCET analysis.

*Keywords:* Worst-case execution time analysis, WCET, path description, annotation language challenge, expressiveness, convenience

*Full Paper:* <http://drops.dagstuhl.de/opus/volltexte/2007/1197>