

# TuBound - A Tool for Worst-Case Execution Time Analysis\*

Adrian Prantl

Institute of Computer Languages

TU Vienna, Austria

adrian@complang.tuwien.ac.at

## Abstract

In this talk, we introduce *TuBound*, a tool to perform worst-case execution time (WCET) analysis. From the user's perspective, TuBound is unique in that it operates on the source code level, which opens new possibilities for the usability and flexibility of the tool. In particular, TuBound supports high-level loop optimizations and the automatic analysis of loop bounds. The tool will be demonstrated to a broader audience for the first time at the WCET Tool Challenge 2008 [7].

TuBound is created by integrating several components that were developed independently of each other. The connecting glue between the components is SATIrE, which is also developed at TU Vienna [5]. SATIrE enables using data flow analyzers specified with AbsInt's Program Analyzer Generator (PAG) [4] together with the C++ infrastructure of the ROSE compiler [2]. Using PAG, we implemented an interval analysis for TuBound. Further, we implemented our own term-based loop bound analyzer with the help of the Termite library [6]. Termite is a Prolog library we developed to operate on the external term representation of the abstract syntax tree that SATIrE can export and import. We are using a customized unparser to generate annotated source code. This source code is compiled by a modified version of the GNU C compiler, emitting annotated assembler code for the Infineon C167 microcontroller. The annotated assembler code is the input of the CalcWCET167 tool [1] that performs the timing analysis for the C167 hardware.

TuBound is the result of a cooperation between the Institute of Computer Languages and the Institute of Computer Engineering at the Vienna University of Technology that is carried out as part of the CoSTA[3] project.

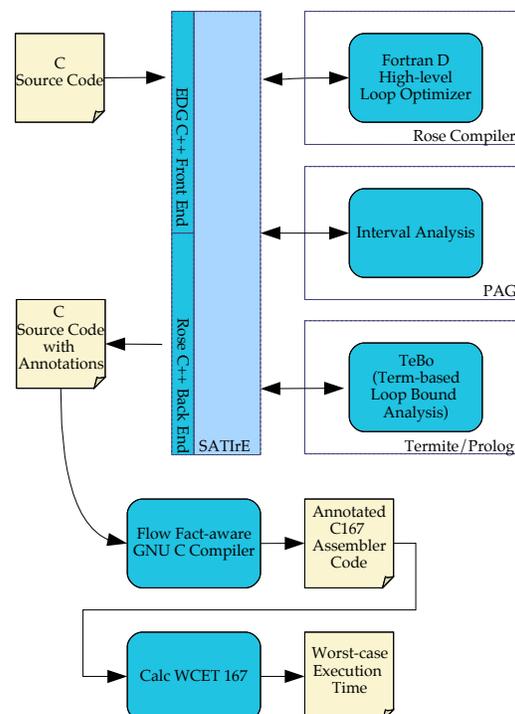


Figure 1: The collaboration of TuBound's components

## References

- [1] CalcWCET167. <http://www.vmars.tuwien.ac.at/~raimund/calc.wcet/>.
- [2] The ROSE compiler. <http://www.rosecompiler.org/>.
- [3] The CoSTA project. <http://costa.tuwien.ac.at/>.
- [4] The Program Analyzer Generator (PAG). <http://www.absint.com/pag/>.
- [5] The Static Analysis Tool Integration Engine SATIrE. <http://www.complang.tuwien.ac.at/markus/satire/>.
- [6] The term manipulation library Termite. <http://www.complang.tuwien.ac.at/adrian/termite/>.
- [7] The WCET tool challenge 2008. <http://www.mrtc.mdh.se/projects/WCC08/>.

\*This work has been partially supported by the Austrian Science Fund (Fonds zur Förderung der wissenschaftlichen Forschung) under contract P18925-N13 and the ARTIST2 Network of Excellence, <http://www.artist-embedded.org/>.