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.*

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.

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 [12]. 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 [22].

From the practical point of view, the results discussed in [16] improve polynomial resource analysis of computer programs as presented in [19]. There the authors consider the size of output as a polynomial function on the sizes of inputs [18, 21]. 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 [10, 15, 20, 25] 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 [16] a partial result for that was provided. The results of recent work [17] 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 [24] 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 [13], at the Software Improvement Group [7], at Utrecht University [5, 6] and at the Open University.

2.2 Energy Consumption Analysis at the Open University

Building upon practical resource analysis work [8] a research track on static analysis of energy consumption. This started with defining a suited Hoare logic that enabled a safely approximating static analysis [9]. This resulted in a webtool, ECAlogic [14], 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 [3]. 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 [4]. 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 [2].

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 [1, 23, 26]. 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 [11]. 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....

References

[1] Maarten de Mol and Marko C. J. D. van Eekelen. 1999. A Proof Tool Dedicated to Clean - The First Prototype. In *Applications of Graph Transformations with Industrial Relevance, International Workshop, AGTIVE'99, Kerkrade, The Netherlands, September 1-3, 1999, Proceedings.* 271–278. https://doi.org/10.1007/3-540-45104-8_22

[2] Pascal van Gastel, Bernard van Gastel, and Marko van Eekelen. 2018. Detecting Energy Bugs and Hotspots in Control Software Using Model Checking. In *Conference Companion of the 2Nd International Conference on Art, Science, and Engineering of Programming (Programming Along the System Stack '18 Companion).* ACM, New York, NY, USA, 93–98. https://doi.org/10.1145/3191697.3213805

[3] Bernard van Gastel, Rody Kersten, and Marko van Eekelen. 2016. Using dependent types to define energy augmented semantics of programs. In *Proceedings of the Fourth International Workshop on Foundational and Practical Aspects of Resource Analysis (FOPARA'15) (LNCS)*, Vol. 9964. Springer, 1–20. https://doi.org/10.1007/978-3-319-46559-3_2

[4] Bernard van Gastel and Marko van Eekelen. 2017. Towards practical, precise and parametric energy analysis of IT controlled systems. In *Proceedings of the Fifth International Workshop on Foundational and Practical Aspects of Resource Analysis (FOPARA'17).*

[5] Erik Jagroep, Jordy Broekman, Jan Martijn E. M. van der Werf, Patricia Lago, Sjaak Brinkkemper, Leen Blom, and Rob van Vliet. 2017. Awakening Awareness on Energy Consumption in Software Engineering. In *39th IEEE/ACM International Conference on Software Engineering: Software Engineering in Society Track, ICSE-SEIS 2017, Bueons Aires, Argentina, May 20-28, 2017.* 76–85. https://doi.org/10.1109/ICSE-SEIS.2017.10

[6] Erik Jagroep, Jan Martijn E. M. van der Werf, Slinger Jansen, Miguel Alexandre Ferreira, and Joost Visser. 2015. Profiling energy profilers. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing, Salamanca, Spain, April 13-17, 2015.* 2198–2203. https://doi.org/10.1145/2695664.2695825

[7] Georgios Kalaitzoglou, Magiel Bruntink, and Joost Visser. 2014. A Practical Model for Evaluating the Energy Efficiency of Software Applications. In *ICT for Sustainability 2014 (ICT4S-14), Stockholm, Sweden, August 25, 2014.* https://doi.org/10.2991/ict4s-14.2014.9

[8] Rody Kersten, Olha Shkaravska, Bernard van Gastel, Manuel Montenegro, and Marko van Eekelen. 2012. Making resource analysis practical for real-time Java. In *Proceedings of the 10th International Workshop on Java Technologies for Real-time and Embedded Systems (JTRES) (JTRES'12)*, Martin Schoeberl and Andy J. Wellings (Eds.). ACM, New York, NY, USA, 135–144. https://doi.org/10.1145/2388936.2388959

[9] Rody W.J. Kersten, Paolo Parisen Toldin, Bernard E. van Gastel, and Marko C.J.D. van Eekelen. 2014. A Hoare Logic for Energy Consumption Analysis. In *Proceedings of the Third International Workshop on Foundational and Practical Aspects of Resource Analysis (FOPARA'13) (LNCS)*, Vol. 8552. Springer, 93–109. https://doi.org/10.1007/978-3-319-12466-7_6 Referenced in the thesis as [BvG-9], see appendix ??.

[10] Rody W. J. Kersten, Bernard van Gastel, Olha Shkaravska, Manuel Montenegro, and Marko C. J. D. van Eekelen. 2014. ResAna: a resource analysis toolset for (real-time) JAVA. *Concurrency and Computation: Practice and Experience* 26, 14 (2014), 2432–2455. https://doi.org/10.1002/cpe.3154

[11] Ken Madlener, Sjaak Smetsers, and Marko C. J. D. van Eekelen. 2010. A Formal Verification Study on the Rotterdam Storm Surge Barrier. In *Formal Methods and Software Engineering - 12th International Conference on Formal Engineering Methods, ICFEM 2010, Shanghai, China, November 17-19, 2010. Proceedings.* 287–302. https://doi.org/10.1007/978-3-642-16901-4_20

[12] Reinhard Wilhelm Florian Zuleger Marco Gaboardi, Jan Hoffmann (Ed.). 2018. Resource Bound Analysis: Report from Dagstuhl Seminar 17291. *Dagstuhl Reports* (2018).

[13] Fahimeh Alizadeh Moghaddam, Patricia Lago, and Iulia Cristina Ban. 2018. Self-adaptation approaches for energy efficiency: a systematic literature review. In *Proceedings of the 6th International Workshop on Green and Sustainable Software, GREENS@ICSE 2018, Gothenburg, Sweden, May 27, 2018.* 35–42. https://doi.org/10.1145/3194078.3194084

[14] Marc Schoolderman, Jascha Neutelings, Rody W.J. Kersten, and Marko C.J.D. van Eekelen. 2014. ECAlogic: Hardware-parametric Energy-consumption Analysis of Algorithms. In *Proceedings of the 13th Workshop on Foundations of Aspect-oriented Languages (FOAL'14).* ACM, New York, NY, USA, 19–22. https://doi.org/10.1145/2588548.2588553

[15] Olha Shkaravska, Rody Kersten, and Marko C. J. D. van Eekelen. 2010. Test-based inference of polynomial loop-bound functions. In *Proceedings of the 8th International Conference on Principles and Practice of Programming in Java, PPPJ 2010, Vienna, Austria, September 15-17, 2010*, Andreas Krall and Hanspeter Mössenböck (Eds.). ACM, 99–108. https://doi.org/10.1145/1852761.1852776

[16] O. Shkaravska and M. van Eekelen. 2014. Univariate polynomial solutions of algebraic difference equations. *Journal of Symbolic Computation* 60 (2014), 15 – 28. https://doi.org/10.1016/j.jsc.2013.10.010

[17] O. Shkaravska and M. van Eekelen. 2018. Polynomial solutions of algebraic difference equations and homogeneous symmetric polynomials. *Journal of Symbolic Computation* (2018). Under Submission.

[18] Olha Shkaravska, Marko C. J. D. van Eekelen, and Alejandro Tamalet. 2013. Collected Size Semantics for Strict Functional Programs over General Polymorphic Lists. In *Foundational and Practical Aspects of Resource Analysis - Third International Workshop, FOPARA 2013, Bertinoro, Italy, August 29-31, 2013, Revised Selected Papers (Lecture Notes in Computer Science)*, Ugo Dal Lago and Ricardo Peña (Eds.), Vol. 8552. Springer, 143–159. https://doi.org/10.1007/978-3-319-12466-7_9

[19] Olha Shkaravska, Marko C. J. D. van Eekelen, and Ron van Kesteren. 2009. Polynomial Size Analysis of First-Order Shapely Functions. *Logical Methods in Computer Science* 5, 2 (2009). http://arxiv.org/abs/0902.2073

[20] Olha Shkaravska, Ron van Kesteren, and Marko C. J. D. van Eekelen. 2007. Polynomial Size Analysis of First-Order Functions. In *Typed Lambda Calculi and Applications, 8th International Conference, TLCA 2007, Paris, France, June 26-28, 2007, Proceedings (Lecture Notes in Computer Science)*, Simona Ronchi Della Rocca (Ed.), Vol. 4583. Springer, 351–365. https://doi.org/10.1007/978-3-540-73228-0_25

[21] Alejandro Tamalet, Olha Shkaravska, and Marko C. J. D. van Eekelen. 2008. Size Analysis of Algebraic Data Types. In *Proceedings of the Nineth Symposium on Trends in Functional Programming, TFP 2008, Nijmegen, The Netherlands, May 26-28, 2008. (Trends in Functional Programming)*, Peter Achten, Pieter W. M. Koopman, and Marco T. Morazán (Eds.), Vol. 9. Intellect, 33–48.

[22] Marko van Eekelen. 2018. ECA: Energy Consumption Analysis of software controlled systems, In Resource Bound Analysis: Report from Dagstuhl Seminar 17291, Reinhard Wilhelm Florian Zuleger Marco Gaboardi, Jan Hoffmann (Ed.). *Dagstuhl Reports*, 84.

[23] Marko C. J. D. van Eekelen and Maarten de Mol. 2005. Proof Tool Support for Explicit Strictness. In *Implementation and Application of Functional Languages, 17th International Workshop, IFL 2005, Dublin, Ireland, September 19-21, 2005, Revised Selected Papers.* 37–54. https://doi.org/10.1007/11964681_3

[24] Bernard van Gastel. 2016. *Assessing sustainability of software.* Ph.D. Dissertation. Open University of the Netherlands.

[25] Ron van Kesteren, Olha Shkaravska, and Marko C. J. D. van Eekelen. 2008. Inferring Static Non-monotone Size-aware Types Through Testing. *Electr. Notes Theor. Comput. Sci.* 216 (2008), 45–63. https://doi.org/10.1016/j.entcs.2008.06.033

[26] Ron van Kesteren, Marko C. J. D. van Eekelen, and Maarten de Mol. 2004. Proof support for generic type classes. In *Revised Selected Papers from the Fifth Symposium on Trends in Functional Programming, TFP 2004, München, Germany, 25-26 November 2004.* 1–16.