TEACHING WCET ANALYSIS IN ACADEMIA AND INDUSTRY: PANEL DISCUSSION

Niklas Holsti (ed.)[1]

1. Introduction

The last item on the programme of the WCET'09 workshop was a panel discussion on "Teaching WCET analysis in academia and industry". The panelists presented three position statements to initiate a general discussion of the subject. This summary is based on the text of the position statements that the panelists gave me, in my role as the panel chair, on the panelists' presentations, and on my notes of the discussion.

The four experts who kindly agreed to form the panel (of whom Reinhard Wilhelm regrettably could not attend the workshop) span the field from academic research in program analysis to commercial developers and providers of WCET tools. Both static analysis methods and measurement-based methods are represented. However, the end users of WCET analysis tools are not directly represented.

As panel chair I asked the panelists to address some or all of the following questions in their position statements:

- Place, time, and role of WCET analysis in the academic curriculum for real-time systems.
- Contrasts between teaching WCET analysis in general to students, and teaching or training professionals to use a specific WCET tool.
- Among the current professional users of WCET analysis, was anyone taught WCET analysis while a student?
- Common misconceptions and mistakes about WCET analysis and WCET tools, among students and professionals, and how to correct them.
- Which issues in WCET analysis are important? Differences between the academic and industrial assumptions and points of view.
- Stupid questions that students and users ask – and are they really so stupid, or are they deep?
- Teaching materials: texts, presentations, exercises, bibliographies. Can they be shared?
- Making WCET analysis and WCET tools more teachable and understandable: ideas and directions.

Section 2 of this summary presents the panelists' position statements and section 3 summarises the ensuing discussion.

1 Tidorum Ltd, Tiirasaarentie 32, FI 00200 Helsinki, Finland

ECRTS 2009
9th International Workshop on Worst-Case Execution Time (WCET) Analysis
http://drops.dagstuhl.de/opus/volltexte/2009/2278

2. Panelist Position Statements

2.1 Peter Puschner (Vienna University of Technology)

The Vienna University of Technology teaches WCET analysis in an optional course that consists of 15 hours of lectures and at least 10 hours of laboratory exercises. A course in Real-Time Systems is a prerequisite; that course is held by Hermann Kopetz and focuses mainly on the Time-Triggered Architecture (TTA). The goals of the WCET-analysis course are to show that WCET analysis is not trivial; to correct misconceptions of WCET analysis; and to show how to achieve predictability of real-time software. The laboratory exercises use a processor simulator and a scheduling simulator. The course covers pure WCET analysis, the relationship of WCET and scheduling (real-time versus non-real-time), and writing time-predictable, WCET-oriented code. The course is taken by about eight students each year, in their fourth or later year of study.

We observe that students are fascinated by this "unknown", multi-faceted field that involves program analysis, programming, programming languages, compilers, computer architectures, and so on. However, "thinking predictability" seems to be very difficult for students, and working with processor simulators is unsatisfactory, except for experimenting with scheduling. On the positive side, our experience shows that WCET-oriented programming can really improve worst-case performance.

The WCET problem is still widely unknown. Moreover, the complexity of WCET analysis and timing analysis in general is under-estimated. For example, why do so few people care about timing interferences, i.e., influences on task timing that are due to the fact that the execution of each task changes the state of shared resources and thus the timing of other tasks? In our WCET course the trainees learn about complex influences on the timing, about (global) timing interferences, and about predictability, both about obstacles to predictability and how to achieve predictability. In conclusion, I suggest that WCET training should be mandatory for all students in Embedded Systems.

2.2 Christian Ferdinand (AbsInt GmbH) and Reinhard Wilhelm (Saarland University)

For AbsInt as a company it is important that students understand that determining safe upper bounds of the worst-case execution time is an interesting and relevant problem in programming real-time applications and that there are solutions for this problem in form of tools (preferably from AbsInt).

WCET research in Saarbrücken started at the Compiler and Programming Languages chair. The underlying principles of program analysis are taught in “Compiler Construction” and “Program Analysis” courses. The specific analysis problems of WCET analysis and especially cache behavior prediction is covered in 2-4 hours. The core course “Embedded Systems” devotes 2 hours to WCET analysis. A subsequent course “Embedded System Design” has a more practical focus. Students learn how (safety-critical) embedded real-time applications are developed with the help of modern development tools. In the practical part the students develop the control software for a *Lego Mindstorm* robot toy car that follows automatically a lane. Used are the *SCADE* Suite of Esterel

Technologies, the real-time operating system *nxtOSEK*, *aiT* WCET analysis tool, and the scheduling analysis tool *Symta/S* of Symtavision.

WCET analysis has been and is being taught at various advanced courses and Summer Schools, ARTIST DESIGN school, Onassis School on Crete. aiT tool usage is covered in a “standard” 2 days course for aiT users. AbsInt also regularly offers presentations and tutorials about aiT at various conferences/trade shows. We higly welcome efforts by universities to teach WCET analysis. Since the focus (underlying theory/tool usage) and available teaching times vary widely, we usually put together a set of slides for each request and provide aiT license keys for our standard targets. Dear teachers, please do not hesitate to contact us (*info@absint.com*).

As reading material, we recommend the chapter on WCET analysis in the *Embedded Systems Handbook*, Taylor & Francis (Ed. R. Zurawski) and/or R. Wilhelm *et al*. “The worst-case execution-time problem-overview of methods and survey of tools” (*Trans. on Embedded Computing Sys*., 7(3):1–53, 2008). Recorded lectures of Reinhard Wilhelm are also available at
http://www.artist-embedded.org/docs/Events/2008/Autrans/Videos/Reinhard_Wilhelm/ and
http://www.forth.gr/onassis/lectures/2008-07-21/programme.htm.

2.3 Guillem Bernat (Rapita Systems Ltd and University of York)

The University of York teaches WCET analysis in an elective module on Real-Time Systems in the last year of degree study. WCET analysis is covered in one or two lecture hours within the total of 18 hours for this module. There was no laboratory work in 2009. The goal is to make the students understand the issues that affect execution-time behaviour; only then can one address the problem of WCET calculation. We strive to impart a general understanding of how to build analysable systems and show how to move away from an "average case execution time" way of thinking towards WCET-analysable designs.

As preliminaries, we assume knowledge of assembly language, processor architecture, scheduling theory (explained during the Real-Time Systems module), and programming languages and program architectures for real-time software. The course does not address any specific target processor. It gives an overview of WCET-analysis techniques and tools, with focus on the RapiTime tool architecture. Some motivational examples are given but detailed examples are left for self-study. So far, we know of no former student of this course who has then used WCET analysis professionally. However, students tell us that the course is useful and has made them realise the complexity of the issues involved in timing analysis.

3. Discussion

The discussion was not recorded, nor were speakers asked to identify themselves, so this summary does not try to attribute comments to specific speakers.

As could be expected for this venue, the need to teach WCET analysis as part of the curriculum for real-time and embedded systems was not questioned. Instead, the discussion had two main foci: the problem of the "average case" mind-set, and finding good examples and laboratory exercises for WCET analysis.

It was generally agreed that WCET analysis is not for early students. Understanding WCET analysis needs background knowledge of processor architectures, assembly languages, real-time systems and scheduling, and perhaps some understanding of program analysis. But the early mainline courses in programming tend to teach students to optimize programs for the average case. The later courses in real-time programming and WCET analysis must then work to erase this average-case mind-set.

It was pointed out that there is one mainline computing domain that has real-time constraints: computer graphics and games. The image rendering algorithms and data structures must be designed to bound the worst-case performance, else the frame-rate can drop suddenly and very observably in some cases, for example when the viewpoint changes. Perhaps some ideas of WCET analysis could be introduced in computer graphics courses.

For examples and laboratory exercises in WCET analysis, work with real devices such as the Lego robotics kit is very motivating, but their fascination can also be distracting. Furthermore, the control systems in real examples are often so robust that a deadline miss has no bad effects, leading at most to a small hiccup in the visible behaviour. This is teaching the wrong lesson! A deadline miss should have a dramatic effect. As an example, a computer-controlled mouse-trap was suggested: the student inserts a finger to trigger the trap; the software detects that the catch is released and must stop the steel before it strikes the finger...