Saving the World

A long and windy road towards sustainability and formal verification in practice

Marko van Eekelen
Full Professor, Head of Department
Computer Science Department
Faculty of Management, Science and Technology; Open University of The Netherlands
Heerlen
Marko.vanEekelen@ou.nl
Associate Professor
Digital Security
Institute for Computing and Information Sciences, Radboud University
Nijmegen
marko@cs.ru.nl

Abstract

My research is spread over two universities, mainly in the following two different research areas:

- *Resource consumption analysis*
- *Formal verification methods for verifying security and correctness in cyber physical systems.*

Keywords formal, practical, sustainable

1 Introduction

1.1 Computer Science Research at the Open University

The OU CS department has an emerging research group within the Netherlands. Only since 2009 de OU formally has a disciplinary research task. Since 2014 the author is chairing the OU Computer Science Department with the intention that research at the Open University is just as important as education. This has lead to a growth of the department to a current size of over 30 members (4 full professors, 1 associate professor, 17 assistant professors, 5 lecturers, 4 postdocs, 13 external Ph.D. students) performing research in 3 focal points:

- Learning (in 3 topics: Tools for Supporting Learning, Computer Science Education, Computer Science Didactics),
- Resilience - Trustworthy Systems (in 2 topics: Verification, Security & Privacy)
- Innovation (Artificial Intelligence, Machine Learning).

The research is embedded in the Faculty of Management, Science and Technology promoting a culture of interdisciplinary research.

The members of the department recently acquired several grants (on Regional, National, European and American level) among which a Rubicon and a Veni grant.

LFM-NL'18, September 03-04, 2018, Leiden, NL
2018.

2 Resource Consumption Analysis

Functional properties of programs are widely studied. It is however less common to study non-functional properties of code. Recently, the resources studied are diversifying [**?**]. In particular, the study of the consumption of other resources than time is an opening field. Studying resources such as memory and energy seems to the most promising [**?**].

From the practical point of view, the results discussed in [**?**] improve polynomial resource analysis of computer programs as presented in [**?**]. There the authors consider the size of output as a polynomial function on the sizes of inputs [**? ?**]. In the NL NWO AHA project (2006-2011), the EU Charter Artemis project (2009-2012) and the NL GoGreen IOP GenCom project (2011-2015) the ResAna tool [**? ? ? ?**] was developed that applies polynomial interpolation to generate an upper bound on Java loop iterations. The tool requires the user to input the degree of the solution. In [**?**] a partial result for that was provided. The results of recent work [**?**] make it possible to automatically obtain the degree of the polynomial in all cases for quadratic algebraic difference equation with constant coefficients.

Building upon this work, the focus moved from size, memory and loop bounds to sustainability of software [**?**] in general and of energy consumption analysis in particular.

2.1 A Moral Appeal

Computer Science is not the most sustainable discipline, to say the least. Every few years new equipment 'has' to be bought. The energy consumption due to digital equipment is seldom an issue. In software development energy consumption is rarely an issue. Instead of paying attention to the sustainability of software in such a way that an important design concern is that during the software life cycle as less as possible energy is consumed, the sole focus seems to be to keep legacy systems running in terms of functionality whatever the influence is on energy consumption.

As a discipline we need to do better with respect to sustainability. In fact, I would like to make a moral appeal for performing research in the area of energy analysis consumption paraphrasing famous words of John F. Kennedy:

"*And so my fellow Formal Method researchers: ask not what the world can do to reduce the energy consumption for you - ask how you can apply Formal Methods to reduce the energy consumption of the world: ask not what other researchers will do for you, but what together we can do for reducing the energy consumption of man.*"

The good news is that interest in energy consumption and in greenIT in the Netherlands is growing, e.g. at the Free University of Amsterdam [**?**], at the Software Improvement Group [**?**], at Utrecht University [**? ?**] and at the Open University.

2.2 Energy Consumption Analysis at the Open University

Building upon practical resource analysis work [**?**] a research track on static analysis of energy consumption. This started with defining a suited Hoare logic that enabled a safely approximating static analysis [**?**]. This resulted in a webtool, ECAlogic [**?**], that made it possible to derive energy consumption bounds for small systems (hardware components controlled by a software application) in a hardware-parametric way. Due to this work the focus of the research changed to analysing IT controlled systems parametrised by hardware finite state machine models [**?**]. The corresponding approach was to focus on systems with multiple components, model the components and analyse the control software to estimate the energy consumption of the system. Using dependent types the analysis was made ready for a practical, precise and parametric energy analysis of IT controlled systems [**?**]. In working towards actual practice a first, small case study revealed that instead of doing a full analysis it can be worthwhile to focus solely on finding energy hot spots and energy bugs [**?**].

The OU memory and energy consumption analysis work was disseminated at the 2013 IPA Winterschool on Software Technology in Eindhoven, at the 2016 EU COST action TACLe Summerschool in Vienna and at the 2017 IPA Fall Days on System and Software Analysis.

2.3 Formal verification methods for verifying security and correctness in cyber physical systems at Radboud Unversity

My Radboud research in formal verification started with work on a dedicated proof assistant for the functional programming language Clean with special support for generic type classes and explicit strictness [**? ? ?**]. In the context of LaQuSO (Laboratory for Quality Software) we were able to verify the core decision algorithm of the Dutch Storm Surge Barrier 'Maeslantkering' protecting the Rotterdam area against flooding. The algorithm was formally specified in Z. We checked the code against the specification and we validated the specification. As a result firstly some minor changes were needed both in the specification and in the code and secondly a scenario popped up from model checking in which the barrier would not close according to the specification while it should close according to the experts [**?**]. Everything was fixed such that the Dutch are saved from 'getting their feet wet'.

Figure 1. Maeslantkering.

Currently, together with Herman Geuvers I am leading the STW Sovereign project (2016-2020) supported by RWS (the Dutch ministry of Transport, Public Works and Water Management) and NRG (the Dutch Nuclear Research Group). The goal of this project is to develop verification techniques for safety critical software based on the following challenging principles. Verification should be (1) scalable (costs should not grow exceedingly as the size of the system increases), (2) compositional (global properties are directly inferable from local properties of the subsystems), (3) incremental (the verification process can be performed iteratively while previous intermediate results are still usable), and (4) effective (the proposed methodology will be applied successfully in some real-world case studies). The fundamental idea of our proposal can be illustrated best with our motto: ÕScalability through modularityÕ. Modularity is commonly recognized as the key for managing complex software systems. With regards to programs, we will elaborate on the concept of design pattern (a description of a solution to a recurring problem) as a modularizing construct. We will investigate both general and security specific design patterns, and develop accompanying proof patterns that simplify the formal verification process. Moreover, as an follow-up of our work on the formalization of the C11 standard, we aim to make an important step in improving the scalability of the C verification process.

3 Moral Discussion

Answer the following questions:

- Is n't it about time that IT starts saving the world instead of consuming it?
- Is n't it about time that IT starts saving the world before it is too late?
- Is n't it about time that designs and implementations of safety critical cyber physical systems are subject to formal verification on a regular basis?

With every 'yes' we contribute to saving the world....