Open Universiteit

REPORT

RESEARCH STRATEGY DEPARTMENT OF COMPUTER SCIENCE

TOWARDS HIGH-QUALITY AND INTELLIGENT SOFTWARE

2020-2025

Open University of the Netherlands
Faculty of Science

Heerlen, September 2020

Towards High-quality and Intelligent Software

Towards High-quality and Intelligent Software

INTRODUCTION

This chapter describes the research program of the Department of Computer Science (CS) at the Open Universiteit (OU) for 2020-2025. Its research mission is to create an appealing and lively research environment, leading to a high(er) research quality, productivity, and international visibility.

The research in the Department of Computer Science is for a large part aiming at improving quality assurance technology for software engineering. It is composed of four program lines:

1 **Software Quality**. This program line focuses on quality assurance techniques for software systems. Research concentrates on techniques for, or based on, automated testing, theorem proving, model checking, run-time verification and static analysis. This program line is led by prof. dr. Tanja E.J. Vos and prof. dr. Marko van Eekelen.
2 **Security & Privacy**. This program line focuses on security of software and computer systems, and privacy-by-design. This program line is led by dr. ir. Harald Vranken.
3 **Artificial Intelligence**. This program line focuses on methods for artificial intelligence (such as Bayesian models, machine learning and deep learning), responsible AI, and applications (smart services). This program line is led by dr. Martijn van Otterlo.
4 **Teaching & Learning**. This program line focuses on learning-supporting tools and on computing education. This program line is led by prof. dr. Erik Barendsen and prof. dr. Johan Jeuring.

The following table provides an overview of the amount of FTE involved in the research program.

Program line	Staff	PhD/postdoc
Software Quality	2.0	1.3
Security & Privacy	1.4	1.7
Artificial Intelligence	0.4	-
Teaching & Learning	1.3	-

In the remainder of this chapter we first position these four program lines in the landscape of Dutch computer science research (section 2.2). Next, we describe the history of the research program of the Department of Computer Science (section 2.3), showing its step-wise evolution since 2011. Finally, in section 2.4 we describe the four program lines in more detail.

1 Computer Science Research in the Netherlands

The Dutch universities that host computer science, created 'Sectorbeeld informatica' in 2018 in which the (desired) computer science research landscape at these universities is outlined. It identifies seven focus areas:
1 Data modelling and analysis. This area focuses on big data and data science.
2 Machine learning. This area focuses on the boundaries of AI and explainable AI.
3 Machine reasoning and interaction. This area focuses on 'AI with the human at the center'.
4 Algorithmics. This area focuses on formally verifiable efficient and correct algorithms and data structures.
5 Software. This area focuses on the development of robust and trustworthy software in changing environments.

6 Security and privacy. This area focuses on the five pillars (design, defense, attacks, governance, and privacy) as described in the National Cyber Security Research Agenda III [13].
7 Networked computing and embedded systems. This area focuses on scalability and consistency of distributed, autonomous systems, also covering embedded systems.

The following table shows how the program lines in the computer science research program at the OU correspond to the focus areas.

Focus area	**Program line**
Data modelling and analysis	-
Machine learning	Artificial Intelligence
Machine reasoning and interaction	Artificial Intelligence
Algorithmics	Software Quality
Software	Software Quality Teaching & Learning
Security and privacy	Security & Privacy
Networked computing and embedded systems	-

Each program line corresponds closely with one or two focus areas. The focus area on Data modelling and analysis is not covered by the program lines of the Department of Computer Science, but is covered by research of the Department of Information Science and Business Processes at the OU. The focus area on Software is covered by the program lines Software Quality and Teaching & Learning. The latter has a clear focus on education, and also considers the application of software technology and programming languages, the development of a generic software framework, and the study of problem domains related to programming. The focus area Networked computing and embedded systems is not covered by the program lines, although some aspects are addressed, in particular research on sustainability and energy analysis, in the Software Quality line.

2 Computer science research at the OU

Traditionally, research at the OU was targeted at educational science. Since the renewed 'Wet op het hoger onderwijs en wetenschappelijk onderzoek' in 2009, the OU has the explicit task to also address research within the faculties, and to integrate research in the bachelor and master programs. Since then, the computer science research effort has been increasing steadily.

The first research program on computer science was created by the School of Computer Science at the OU in May 2011: Software Technology Research Plan 2010-2015 [44]. This research program contained two research lines on software technology: Software Technology for Teaching and Learning and Software Technology for Quality Improvement.

In 2014 the School of Computer Science was integrated in the Faculty of Management, Science & Technology (MST), and renamed into Department of Computer Science. The MST research committee created an interdisciplinary research program in December 2014: Learning and Innovation in Resilient Systems: MST Research Program 2015-2020 [41]. This MST research program organized and integrated the ongoing research activities in the disciplines of management science, environmental science, and computer science. The MST research program was composed of three research lines:

1 **Resilience.** This research program is aiming to increase our understanding of the capacity of systems to absorb disturbance and reorganize while undergoing change so as to still retain essentially the same function, consisting of the three subprograms Sustainability, Services and Supply Chains, and Trustworthy Systems.

2 **Learning.** This research program is aiming to improve our understanding of learning by individuals and organizations, consisting of the three subprograms Teaching and Social Learning, Learning Organizations and Interorganizational Learning, and Learning Regions.

3 **Innovation.** This research program is aiming to enhance our knowledge of how innovations emerge, diffuse and impact the world, and the role of agency therein, consisting of the three subprograms Innovative Systems, Diffusion of Innovations, and Impact of Innovations.

The MST research program integrated the two program lines of the Department of Computer Science.

1 The program line *Software technology for quality improvement* was integrated in the *Trustworthy Systems* subprogram of the *Resilience* program. The focus of the *Trustworthy Systems* subprogram is on aspects of resilience that relate to the trustworthiness of software systems and information systems. The focus is on analysis and evaluation of aspects such as correctness and security. These aspects cover both functional properties, that define specific behavior or functions of a system, as well as non-functional properties, that define criteria that can be used to judge the operation of a system. The subprogram includes: research on formal verification of non-functional properties of software such as consumption of resources (energy, memory, processing time); formal verification of security properties such as non-inference in automotive systems; security of computer systems and information systems such as security of on-line banking and smart grids; and security of society such as defense against malicious infrastructures (botnets).

2 The program line *Software technology for teaching and learning* was integrated in the subprogram *Teaching and Social Learning* of the *Learning program*. The focus of the *Teaching and Social Learning* subprogram is on (online) learning, including automated feedback to students, virtual labs, and didactics. Research on automated feedback focuses on the development and evaluation of automated systems that provide feedback to students in an educational setting when working on assignments. Software technology is developed for teaching and learning. It includes the development of tools to support student learning of procedural skills, particularly in the field of mathematics, logic, and programming. In virtual lab environments students can work on practical assignments in computer security and computer networks. This research also covers automated feedback to support students when working on practical assignments in the virtual lab. This subprogram includes research on the didactics for teaching subjects in computer science, as well as on learning by students and the management of higher education.

On October 31, 2017, the MST research program on Learning and Innovation in Resilient Systems was assessed in a midterm review over the period 2014-2016. A self-evaluation of the research program was written [42]. The midterm review followed the SEP 2015-2021 'Protocol for Research Assessments in the Netherlands' (amended version, September 2016). The assessment committee took into account three assessment criteria: research quality; relevance to society; and viability. Furthermore, three additional aspects were considered: PhD programs, research integrity, and diversity. The outcome of the assessment was very good [4].

3 Current research lines

In 2020 the organizational structure of the OU was reorganized. Since then the Department of Computer Science is part of the Faculty of Science. In this section the four program lines of the research program of the Department of Computer Science are described. We describe the key research themes that are covered in the program lines. For each program line, we also mention the staff members of the Department of Computer Science that are active in it (in alphabetic order). Note that some of these researchers are active in multiple program lines, also because there is a lot of collaboration between researchers.

3.1 SOFTWARE QUALITY

The influence of software-intensive systems on our daily lives is continuously growing. Studies show that software failures exist and they have far-reaching implications in terms of money, safety, privacy, etc. Hence, guaranteeing the quality of software is increasingly important. Unfortunately, this does not always happen in practice, and some already go as far as arguing that there are signs of a 'coming software apocalypse' [51]. This research line focuses on quality assurance techniques related to software testing as well as formal methods.

3.1.1 *Software testing*

Our research aims at two things. On the one hand, automating testing as much as possible, by making intelligent testing tools that support and relieve the tester instead of generating more work. On the other hand, by helping the introduction of testing as early as possible, through a shift left of testing in education as well as in software development.

Scriptless test automation

Many companies have not been able to get the desired return of investment from their attempts to automate testing [34, 50]. This is because most frameworks are script-based and designed to automate manual steps. Existing script-based tools are not intelligent enough and hence cannot react to changes in a flexible manner. This has led to significant maintenance effort, particularly at a time when apps are developing at a pace that makes it hard for test teams to keep up [50].

TESTAR [69] is an open source tool that implements a scriptless approach for completely automated test case generation based on agents that implement various action selection mechanisms and test oracles. The underlying principle of this type of testing is very simple: generate test sequences of (state, action)-pairs by starting up the System Under Test (SUT) in its initial state and continuously select an action to bring the SUT in another state. The action selection characterizes the most basic problem of intelligent systems: what to do next. The difficult part is optimizing the action selection to find faults and recognizing a faulty state when it is found. This research will totally shift the paradigm of GUI testing: from developing scripts to developing intelligent AI-enabled agents.

Research directions within this topic consist of:
- Using search-based algorithms, genetic programming and other nature inspired algorithms to optimize action selection
- Reinforcement learning to learn what the best action selection strategy is
- Machine learning to enable derivation of the GUI state from screen-captures
- Dynamic model-inference and learning during testing
- Oracle definitions
- Definition of new test adequacy criteria
- Model-checking of inferred models.

Early shift-left testing

To address the increasing problem of poor software quality, we need a fundamental change in the way programming is learned. Future software builders must be trained with an appreciation for software quality and quality assurance techniques must become a natural aspect of software development, rather than a niche topic.

Programming courses in computer science education, however, do not address testing enough, or introduce it too late. Current courses are usually inclined to concentrate more on the creative aspects of programming and coding. This is of course not surprising. Many techniques to improve the quality of the software (e.g. documenting your code and testing it) can seem tedious and give the feeling of doing 'double work' when introduced too late. Programming education must be inextricably linked to quality and testing.

This requires a fundamental change in the way programming is taught as well as the attitude of teachers and their perception of this subject and the way it is taught.

Research directions within this topic consist of:
- Gamification of test education
- Empirical validation in the classroom of teaching material for early test introduction
- Tools for early, shift-left, model-based test education.

Our research on software testing is seamlessly integrated with our education programs, including a dedicated course on software testing in the bachelor and over twenty bachelor and master theses in recent years (2017–2019).

Facts and figures

People:
- Pekka Aho (senior researcher)
- Harrie Passier (assistant professor)
- Tanja Vos (full professor)
- Nikè van Vugt-Hage (lecturer)
- Olivia Rodriguez (PhD student).

Key collaborators:
- Wishnu Prasetya (UU, The Netherlands)
- Machiel van der Bijl (Axini, The Netherlands)
- Fabiano Delpiaz (UU, The Netherlands)
- Jan Tretmans (TNO, RU, The Netherlands)
- James Bach (Satisfice, USA)
- Gordon Fraser (University of Passau, Germany)
- Fernando Pastor Ricos (Universidad Politécnica de Valencia, Spain)
- Mark van Helvoort (Philips, The Netherlands)
- Marv van Herwaarden (SOGETI, The Netherlands)
- Luna Yaping (ING, The Netherlands).

Projects and funding:
- TESTOMAT (2017–2020) (672.300 euros, ITEA3)
- IVVES (2019–2022) (936.641 euros, ITEA3)
- IMPRESS (2017–2020) (450.000 euros, ERASMUS)
- SERF (2018–2021)
- UPV collaboration: DECODER (2019–2022) and iv4XR (2019–2022).

Selected publications:
Aho, P., Alégroth, E., Oliveira, R.A.P., & Vos, T.E.J. (2016). Evolution of automated regression testing of software systems through the graphical user interface. In *the First International Conference on Advances in Computation, Communications and Services (ACCSE 2016)*, pages 16–21.

Aho, P., & Vos, T.E.J. (2018). Challenges in automated testing through graphical user interface. In *2018 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pages 118– 121, Los Alamitos, CA, USA. IEEE Computer Society.

Aho, P., Vos, T.E.J., Ahonen, S., Piirainen, T., Moilanen, P., & Ricos, F.P. (2019). Continuous piloting of an open source test automation tool in an industrial environment. In *Jornadas de Ingeniera del Software y Bases de Datos (JISBD)*, pages 1–4. Sistedes.

De Gier, F., Kager, D., De Gouw, S., & Vos, T.E.J. (2019). Offline oracles for accessibility evaluation with the testar tool. In *2019 13th International Conference on Research Challenges in Information Science (RCIS)*, pages 1–12.

Esparcia, A.I., Almenar, F., Martnez, M., Rueda, U., & Vos T.E.J. (2016). Q-learning strategies for action selection in the testar automated testing tool. In *6th International Conference on Metaheuristics and nature inspired computing (META 2016)*, pages 130–137.

Esparcia, A.I., Almenar, F., Rueda, U., & Vos T.E.J. (2017). Evolving rules for action selection in automated testing via genetic programming - A first approach. In *Applications of Evolutionary Computation - 20th European Conference, EvoApplications 2017, Amsterdam, The Netherlands, April 19-21, 2017, Proceedings, Part II*, pages 82–95.

Esparcia, A.I., Almenar, F., Vos, T.E.J., & Rueda, U. (2018). Using genetic programming to evolve action selection rules in traversal-based automated software testing: results obtained with the TESTAR tool. *Memetic Computing*, 10(3), 257–265.

Vos, T.E.J., & Aho, P. (2017). Searching for the best test . In *2017 IEEE/ACM 10th International Workshop on Search-Based Software Testing (SBST)*, pages 3–4. *

Vos, T.E.J., Kruse, P.M., Condori-Fernández, N., Bauersfeld, S., & Wegener, J. (2015). TESTAR: Tool support for test automation at the user interface level. *Int. J. Inf. Syst. Model. Des.*, 6(3), 46–83.

3.1.2 *Formal methods (FM)*

Historically, formal methods (FM) tools and algorithms have been available only to expert users in niche areas; not because they cannot be bought or downloaded, but because it requires prohibitively many expertise—and, seemingly, an intimidating amount of knowledge—to effectively deploy them. At the CS department of OU, we aim to make FM accessible to everyone—from the few computer scientists today, to the many software engineers and stakeholders tomorrow—as a key enabler for safe digital infrastructure. Following OU's tradition of servicing society's need in extraordinary ways unlike other Dutch universities, we are pursuing a unique research program towards *FM for the masses*, both in terms of users and impact. Specifically, with dedicated staff and graduate students (both at MSc and PhD level), we are gearing our efforts in four directions.

Formal methods for standard libraries

Standard libraries are among the most commonly (re)used software components and a vital part of the ecosystem of any mainstream programming language. Yet, despite their importance, their safety and correctness are generally an open question.

We aim to develop new compositional techniques for deductive verification of (standard) libraries. Importantly, we are also applying these techniques to verify key classes in the standard library of Java. Through the resulting provably safe and *massively used code*, our vision is to enable *masses of software engineers* to benefit from FM research, unbeknownst to them.

Our initial work in this area—published in CAV 2015 [16], J. Autom. Reasoning 2019 [15], and TACAS 2020 [28]—has been highly promising and, notably, uncovered decades-old bugs in the implementation of Java's sorting algorithm and list implementation.

Formal methods for concurrency

Type systems are arguably the most successful FM technique ever developed. At the same time, concurrent programming is arguably *both* the most error-prone *and* the most highly anticipated—crucial to attain scalability on multicore processors—software engineering activity.
We aim to develop new type-based approaches to facilitate safe concurrent programming. Our unique perspective is that of runtime verification: we combine model-checking algorithms with light-weight instrumentation/monitoring to build runtime systems that can dynamically detect concurrency errors and even deploy recovery mechanisms. By packaging these technologies in programmer-friendly DSLs, hiding all FM behind high-level syntax and intuitive abstractions, our vision is to aid *masses of software engineers* in safely leveraging the *massive potential* of modern hardware resources, without ever exposing them to the underlying FM.

Our initial efforts in this area—published in POPL 2019 [12], ESOP 2020 [29], and TACAS 2020 [24]—are encouraging; this research is partially funded by NWO (Veni).

Formal methods for binaries

From legacy and abandonware to (smart) apps in virtual market places, and from closed-source enterprise applications to embedded software; millions of binaries are being deployed and used daily, whose source code is inaccessible to stakeholders. Problematically, there is essentially nothing stakeholders can do to establish third-party binaries are indeed safe, except for just *believing* the original developers did a thorough job. But faith is brittle, and developers have shown—time and again—inadequacy in this area.

We aim to develop a novel holistic approach to decompilation and verification of binaries, by exploiting synergy between new techniques for reverse-engineering and model-checking; to maximize impact, we are working on tools that target binaries originally written in C++. By placing particular emphasis on automation ("push-button FM"), our vision is to allow *masses of stakeholders* to apply FM to *massively available binaries*.

First results—published in MEMOCODE 2019 [67], SAFECOMP 2019 [8], and TACAS 2020 [66]—have attracted attention, and even funding, from DARPA, as we continue to closely collaborate with top experts from Virginia Tech.

Formal Resource Consumption Analysis

This research studies the consumption of resources by software. In particular, the focus is on energy consumption. Energy Consumption Analysis (ECA) of IT-controlled systems can play a major role in minimizing the overall energy consumption and sustainability of such IT systems, during the development phase, or for optimization in the field. The goal is to aid software developers in their workflow, by e.g. visualization the energy consumption in their development environment.

Our research focuses on the analysis that can predict the energy consumption of different external devices. This analysis is implemented in a tool called ECA which can analyze the source code controlling external devices, combined with a formal hardware model for each device. The results of the analysis can be used to predict and optimize algorithms, or to support decisions about which hardware to use.

Our researchers play an active role in the Dutch FM community, including participation in the annual "Formal Methods in the Netherlands" workshop, joint project proposals with other Dutch research groups, and collaborations.

Facts and figures

People:
- Bernard van Gastel (assistant professor)
- Stijn de Gouw (assistant professor)
- Sung-Shik Jongmans (assistant professor)
- Freek Verbeek (assistant professor)
- Marko van Eekelen (full professor)
- Luc Edixhoven (PhD student)
- Arjan Lamers (PhD student, external)
- Nico Naus (postdoc)
- Timmy Weerwag (PhD student).

Key collaborators:
- Prof. Frank de Boer (CWI & Universiteit Leiden)
- Prof. Binoy Ravindran (Virginia Tech)
- Prof. Nobuko Yoshida (Imperial College London).

Projects & funding:
- NWO Veni (2019–2022, 248K): New unit-testing techniques for shared-memory concurrent programs
- DARPA AIE (2019-2022, 500K+500K): A formal methodology and algorithms for compile-time reasoning of memory corruption-related emergent behaviors.

Selected publications:

Bockenek, J.A., Verbeek, F., Lammich, P., & Ravindran, B. (2019). Formal verification of memory preservation of x86-64 binaries. In *SAFECOMP*, volume 11698 of *Lecture Notes in Computer Science*, pages 35–49. Springer.

Castro, D., Hu, R., Jongmans, S., Ng, N., & Yoshida, N. (2019). Distributed programming using role-parametric session types in go: statically-typed endpoint apis for dynamically-instantiated communication structures. *PACMPL*, 3(POPL), 29:1–29:30.

De Gouw, S., De Boer, F.S., Bubel, R., Hähnle, R., Rot, J., & Steinhöfel, D. (2019). Verifying openjdk's sort method for generic collections. *J. Autom. Reasoning*, 62(1), 93–126.

De Gouw, S., Rot, J., De Boer, F.S., Bubel, R., & Hähnle, R. (2015). Openjdk's java.utils.collection.sort() is broken: The good, the bad and the worst case. In *CAV (1)*, volume 9206 of *Lecture Notes in Computer Science*, pages 273–289. Springer.

Hamers R., & Jongmans, S. Discourje: Runtime verification of communication protocols in clojure. In *TACAS 2020*, in press.

Hiep, H., Maathuis, O., Bian, J., De Boer, F., Van Eekelen, M., & De Gouw, S. Verifying openjdks linkedlist using key. In *TACAS 2020*, in press.

Jongmans, S., & Yoshida, N. Exploring type-level bisimilarity towards more expressive multiparty session types. In *ESOP 2020*, in press.

Verbeek, F., Bockenek, J., & Ravindran, B. Highly automated formal proofs over memory usage of assembly code. In *TACAS 2020*, in press.

Verbeek, F., Bockenek, J.A., Bharadwaj, A., Ravindran, B., & Roessle, I. (2019). Establishing a refinement relation between binaries and abstract code. In *MEMOCODE*, pages 17:1–17:5. ACM.

3.2 SECURITY & PRIVACY

Security and privacy are crucial to be constantly investigated and improved "in a world where everything is a computer" [49]. Increasingly more things are computerized and connected to the internet, while breaches happen on a daily basis from small data leaks through organized criminal activities to large-scale attacks. As people and the objects around them are becoming a part of this digital world, we need innovative and holistic approaches to guarantee security and privacy for the society.

This research line (entitled 'secOUrity') embraces the challenges from various directions. We analyze existing systems at the network, hardware and software levels. Furthermore, we design and implement technologies and measures to improve security and privacy. Finally, we develop and test novel tools in education. To achieve the best results, this research line combines knowledge and experience from computer science, cryptography, security, artificial intelligence and big data.

We address security and privacy of software, computer systems, and information systems. We focus on the following four directions:

Applied cryptography for security and privacy

Cryptography can contribute to improve security and privacy. Our group develops and implements privacy-enhancing technologies to protect information related to a person's identity. Examples include attributes (a generalization of identities) and polymorphic pseudonymisation (which hides identities).

The former offers attribute-based authentication, signature and access control, while the latter can be used in the context of big data. Our research focuses on the implementation of attribute-based credentials (IRMA), practical cryptographic protection of medical data (PEP) and attribute-based encryption (fundamental cryptographic research).

AI for security

To strengthen diverse security aspects of systems and services, different AI techniques have been used in the last decades by academic researchers and industry practitioners. Such techniques allow intelligent solutions to be designed and implemented for tackling and solving different security challenges and problems. These challenges and problems range from proactively protecting and defending systems and services to responding to security incidents after their occurrence as well as analyzing their impact and decision-making support.

We research the application of hybrid AI techniques for improving security, such as botnet detection based on network traffic analysis, software vulnerability detection for software code and metadata analysis, and impact assessment and decision support for military Cyber/Information Operations.

Digital fingerprinting

Digital fingerprinting is a means to track or identify visitors to websites. The secOUrity group investigates the spread of fingerprinting techniques and countermeasures. Furthermore, we investigate the extent to which browser fingerprinting is used to detect scrapers and the impact on reliability this has on studies performed using these scrapers. Finally, we design and implement countermeasures that enable large-scale web studies whose results are not marred by scraper detection.

Formal methods for security

We develop formal models of real-life systems and describe how they can be disrupted by malicious actors. Thanks to the application of formal methods such as model checking, we can obtain useful insights into how system weaknesses can be exploited. This allows to support decisions on strengthening the security of systems. Thanks to software tools developed for this purpose using advanced software engineering techniques [48], the power of formal methods is made available to security experts without the need for additional formal training [35].

Software tools to support security education

We consider that, next to teaching theoretical concepts and principles of security and privacy, it helps students to improve their understanding and deepen their knowledge by offering hands-on experience in labs. We provide this by researching, developing, and applying virtual labs in which students can simulate both defensive and offensive techniques in a realistic environment. We research both the technical infrastructure of distributed virtual labs, in which groups of remote students can work together, and how such labs can be applied for security education in distance teaching.

Societal relevance

The Dutch government has defined the 'missie gedreven topsectoren- en innovatiebeleid' in April 2019. The top sectors subsequently have implemented this mission into four thematic 'Kennis en Innovatie Agenda's' (KIAs) that resulted in 'Meerjarige Missiegedreven Innovatie Programma's' (MMIPs). The 'KIA Veiligheid' addresses the MMIP Cybersecurity [58], which covers five programs: (i) fighting cybercrime, (ii) improving development of cybercompetences, (iii) defensive cyber technology, (iv) offensive cyber technology, and (v) chain resilience and governance. Our research addresses all of these five programs, as indicated in the following table.

MMIP Cybersecurity program	Our research
Fighting cybercrime	Botnet detection; illegal cryptocurrency mining; digital forensics; scrapology
Improving development of cyber competences	Usable security in e-banking; attribute-based authentication, software tools for security education, cybersecurity training programs
Defensive cyber technology	Attribute-based technologies (IRMA, PEP); cryptography in attribute-based technologies; system and software vulnerability prevention and detection; privacy engineering and measurement
Offensive cyber technology	Hardware encryption attacks; security evaluation; impact assessment
Chain resilience and governance	Policy aspects of hardware encryption attacks; cyber safety; IoT security & privacy

Facts and figures

People:
- Greg Alpár (assistant professor)
- Bernard van Gastel (assistant professor)
- Hugo Jonker (assistant professor)
- Clara Maathuis (assistant professor)
- Stefano Schivo (assistant professor)
- Harald Vranken (associate professor)
- Hassan Alizadeh (researcher)
- Fabian van den Broek (researcher)
- Benjamin Krumnow (PhD student, external)
- Vincent van der Meer (PhD student, external)
- Sietse Ringers (researcher).

Key collaborators:
- Digital Security group at Radboud University (prof. dr. Bart Jacobs, dr. Jaap-Henk Hoepman, dr. ir. Erik Poll)
- University of Luxembourg (prof. dr. Sjouke Mauw)
- Delft University of Technology
- Cologne University of Applied Sciences (prof. dr. Stefan Karsch)
- Fern Universität in Hagen (prof. dr. Jörg Keller)
- NFI (dr. Jeroen van den Bos)
- SIDN Labs
- SURFnet
- TNO.

Projects & funding:
- Dagobert: Design, application and governance of a botnet detection and profiling system (funded by OU (LIRS) and SIDN Labs; in cooperation with SURFnet and Tesorion)
- Digital forensics of file systems, NWO Doctoral Grant for Teachers (Vincent van der Meer, PhD project, in cooperation with NFI (Nederlands Forensisch Instituut))
- Design of a privacy-friendly parking payment system using license plate scanners (Haarlem municipality).

Selected publications:
Haag, J., Vranken, H.P.E., & Van Eekelen, M.C.J.D. (2019). A virtual classroom for cybersecurity education. *T. Edutainment*, 15, 173–208.

Jonker, H., Krumnow, B., & Vlot, G. (2019). Fingerprint surface-based detection of web bot detectors. In *Proc. 24th European Symposium on Research in Computer Security Part II (ESORICS'19)*, volume 11736 of *LNCS*, pages 586–605. Springer.

Kronjee, J., Hommersom, A., & Vranken, H. (2018). Discovering software vulnerabilities using data-flow analysis and machine learning. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*, pages 1–10. ACM.

Maathuis, C., Pieters, W., & Van den Berg, J. (2018). A knowledge-based model for assessing the effects of cyber warfare. In *Proceedings of the 12th NATO Conference on Operations Research and Analysis*. NATO.

Torres Ferreira, Jonker, H.L., & Mauw, S. (2015). Fp-block: usable web privacy by controlling browser fingerprinting. In *Proc. 20th European Symposium on Research in Computer Security, part II (ESORICS'15)*, volume 9327 of *LNCS*, pages 3–19. Springer.

Van Roosmalen, J., Vranken, H.P.E., & Van Eekelen, M.C.J.D. (2018). Applying deep learning on packet flows for botnet detection. In *SAC*, pages 1629–1636. ACM.

3.3 ARTIFICIAL INTELLIGENCE

Artificial intelligence (AI) as a domain is growing in society, in business, but also in the computer science department. Furthermore, national and European research agendas in AI strongly promote both technical advances (such as deep learning) and societal challenges to be tackled (such as the ethical study of consequences of AI technology). Locally here at CS, we are actively involved in both lines of research, and both applied and academically. In addition, educational developments in the AI subgroup are ongoing, including a new master program in AI which is under development (projected start: 2021 or 2022). Research-wise, we address AI largely in two themes:

Trustworthy artificial intelligence

Trustworthy systems need to be understood in the same way as the current national and European research agendas are structured, with one dimension focusing on technical solutions and other on analysis of societal and business consequences of the application of AI systems. The first is about new algorithms, and for example deep learning architectures and new sensor data perception mechanisms, whereas the second is about transparent and explainable AI and its consequences on life, work, society, business opportunities, but also privacy and surveillance [63].

In our research, we tackle both challenges, to some extent, by an explicit focus on models to be used in various AI solutions. Models can represent domain knowledge (extracted from domain experts) but they can also provide ways for AI systems to explain themselves and their decisions, thereby contributing to trust between humans and AI. Typically such models include Bayesian networks and logical knowledge representation and graphs, and they get combined with effective learning settings and algorithms from supervised and reinforcement learning.

The group has expertise in (socio-ethical) analysis of algorithms [63], and also in the study of explainable systems for design [62] (for example in legal contexts). Other, more general studies on explainability and algorithms have been done too, cf. [11, 43]. This subfield of AI is rapidly expanding [61] and includes work on the use of formal AI models to reason and learn about ethical contexts [64].

AI in practice

AI can be very effective for a very wide range of problems. In our research we have a keen eye for the *applicability* of new algorithms and concepts. One key collaboration is with the Centre for Actionable Research (CAROU), located at Brightlands Smart Services Campus in Heerlen, where these insights are used to support innovation efforts with business partners. All of the AI researchers at the OU apply AI to various degrees and on various domains, including health, education, logistics and cybersecurity.

Reinforcement learning (RL) is a core expertise of the group [70] and recent research includes games and personalized health. For example, the use of RL in *mhealth* settings to provide users with reminders for physical exercise in combination with clustering techniques for precise *personalization* [25] can be extended to *end-to-end learning* in which *deep neural networks* effectively deal with not-preprocessed, *raw* sensor data [17]. An example of recent work on *supervised* approaches in *marketing* (e.g. *customer profiling*) settings targets

the prediction of customer actions in *loyalty programs* [60]. AI in practice, e.g. in medicine [36, 10], software security [33], military domain and social manipulation [39, 9]. In addition, a collaboration is ongoing with the *testing* group (e.g. Prof. Tanja Vos, Pekka Aho, and others) to apply existing AI techniques in (software and GUI) testing and to devise new AI algorithms for this application domain.

Facts and figures

People:
- Arjen Hommersom (assistant professor)
- Clara Maathuis (assistant professor)
- Martijn van Otterlo (assistant professor)
- Stefano Bromuri (associate professor, and CAROU)
- Raphaela Butz (PhD student, external)
- Andreas Schuderer (PhD student, external)
- Simone Tummers (PhD student).

Projects & funding:
- Active4Life: project funded by ZonMW on using data science methods to identify determinants of physical activity in vulnerable groups
- Graph-based Neural Models for Dialogue Management (funded by the Norwegian Research Council) with other partners from Norway, Japan, and Germany
- Deep Solaris (Stefano Bromuri): EUROSTAT project in which we evaluate the feasibility of using object detection on aerial images to detect solar panels on rooftops.

3.4 TEACHING AND LEARNING

3.4.1 *Software technology for learning and teaching*

In our research, we investigate how techniques from Software Technology (ST) and Programming Languages (PL) can be used for designing and building tools for education. More specifically, we investigate how to automatically generate hints and feedback for such tools. An Intelligent Tutoring System (ITS) helps students with learning a particular topic. It typically does this by offering learning material to study, tasks to solve, and by providing various kinds of help. The help provided by an ITS may take several forms: it might be through sequencing the tasks in a way that suits the student, providing scaffolding at the level of the student, giving elaborated feedback on the steps a student takes towards a solution for a task, helping a student take a next step towards a solution, etc. In our research, we focus on two topics.

Generic ITS framework

On the one hand we work on the generic framework IDEAS (Interactive Domain-specific Exercise Assistants; https://hackage.haskell.org/package/ideas) for constructing expert-knowledge modules, also called domain reasoners. Examples of ST/PL techniques that are part of the framework are the domain-specific language for describing problem-solving procedures in a compositional way, datatype-generic rewriting techniques and traversals, and type-level APIs for describing feedback services. The framework is independent of the problem domain. In our research, we study the comparison between our framework and other ITS approaches found in literature. Furthermore, we extend the framework's expressiveness. Topics that we want to investigate further include constraints from the constraint-based tutoring paradigm, for which we have found new applications in our tools (e.g. for inductive proofs and hypothesis testing). The question is how constraints can be combined with other knowledge components, in particular problem-solving procedures, and what kind of feedback can be reported. Then there is a trend towards data-driven ITSs: these systems use AI techniques to generate feedback from collected data, and typically scale well. We are interested in hybrid solutions that combine data-driven techniques with expert knowledge.

There is a need for further adaptation and personalization, both for the inner loop and the outer loop. This requires models for mastery learning (e.g. techniques for Bayesian knowledge tracing), and more advanced techniques to use the information from such models in domain reasoners. We plan to extend our tools with student modeling features.

We want to better understand the combinators for describing problem-solving procedures by formalizing their semantics, and studying the underlying algebra.

Specific ITS problem domains

On the other hand, we use the IDEAS framework for specific problem domains to build Intelligent Tutoring Systems. By studying specific problem domains, we discover new opportunities and challenges, which typically leads to general solutions that can also be applied in other problem domains. The tools we develop are tested in a classroom setting, after which logfiles are analyzed and effects are measured. Examples of problem-domains we have studied are:

- *Logic*. We have developed several tools for teaching logic, focusing on (disjunctive) normal forms, equivalence proofs, axiomatic proofs, and proofs by induction. We investigate how parts can be reused across these problem types, what type of support (and how much support) should be provided, and what other types of problems can be supported.
- *Functional programming*. The Ask-Elle tutor generates feedback automatically based on teacher-annotated model solutions. We want to explore the use of refinement types, program synthesis and subcontracts, together with syntax-directed (and type-directed) programming, to provide better hints for students that are diverging from the anticipated model solutions.
- *Imperative programming*. We study the construction of simple imperative programs on the level of single methods. In particular, we are interested in providing feedback about the quality of such programs, and in offering step-wise refactorings. Further research is needed to better deal with differences in the granularity of rewriting steps, and to allow customization of refactorings by teachers. We want to develop a theory of refactoring.

In the Advise-Me project (http://advise-me.ou.nl) we developed an approach to using ICT for assessing mathematics achievement of pupils using learning environments for mathematics. In particular, we looked at fine-grained cognitive assessment of free-form answers to math story problems, which requires determining the steps a pupil takes towards a solution, together with the high-level solution approach used by the pupil. We have performed multiple pilot studies in different European countries and collected data that requires further processing.

Facts and figures

People:
- Bastiaan Heeren (associate professor)
- Johan Jeuring (full professor)
- Josje Lodder (assistant professor)
- Hieke Keuning (PhD student, external).

Key collaborators:
- Research group Software Technology for Learning and Teaching at Utrecht University
- Alex Gerdes (University of Gothenburg and Chalmers)
- Math education group at Freudenthal Institute (in particular, prof. Paul Drijvers and Sietske Tacoma).

Projects & funding:
- Providing feedback in an interactive learning tool for imperative programming, NWO Doctoral Grant for Teachers (Hieke Keuning, PhD defence planned for 2020)
- Tools for teaching logic (Josje Lodder, PhD defence planned for 2020)
- In the Digital support for university-level statistics courses project (funded by Utrecht University) we have looked at developing support for several statistics courses at Utrecht University (Sietske Tacoma, UU, PhD defence planned for 2020)

Selected publications:
Gerdes, A., Heeren, B., Jeuring, J., & Van BinsBergen, L.T. (2017). Ask-elle: an adaptable programming tutor for haskell giving automated feedback. *I. J. Artificial Intelligence in Education*, 27(1), 65–100.

Heeren, B., & Jeuring, J. (2017). An extensible domain-specific language for describing problem-solving procedures. In *AIED*, volume 10331 of *Lecture Notes in Computer Science*, pages 77–89. Springer.

Heeren, B., Jeuring, J., Sosnovsky, S.A., Drijvers, P., Boon, P., Tacoma, S., Koops, J., Weinberger, A., Grugeon-Allys, B., Chenevotot-Quentin, F., Van Wijk, J., & Van Walree, F. (2018). Fine-grained cognitive assessment based on free-form input for math story problems. In *EC-TEL*, volume 11082 of *Lecture Notes in Computer Science*, pages 262–276. Springer.

Keuning, H., Heeren, B., & Jeuring, J. (2017). Code quality issues in student programs. In *ITiCSE*, pages 110–115. ACM.

Keuning, H., Jeuring, J., & Heeren, B. (2019). A systematic literature review of automated feedback generation for programming exercises. *TOCE*, 19(1), 3:1–3:43.

Lodder, J., Heeren, B., & Jeuring, J. (2017). Generating hints and feedback for hilbert-style axiomatic proofs. In *SIGCSE*, pages 387–392. ACM.

Lodder, J., Heeren, B., & Jeuring, J. (2019). A comparison of elaborated and restricted feedback in logex, a tool for teaching rewriting logical formulae. *J. Comp. Assisted Learning*, 35(5), 620–632.

Tacoma, S., Heeren, B., Jeuring, J., & Drijvers, P. (2019). Automated feedback on the structure of hypothesis tests. In *AIED (2)*, volume 11626 of *Lecture Notes in Computer Science*, pages 281–285. Springer.

3.4.2 *Computing Education*

Our research concerns *content specific pedagogy* (Dutch: vakdidactiek) for computer science and related areas such as digital literacy, in particular computational thinking. We investigate teaching and learning of specific content (i.e., concepts or practices), with four key aspects: (1) *goals and objectives* connected to this content, (2) *students' understanding* of this content, (3) *instructional strategies* for teaching this content, and (4) *assessment* connected to this content.

Our research activities are mostly practice-oriented and focus on particular content and education levels, such as software engineering in higher education, programming in primary, secondary and higher education, and computational thinking in primary and secondary education. Apart from the four pedagogical aspects mentioned above, we study teachers' practical knowledge and skills connected to these aspects, in particular teachers' pedagogical content knowledge (PCK). We address three main themes in our research. We describe them below, including some characteristic examples of ongoing research activities.

Scaffolding in Software Engineering

Software Engineering involves several complex tasks, for which students need to acquire both conceptual and procedural knowledge. In this theme we study student support in the form of *procedural guidance* as an instructional strategy. The idea of *scaffolding* entails that support is provided for beginning students and is

gradually removed as students become more autonomous. Existing procedures and guidelines from computer science or software engineering are taken as a starting point with the aim to adapt these for educational use. In some cases, however, procedures are tacit or do not exist at all. In such cases the first step is making such procedures explicit.

Among others, we have studied procedural guidance for constructing XML content models, designing concurrent programs, integrated test development, software redesign, and maintaining test coverage during refactoring. We mainly focus on the Software Engineering domains *refactoring* and *testing*. Within these domains, we are interested in developing *procedures*, automated *tools* based on these procedures, and the applications of these in *teaching*.

Besides, we investigate the difficulties and hindrances students experience, and how procedural guidance and tools help them in overcoming these hindrances. Moreover, we address the effectiveness and efficiency of the procedural guidance. The resulting insights are used to improve our procedures and scaffolding strategies.

Programming Education

Code quality is considered an important aspect of programming, but research on teaching and learning about code quality is rare. In our research we investigate which criteria for code quality are suitable for teaching novice programmers. Moreover, we study students' perceptions of code quality and the stages they go through while learning about this topic. Finally, we develop and test instruments for assessing code quality, as well as the application of these as tools for formative assessment, both for teacher feedback and for peer review of students' program code.

Programming can be applied in so-called *design-based education*, in which students learn by developing digital artifacts. Designing can serve as an authentic and engaging context for programming education. It is known that students not only *use* fundamental concepts when designing, but also deepen their conceptual knowledge through the development process. We investigate how to monitor students' conceptual development without disturbing the authentic character of the design context. Intermediate design products play a key role in our instructional model. Moreover, we study teachers' PCK with respect to design in computer science.

Nowadays, students of all ages get engaged in programming, most recently students in lower secondary and even in primary schools. We investigate ways to introduce *programming to young learners* in an age-appropriate way, that is in accordance with the students' cognitive development. We study students' understanding of programming concepts, including the occurrence of misconceptions, as well as specific instructional strategies for young learners. We apply our results to advise on curriculum development issues.

Computational Thinking

The term *Computational Thinking* refers to a set of problem-solving skills that make use of concepts and methods stemming from computer science. Computational Thinking is the thought process used to understand and formulate problems in such a way that they can be solved in terms of computations. Common elements in characterizations of Computational Thinking are decomposition, abstraction, algorithmic thinking, evaluation and generalization. We investigate students' understanding and instructional strategies for Computational Thinking, especially with respect to abstraction skills of primary school students. Special attention is given to teachers' knowledge and skills required for incorporating Computational Thinking in their lessons.

Facts & figures

People:
- Greg Alpar (assistant professor)
- Fenia Aivaloglou (assistant professor)
- Erik Barendsen (full professor)
- Arjan Kok (assistant professor)
- Harrie Passier (assistant professor)
- Ebrahim Rahimi (assistant professor)
- Stefano Schivo (assistant professor)
- Sylvia Stuurman (assistant professor)
- Alaaeddin Swidan (assistant professor)
- Hylke Faber (PhD student)
- Martijn Stegeman (PhD student, external).

Key collaborators:
- Lex Bijlsma, Harold Pootjes
- Delft University of Technology
- Eindhoven University of Technology
- Fontys University of Applied Sciences (Tilburg)
- Hanze University of Applied Sciences (Groningen)
- Leiden University
- NHL Stenden University of Applied Sciences (Emmen)
- Radboud University, University of Amsterdam
- University of Kiel
- University of Marburg
- University of Paderborn
- VHTO.

Projects & funding:
- SERF (funding: SURF)
- Effective feedback on code quality in programming courses (funding: University of Amsterdam)
- Programmeren in het basisonderwijs (funding: TechYourFuture)
- Development of Abstraction in Primary Education (funding: Hanze University of Applied Sciences, Open University, Gemeente Groningen).

Selected publications:
Bijlsma, L., Huizing, C., Kuiper, R., Passier, H.J.M., Pootjes, H.J., & Smetsers, A.J.E.W. (2017). A structured design methodology for concurrent programming. In *Proceedings of the 6th Computer Science Education Research Conference*, CSERC 17, page 19, New York, NY, USA. Association for Computing Machinery.

Bijlsma, L., Passier, H.J.M., Pootjes, H.J., & Stuurman, S. (2018). Integrated test development: An integrated and incremental approach to write software of high quality. In *Proceedings of the 7th Computer Science Education Research Conference*, CSERC 18, page 920, New York, NY, USA. Association for Computing Machinery.

Bijlsma, L., Passier, H.J.M., Pootjes, H.J., & Stuurman, S. (2018). Template method test pattern. *Information Processing Letters*, 139, 8–12.

Faber, H., Koning, J.I., Wierdsma, M., Steenbeek, H., & Barendsen, E. (2019). Observing abstraction in young children solving algorithmic tasks. In *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*, pages 95–106.

Passier, H., Bijlsma, L., & Bockisch, C. (2016). Maintaining unit tests during refactoring. In *Proceedings of the 13th International Conference on Principles and Practices of Programming on the Java Platform: Virtual Machines, Languages, and Tools*, PPPJ 16, New York, NY, USA. Association for Computing Machinery.

Rahimi, E., Barendsen, E., & Henze, I. (2016). Typifying informatics teachers' PCK of designing digital artefacts in Dutch upper secondary education. In *Informatics in Schools: Improvement of Informatics Knowledge and Perception*, pages 65–77. Springer.

Rahimi, E., Barendsen, E., & Henze, I. (2018). An instructional model to link designing and conceptual understanding in secondary computer science education. In *Proceedings of the 13th Workshop on Primary and Secondary Computing Education*, pages 67–70. ACM.
Stegeman, M., Barendsen, E., & Smetsers, S. (2014). Towards an empirically validated model for assessment of code quality. In *Proceedings of the 14th Koli Calling International Conference on Computing Education Research*, pages 99–108. ACM.
Stegeman, M., Barendsen, E., & Smetsers, S. (2016). Designing a rubric for feedback on code quality in programming courses. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*, pages 160–164. ACM.
Stuurman, S., Passier, H., & Barendsen, E. (2016). Analyzing students software redesign strategies. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*, Koli Calling 16, page 110119, New York, NY, USA. Association for Computing Machinery.
Swidan A., & Hermans, F. (2019). The effect of reading code aloud on comprehension: An empirical study with school students. In *Proceedings of the ACM Conference on Global Computing Education*, pages 178–184.
Swidan, A., & Hermans, F., & Smit, M. (2018). Programming misconceptions for school students. In *Proceedings of the 2018 ACM Conference on International Computing Education Research*, pages 151–159.

3.5 MULTI-DISCIPLINARY RESEARCH

The core of the research in the Department of Computer Science is mono-disciplinary, covering a focused set of topics in computer science as explained above. In addition, we are also involved in multi-disciplinary research within the OU.

We cooperate with Management Science and Environmental Science in the multi-disciplinary MST research program *Learning and Innovation in Resilient Systems*. The aim of this research program is to increase our understanding of the innovative and learning capacity of resilient systems, with a focus on (i) information and computer systems, (ii) organizational and management systems, and (iii) environmental systems. Resilience here means the capacity of a system to absorb disturbance and reorganize while undergoing change so as to still retain essentially the same function. We also cooperate with Humanities, Law, Psychology, and Educational Sciences.

Below are some examples of multi-disciplinary research projects in which the Department of Computer Science contributes. We explore the energy consumption and environmental footprint of cryptocurrency mining, which combines research in the field of computer science and research in the field of environmental science. We explore how real-time analysis of network traffic streams at network infrastructure providers can help to detect malicious activities. This not only involves technical solutions, but also taking care of governance aspects such as how to organize the follow-up after detecting malicious activity, both by the provider as well as law enforcement. This combines research in the field of computer science and research in the field of information science and business processes. We develop innovations in online mathematics education in the 'Open Maths activating learning mathematics online' project (Comenius Teaching Fellow-funding). This combines research in the fields of neuro science, psychology, and didactics.

3.6 COLLABORATIONS

We have a strong collaboration with Radboud University both in teaching and in research. Additionally, we collaborate with:
- Utrecht University
- Leiden University
- Groningen University
- Technical University Delft
- Technical University Eindhoven
- Centrum Wiskunde & Informatica.

We are actively contributing to the activities of Software Engineering Netherlands (SEN). We also collaborate with private/public organizations in the Netherlands (including RDW, NRG, SIDN Labs, Solcon, XS4ALL and the Privacy by Design foundation).

Internationally, we collaborate both via long research visits of individual researchers in particular:
- Imperial College London (United Kingdom)
- Virginia Tech and the University of Maryland (United States of America).

And via European projects with universities outside the Netherlands, in particular
- University of Passau and TH Köln (Germany)
- University of Valencia
- Universidad Complutense de Madrid
- University of Mondragon Goi Eskola Politeknikoa (Spain)
- University of Luxembourg
- University of Southern Denmark (Denmark)
- KTH Royal Institute of Technology and Mälardalen University (Sweden)
- University of Oslo (Norway)
- University of Oulou (Finland)
- University of Bologna (Italy)
- Middle East Technical University (Turkey)
- Instituto de Engenharia de Sistemas e Computadores, Investigação e Desenvolvimento em Lisboa (Portugal).

4 References

Aho, P., Alégroth, E., Oliveira, R.A.P., & Vos, T.E.J. (2016). Evolution of automated regression testing of software systems through the graphical user interface. In *the First International Conference on Advances in Computation, Communications and Services (ACCSE 2016)*, pages 16–21.

Aho, P., & Vos, T.E.J. (2018). Challenges in automated testing through graphical user interface. In *2018 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pages 118–121, Los Alamitos, CA, USA. IEEE Computer Society.

Aho, P., Vos, T.E.J., Ahonen, S., Piirainen, T., Moilanen, P., & Ricos, F.P. (2019). Continuous piloting of an open source test automation tool in an industrial environment. In *Jornadas de Ingeniera del Software y Bases de Datos (JISBD)*, pages 1–4. Sistedes.

Assessment Committee (Rik Leemans, Jos Benders, Michel van Eeten, and Wim Lambrechts) (2017). Midterm review Learning and Innovation in Resilient Systems 2014-2016, October 31.

Bijlsma, L., Huizing, C., Kuiper, R., Passier, H.J.M., Pootjes, H.J., & Smetsers, A.J.E.W. (2017). A structured design methodology for concurrent programming. In *Proceedings of the 6th Computer Science Education Research Conference*, CSERC 17, page 19, New York, NY, USA. Association for Computing Machinery.

Bijlsma, L., Passier, H.J.M., Pootjes, H.J., & Stuurman, S. (2018). Integrated test development: An integrated and incremental approach to write software of high quality. In *Proceedings of the 7th Computer Science Education Research Conference*, CSERC 18, page 920, New York, NY, USA. Association for Computing Machinery.

Bijlsma, L., Passier, H.J.M., Pootjes, H.J., & Stuurman, S. (2018). Template method test pattern. *Information Processing Letters*, 139, 8–12.

Bockenek, J.A., Verbeek, F., Lammich, P., & Ravindran, B. (2019). Formal verification of memory preservation of x86-64 binaries. In *SAFECOMP*, volume 11698 of *Lecture Notes in Computer Science*, pages 35–49. Springer.

Boltjes, Maathuis, C., & Van den Berg, T. (2019). Developing modelling and simulation standards for including the cyber domain in military training and exercises. In *Proceedings of the Simulation Interoperability Standards Organization Innovation Workshop (SISO)*.

Bueno, M.L.P., Hommersom, A., Lucas, P.J.F., & Janzing, J. (2019). A probabilistic framework for predicting disease dynamics: A case study of psychotic depression. *Journal of biomedical informatics*, 95, 103232.

Butz, R., Hommersom, A., & Van Eekelen, M. (2018). Explaining the most probable explanation. In *International Conference on Scalable Uncertainty Management*, pages 50–63. Springer.

Castro, D., Hu, R., Jongmans, S., Ng, N., & Yoshida, N. (2019). Distributed programming using role-parametric session types in go: statically-typed endpoint apis for dynamically-instantiated communication structures. *PACMPL*, 3(POPL), 29:1–29:30.

Dcypher. NCSRA III: National Cyber Security Research Agenda, June 5, 2018.

De Gier, F., Kager, D., De Gouw, S., & Vos, T.E.J. (2019). Offline oracles for accessibility evaluation with the testar tool. In *2019 13th International Conference on Research Challenges in Information Science (RCIS)*, pages 1–12.

De Gouw, S., De Boer, F.S., Bubel, R., Hähnle, R., Rot, J., & Steinhöfel, D. (2019). Verifying openjdk's sort method for generic collections. *J. Autom. Reasoning*, 62(1), 93–126.

De Gouw, S., Rot, J., De Boer, F.S., Bubel, R., & Hähnle, R. (2015). Openjdk's java.utils.collection.sort() is broken: The good, the bad and the worst case. In *CAV (1)*, volume 9206 of *Lecture Notes in Computer Science*, pages 273–289. Springer.

El Hassouni, A., Hoogendoorn M., Eiben, A.E., Van Otterlo, M., & Muhonen V. (2019). End-to-end personalization of health interventions using raw sensor data with deep reinforcement learning. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI2019)*.

Esparcia, A.I., Almenar, F., Martnez, M., Rueda, U., & Vos T.E.J. (2016). Q-learning strategies for action selection in the testar automated testing tool. In *6th International Conference on Metaheuristics and nature inspired computing (META 2016)*, pages 130–137.

Esparcia, A.I., Almenar, F., Rueda, U., & Vos T.E.J. (2017). Evolving rules for action selection in automated testing via genetic programming - A first approach. In *Applications of Evolutionary Computation - 20th European Conference, EvoApplications 2017, Amsterdam, The Netherlands, April 19-21, 2017, Proceedings, Part II*, pages 82–95.

Esparcia, A.I., Almenar, F., Vos, T.E.J., & Rueda, U. (2018). Using genetic programming to evolve action selection rules in traversal-based automated software testing: results obtained with the TESTAR tool. *Memetic Computing*, 10(3), 257–265.

Faber, H., Koning, J.I., Wierdsma, M., Steenbeek, H., & Barendsen, E. (2019). Observing abstraction in young children solving algorithmic tasks. In *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*, pages 95–106.

Gerdes, A., Heeren, B., Jeuring, J., & Van BinsBergen, L.T. (2017). Ask-elle: an adaptable programming tutor for haskell giving automated feedback. *I. J. Artificial Intelligence in Education*, 27(1):65–100.

Haag, J., Vranken, H.P.E., & Van Eekelen, M.C.J.D. (2019). A virtual classroom for cybersecurity education. *T. Edutainment*, 15, 173–208.

Hamers R., & Jongmans, S. Discourje: Runtime verification of communication protocols in clojure. In *TACAS 2020*, in press.

El Hassouni, A., Hoogendoorn, M., Van Otterlo, M., & Barbaro, E. (2018). Personalization of health interventions using cluster-based reinforcement learning. In *PRIMA 2018: Principles and Practice of Multi-Agent Systems*, pages 467–475, Cham, Springer International Publishing.

Heeren, B., & Jeuring, J. (2017). An extensible domain-specific language for describing problem-solving procedures. In *AIED*, volume 10331 of *Lecture Notes in Computer Science*, pages 77–89. Springer.

Heeren, B., Jeuring, J., Sosnovsky, S.A., Drijvers, P., Boon, P., Tacoma, S., Koops, J., Weinberger, A., Grugeon-Allys, B., Chenevotot-Quentin, F., Van Wijk, J., & Van Walree, F. (2018). Fine-grained cognitive assessment based on free-form input for math story problems. In *EC-TEL*, volume 11082 of *Lecture Notes in Computer Science*, pages 262–276. Springer.

Hiep, H., Maathuis, O., Bian, J., De Boer, F., Van Eekelen, M., & De Gouw, S. Verifying openjdks linkedlist using key. In *TACAS 2020*, in press.

Jongmans, S., & Yoshida, N. Exploring type-level bisimilarity towards more expressive multiparty session types. In *ESOP 2020*, in press.

Jonker, H., Krumnow, B., & Vlot, G. (2019). Fingerprint surface-based detection of web bot detectors. In *Proc. 24th European Symposium on Research in Computer Security Part II (ESORICS'19)*, volume 11736 of *LNCS*, pages 586–605. Springer.

Keuning, H., Heeren, B., & Jeuring, J. (2017). Code quality issues in student programs. In *ITiCSE*, pages 110–115. ACM.

Keuning, H., Jeuring, J., & Heeren, B. (2019). A systematic literature review of automated feedback generation for programming exercises. *TOCE*, 19(1), 3:1–3:43.

Kronjee, J., Hommersom, A., & Vranken, H. (2018). Discovering software vulnerabilities using data-flow analysis and machine learning. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*, pages 1–10. ACM.

Kuhrmann, H., Diebold, P., Münch, J., Tell, P., Trektere, K., McCaffery, F., Vahid, G., Felderer, M., Linssen, O., & Hanser, E. (2018). Hybrid software development approaches in practice: A European perspective. *IEEE Software*.

Kumar, R., Schivo, S., Ruijters, E.J.J., Yildiz, B.M., Huistra, D.J., Brandt, J., Rensink, A., & Stoelinga, M.I.A. (2018). Effective analysis of attack trees: A model-driven approach.

Liu, M., Stella, F., Hommersom, A., Lucas, P.J.F., Boer, L., & Bischoff, E. (2019). A comparison between discrete and continuous time bayesian networks in learning from clinical time series data with irregularity. *Artificial intelligence in medicine*, 95, 104–117.

Lodder, J., Heeren, B., & Jeuring, J. (2017). Generating hints and feedback for hilbert-style axiomatic proofs. In *SIGCSE*, pages 387–392. ACM.

Lodder, J., Heeren, B., & Jeuring, J. (2019). A comparison of elaborated and restricted feedback in logex, a tool for teaching rewriting logical formulae. *J. Comp. Assisted Learning*, 35(5), 620–632.

Maathuis, C., Pieters, W., & Van den Berg, J. Developing a cyber operations computational ontology. *Journal of Information Warfare*, 17(3), 32–49.

Maathuis, C., Pieters, W., & Van den Berg, J. (2018). A knowledge-based model for assessing the effects of cyber warfare. In *Proceedings of the 12th NATO Conference on Operations Research and Analysis*. NATO.

MST Research Committee (Carolien Kroeze, Marjolein Caniëls, Dave Huitema, and Harald Vranken) (2014). Learning and Innovation in Resilient Systems: MST Research Program 2015-2020, December 18.

MST Research Committee (Petru L. Curseu, Marjolein Caniëls, Dave Huitema, Harold Krikke, Harald Vranken, and Annemarie Cremers) (2017). Learning and Innovation in Resilient Systems 2014-2016, Januari 10.

Nalepa, G.J., Van Otterlo, M., Bobek, S., & Atzmueller, M. (2019). From context mediation to declarative values and explainability. In *Proceedings of the Second International Explainable Artificial Intelligence Workshop (XAI) colocated with the International Joint Conference on Artificial Intelligence (IJCAI)*.

School of Computer Science (2011). Software Technology Research Plan 2010-2015.

Passier, H., Bijlsma, L., & Bockisch, C. (2016). Maintaining unit tests during refactoring. In *Proceedings of the 13th International Conference on Principles and Practices of Programming on the Java Platform: Virtual Machines, Languages, and Tools*, PPPJ 16, New York, NY, USA. Association for Computing Machinery.

Rahimi, E., Barendsen, E., & Henze, I. (2016). Typifying informatics teachers' PCK of designing digital artefacts in Dutch upper secondary education. In *Informatics in Schools: Improvement of Informatics Knowledge and Perception*, pages 65–77. Springer.

Rahimi, E., Barendsen, E., & Henze, I. (2018). An instructional model to link designing and conceptual understanding in secondary computer science education. In *Proceedings of the 13th Workshop on Primary and Secondary Computing Education*, pages 67–70. ACM.

Schivo, S., Yildiz, B.M., Ruijters, E.J.J., Gerking, C., Kumar, R., Dziwok, S., Rensink, A., & Stoelinga, M.I.A. (2017). How to efficiently build a front-end tool for uppaal: A model-driven approach.

Bruce Schneier. Click here to kill everybody (talks at google). https://www.youtube.com/watch?v=GkJCI3 jbtg, 2018.

SOGETI (2019). The world quality report 2019-2020.

Somers, J. (2017). The coming software apocalypse. *The Atlantic*, 26.

Stegeman, M., Barendsen, E., & Smetsers, S. (2014). Towards an empirically validated model for assessment of code quality. In *Proceedings of the 14th Koli Calling International Conference on Computing Education Research*, pages 99–108. ACM.

Stegeman, M., Barendsen, E., & Smetsers, S. (2016). Designing a rubric for feedback on code quality in programming courses. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*, pages 160–164. ACM.

Stuurman, S., Passier, H., & Barendsen, E. (2016). Analyzing students software redesign strategies. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*, Koli Calling 16, page 110119, New York, NY, USA. Association for Computing Machinery.

Swidan A., & Hermans, F. (2019). The effect of reading code aloud on comprehension: An empirical study with school students. In *Proceedings of the ACM Conference on Global Computing Education*, pages 178–184.

Swidan, A., & Hermans, F., & Smit, M. (2018). Programming misconceptions for school students. In *Proceedings of the 2018 ACM Conference on International Computing Education Research*, pages 151–159.

Tacoma, S., Heeren, B., Jeuring, J., & Drijvers, P. (2019). Automated feedback on the structure of hypothesis tests. In *AIED (2)*, volume 11626 of *Lecture Notes in Computer Science*, pages 281–285. Springer.

Topsector High Tech Systemen en Materialen, Team Dutch Digital Delta, Topsector Creatieve Industrie, Topsector Logistiek en Topsector Water & Maritiem. Kennis-en innovatieagenda veiligheid, October 2019.

Torres Ferreira, Jonker, H.L., & Mauw, S. (2015). Fp-block: usable web privacy by controlling browser fingerprinting. In *Proc. 20th European Symposium on Research in Computer Security, part II (ESORICS'15)*, volume 9327 of *LNCS*, pages 3–19. Springer.

Van den Boogaart, Hantke, T., & Van Otterlo, M. (2018). Predicting next week redemption in a digital frequency loyalty program. In *Proceedings of the Dutch-Belgian Conference on Machine Learning (BeNeLearn)*.

Van Otterlo, M. (2018). Ethics and the value(s) of artificial intelligence. *Nieuw Archief Wiskunde (NAW)*, 5/19(3).

Van Otterlo, M., & Atzmueller, M. (2019). On requirements and design criteria for explainability in legal ai. In *Proceedings of the EXplainable AI in Law Workshop (XAILA 2018) co-located with the 31st International Conference on Legal Knowledge and Information Systems (JURIX 2018)*, pages 29–32.

Van Otterlo, M. (2014). Automated experimentation in walden 3.0.: The next step in profiling, predicting, control and surveillance. *Surveillance & Society*, 12(2), 255–272.

Van Otterlo, M. (2018). From algorithmic black boxes to adaptive white boxes: Declarative decision-theoretic ethical programs as codes of ethics. In *Proceedings of the AAAI/ACM Conference on Artificial Intelligence, Ethics, and Society*.

Van Roosmalen, J., Vranken, H.P.E., & Van Eekelen, M.C.J.D. (2018). Applying deep learning on packet flows for botnet detection. In *SAC*, pages 1629–1636. ACM.

Verbeek, F., Bockenek, J., & Ravindran, B. Highly automated formal proofs over memory usage of assembly code. In *TACAS 2020*, in press.

Verbeek, F., Bockenek, J.A., Bharadwaj, A., Ravindran, B., & Roessle, I. (2019). Establishing a refinement relation between binaries and abstract code. In *MEMOCODE*, pages 17, 1–17:5. ACM.

Vos, T.E.J., & Aho, P. (2017). Searching for the best test. In *2017 IEEE/ACM 10th International Workshop on Search-Based Software Testing (SBST)*, pages 3–4. *

Vos, T.E.J., Kruse, P.M., Condori-Fernández, N., Bauersfeld, S., & Wegener, J. (2015). TESTAR: Tool support for test automation at the user interface level. *Int. J. Inf. Syst. Model. Des.*, 6(3), 46–83.

Wiering, M.A., & Van Otterlo, M. *Reinforcement Learning: State-of-the-Art*. Springer, Berlin, Heidelberg.

5 People

FULL PROFESSORS

prof. dr. Erik Barendsen

Research line(s): Teaching & learning

Prof. dr. Erik Barendsen is a professor of Computing Education at Open University and a professor of Science Education at Radboud University (Nijmegen, NL). He holds a PhD in mathematics and computer science from Radboud University.
His scientific interests include design-based and context-based teaching and learning in computer science and STEM, computational thinking and its integration into the school curriculum, digital literacy, and teachers' practical knowledge (in particular Pedagogical Content Knowledge, PCK).
Erik Barendsen is involved in design and implementation of computing and digital literacy curricula for primary and secondary education in the Netherlands. His research activities include practice-oriented research projects involving schools as project partners. He chairs the program council of the Dutch Dudoc Bèta programme in which STEM teachers carry out PhD projects at eight universities.
Erik Barendsen is an Associate Editor of the Taylor & Francis journal Computer Science Education.

Key publications:

Nijenhuis-Voogt, J., Bayram-Jacobs, D., Meijer, P.C., & Barendsen, E. (2020). Omnipresent yet elusive: Teachers' views on contexts for teaching algorithms in secondary education. *Computer Science Education*. (Accepted for publication)

Stammes, H., Henze, I., Barendsen, E., & De Vries, M.J. (2020). Bringing design practices into chemistry classrooms: Studying teachers' pedagogical ideas in the context of a professional learning community. *International Journal of Science Education*, 1–21. https://doi.org/10.1080/09500693.2020.1717015

Barendsen, E., & Henze, I. (2019). Relating teacher PCK and teacher practice using classroom observation. *Research in Science Education*, 49(5), 1141–1175.

Henze, I., & Barendsen, E. (2019). Unravelling student teachers' pPCK development and the influence of personal factors using authentic data sources. In A. Hume, R. Cooper, & A. Borowski (Eds.), *Repositioning Pedagogical Content Knowledge in teachers' knowledge for teaching science* (pp. 201–221). Springer.

Sentance, S., Barendsen, E., & Schulte, C. (Eds.). (2018). *Computer science education: Perspectives on teaching and learning*. London: Bloomsbury Academic.

Rahimi, E., Barendsen, E., & Henze, I. (2016). Typifying informatics teachers' PCK of designing digital artefacts in Dutch upper secondary education. In A. Brodnik & F. Tort (Eds.), *Informatics in schools: Improvement of informatics knowledge and perception* (pp. 65–77). Springer.

Stegeman, M., Barendsen, E., & Smetsers, S. (2014). Towards an empirically validated model for assessment of code quality. In *Proceedings of the 14th Koli Calling international conference on computing education research* (pp. 99–108).

OU research portal profile: https://research.ou.nl/en/persons/erik-barendsen-2

prof. dr. Marko van Eekelen (faculty dean)

Research line(s): Software quality

After a Math degree in 1981 and a Ph.D. degree in 1988, Prof.dr. Marko van Eekelen became a professor in Software Technology at the Open University (OU) in 2009 via a secondment of 0.3 fte from the Radboud University. Over the years this secondment grew to 0.7 fte through programmeleader for the new OU Master Software Engineering in 2012, and head of the department of Computer Science in 2015. Since January 1, 2020 he is Dean of the then created OU faculty of Science (incorporating 3 departments: Software Science, Information Science and Environmental Sciences).
Bringing theory closer to practice was an important perspective of his long term research. From 2005-2019, 15 years on he was Scientific Director of the Nijmegen part of LaQuSo (a Laboratory of Quality

Software, jointly set up by the Radboud University and Technical University Eindhoven, for analysing concrete applications of research in commercial and industrial contexts on different aspects e.g. security analysis.
In his early scientific work he was active in functional programming and graph rewriting research. Later his research attention moved to analysis and formal verification. In this context his main focus is now resource consumption analysis of software controlled systems, with energy consumption analysis in particular.

Key publications:

Polynomial solutions of algebraic difference equations and homogeneous symmetric polynomials. Shkaravska, M. van Eekelen. *Journal of Symbolic Computation* (2019). Accepted for publication October 3, 2019, 24 pages, In press.

Bernard van Gastel, Marko C.J.D. van Eekelen. *Towards Practical, Precise and Parametric Energy Analysis of IT Controlled Systems*. DICE-FOPARA@ETAPS 2017, 24-37

Rody W.J. Kersten, Bernard van Gastel, Olha Shkaravska, Manuel Montenegro, Marko C.J.D. van Eekelen. ResAna: a resource analysis toolset for (real-time) JAVA. *Concurr. Comput. Pract. Exp. 26*(14), 2432-2455 (2014)

Ken Madlener, Sjaak Smetsers, Marko C.J.D. van Eekelen. A Formal Verification Study on the Rotterdam Storm Surge Barrier. *ICFEM 2010*: 287-302, (2010)

Jongmans Sung-Shik, Lamers Arjan, Eekelen, Marko van. Service-Oriented Architecture and The Button Problem, *23rd International Symposium on Formal Methods - 3rd World Congress on Formal methods (FM2019)*, Porto, Portugal, Springer LNCS 11800, pp 689-706, 2019.

Hans-Dieter Hiep, Olaf Maathuis, Jinting Bian, Frank De Boer, Marko Van Eekelen and Stijn De Gouw. Verifying OpenJDK's LinkedList using KeY. *TACAS 2020*, pp 217-234.

Sven Kiljan, Harald P. E. Vranken, Marko C.J.D. van Eekelen. Evaluation of transaction authentication methods for online banking. *Future Gener. Comput. Syst.* 80, 430-447 (2018)

Vincent van der Meer, Hugo Jonker, Guy Dols, Harm van Beek, Jeroen van den Bos, Marko C.J.D. van Eekelen. File Fragmentation in the Wild: a Privacy-Friendly Approach. *WIFS 2019*, 1-6

Marc Schoolderman, Sjaak Smetsers, Marko C. J. D. van Eekelen. Is Deductive Program Verification Mature Enough to be Taught to Software Engineers? *CSERC 2019*, 50-57.

OU research portal profile: https://research.ou.nl/en/persons/marko-van-eekelen-3

prof. dr. Johan Jeuring

Research line(s): Teaching & learning

Johan Jeuring is a professor of Software Technology for Learning and Teaching, with a minor appointment at the Open University. Besides his appointment at the Open University, he is a professor at, and head of, the Department of Information and Computing Sciences of Utrecht University. He is also CSO of DialogueTrainer, a company that sells software for practicing communication skills. He develops software technology to support learning and teaching, such as feedback in tutoring systems for mathematics, logic, and programming, automatic assessment of mathematics and programming, and artificial intelligence in games. He has been the driving force behind the development of Communicate!, a serious game for practicing communication skills, used by thousands of students and tens of teachers and trainers. His work is used in several learning environments, both in the Netherlands and other European countries.

Key publications:

Alex Gerdes, Bastiaan Heeren, Johan Jeuring, L. Thomas van Binsbergen. Ask-Elle: an Adaptable Programming Tutor for Haskell Giving Automated Feedback. *I. J. Artificial Intelligence in Education* 27(1), 65-100 (2017).

Hieke Keuning, Johan Jeuring, Bastiaan Heeren: A Systematic Literature Review of Automated Feedback Generation for Programming Exercises. *ACM Trans. Comput. Educ. 19*(1), 3:1-3:43 (2019).

OU research portal profile: https://research.ou.nl/en/persons/johan-jeuring-2

prof. dr. Tanja Vos (program leader bachelor CS)

Research line(s): Software quality

Tanja E.J. Vos is a full professor at the Open University (Netherlands) and an associate professor at the Universitat Politècnica de València (Spain). For over 20 years she has been teaching and researching in the area of software testing. She has worked with many companies on automated testing projects in an industrial setting. She is currently project lead for the TESTAR.org, a new approach for automated testing at the Graphical User Interface level that ultimately will learn what the best way is to test software.
Tanja has successfully coordinated the EU-funded projects (FITTEST, EvoTest) related to software testing and has been involved in various Erasmus and Leonardo initiatives that try to help business understand academia and vice versa. She started the Software Testing Innovation Alliance in Spain and is now involved in the European Alliance. Currently, she is part of the Dutch consortium of the ITEA TESTOMAT project (2017-2020) researching and developing the next level of test automation, and the ITEA IVVES project on artificial intelligence and testing.
She is part of the Spanish consortium in the H2020 DECODER project and the iv4XR project.

Key publications:

Floren de Gier, Davy Kager, Stijn de Gouw, Tanja E. J. Vos. Offline Oracles for Accessibility Evaluation with the TESTAR Tool. *RCIS 2019*, 1-12.

Anna Isabel Esparcia-Alcázar, Francisco Almenar, Tanja E.J. Vos, Urko Rueda. Using genetic programming to evolve action selection rules in traversal-based automated software testing: results obtained with the TESTAR tool. *Memetic Comput. 10*(3), 257-265 (2018).

Tanja E. J. Vos, Peter M. Kruse, Nelly Condori-Fernández, Sebastian Bauersfeld, Joachim Wegener. TESTAR: Tool Support for Test Automation at the User Interface Level. *IJISMD 6*(3), 46-83 (2015).

Alexander Elyasov, Wishnu Prasetya, Jurriaan Hage, Urko Rueda, Tanja E. J. Vos, Nelly Condori-Fernández: AB=BA: execution equivalence as a new type of testing oracle. *SAC 2015*, 1559-1566.

Sebastian Bauersfeld, Tanja E.J. Vos, Nelly Condori-Fernández, Alessandra Bagnato, Etienne Brosse: Evaluating the TESTAR tool in an industrial case study. *ESEM 2014*, 4:1-4:9.

OU research portal profile: https://research.ou.nl/en/persons/tanja-vos-3

ASSOCIATE PROFESSORS

dr. Stefano Bromuri

Research line(s): Artificial intelligence

Dr. Stefano Bromuri graduated with an MSc in ICT Engineering from the University of Bologna (2006) and obtained a PhD in Computer Science (focused on AI) from Royal Holloway University of London (2009). Between 2010 and 2016, dr. Bromuri worked in the University of Applied Sciences Western Switzerland, focusing on medical informatics projects. Since 2016 he works at the Open University of the Netherlands teaching subjects such as Data Analytics and Deep Learning in regular courses and professionalization ones. Since 2018, Dr. Bromuri collaborates on frequent basis with DHL in projects concerning machine learning and logistics.

Key Publications:

Henkel, Alexander P., Stefano Bromuri, Deniz Iren, Visara Urovi (2020), "Half Human, Half Machine—Augmenting Service Employees with AI for Interpersonal Emotion Regulation," Special Issue on AI and ML in Service Management, *Journal of Service Management*, forthcoming.

Nicola Falcionelli, Paolo Sernani, Albert Brugués de la Torre, Dagmawi Neway Mekuria, Davide Calvaresi, Michael Schumacher, Aldo Franco Dragoni, Stefano Bromuri: Indexing the Event Calculus: Towards practical human-readable Personal Health Systems. *Artif. Intell. Medicine* 96, 154-166 (2019).

Stefano Bromuri, Dynamic heuristic acceleration of linearly approximated SARSA(λ): using ant colony optimization to learn heuristics dynamically. *J. Heuristics 25*(6), 901-932 (2019).

Stefano Bromuri, Damien Zufferey, Jean Hennebert, Michael Ignaz Schumacher: Multi-label classification of chronically ill patients with bag of words and supervised dimensionality reduction algorithms. *J. Biomed. Informatics* 51, 165-175 (2014).

OU research portal profile: https://research.ou.nl/en/persons/stefano-bromuri-2

dr. Bastiaan Heeren (head of department)

Research line(s): Teaching & learning

dr. BJ (Bastiaan) Heeren has a PhD degree in Software Technology (Utrecht University, 2005). He is an associate professor and Head of the Computer Science Department of the Open University of the Netherlands. He is also a guest researcher at Utrecht University with the Software Technology for Learning and Teaching research group. His research interests are functional programming languages in education, and advanced software technology concepts to support teaching and learning. He is the core designer and developer of the Ideas software framework, which uses software and compiler technology for building state-of-the-art components for intelligent tutoring systems (ITS) and learning environments. These components provide automated feedback and hints to students solving stepwise exercises. The Ideas software framework has been used in many problem domains, including high-school mathematics, logic, introductory functional programming, program refactoring, hypothesis testing, and practicing communication skills. His research goal is to further advance the design of modular, reusable software components for intelligent tutoring systems that are domain-general, and to cross-fertilize solutions from one domain to another.

Key publications:

Alex Gerdes, Bastiaan Heeren, Johan Jeuring, and L. Thomas van Binsbergen. Ask-Elle: an adaptable programming tutor for Haskell giving automated feedback. *I. J. Artificial Intelligence in Education*, 27(1), 65-100, 2017.

Bastiaan Heeren and Johan Jeuring. An extensible domain-specific language for describing problem-solving procedures. In *AIED*, volume 10331 of Lecture Notes in Computer Science, pages 77-89. Springer, 2017.

Hieke Keuning, Johan Jeuring, and Bastiaan Heeren. A systematic literature review of automated feedback generation for programming exercises. *TOCE*, 19(1):3:1-3:43, 2019.

Josje Lodder, Bastiaan Heeren, and Johan Jeuring. Generating hints and feedback for Hilbert-style axiomatic proofs. In *SIGCSE*, pages 387-392. ACM, 2017.

Bastiaan Heeren and Johan Jeuring. Feedback services for stepwise exercises. In *Science of Computer Programming*, 88:110-129, Elsevier, 2014.

OU research portal profile: https://research.ou.nl/en/persons/bastiaan-heeren-2

Google Scholar profile: https://scholar.google.nl/citations?user=A_vVgqgAAAAJ

dr. ir. Harald Vranken (program leader master CS and master SE)

Research line(s): Security & Privacy

Dr. ir. HPE (Harald) Vranken has an MSc degree in Information Technology, an MSc degree in Science Education and Communication, a PDEng degree in Information- and Communication Technology, and a PhD degree in Electrical Engineering, all from the Eindhoven University of Technology. He is an associate professor at the Faculty of Science of the Open University of the Netherlands. He is also a visiting lecturer/researcher at the Radboud University (Nijmegen). He is leading the program line on Security & Privacy in the Computer Science research program, and he is coordinating the program line Resilience in the LIRS research program. His teaching and research activities are mainly focused on digital security, covering among others distributed virtual security labs, cyber security, and the application of artificial intelligence in software security and network security. He is also researching the energy consumption of cryptocurrency mining.

Key publications:

Haag, J., Vranken, H., and Van Eekelen, M. 2019. A Virtual Classroom for Cybersecurity Education. In *Transactions on Edutainment XV*, LNCS 11345 (2019): 173–208, Springer.

Kröhnke, L., Jansen, J., and Vranken, H. 2018. Resilience of the Domain Name System: A case study of the .nl-domain. In *Computer Networks* 139 (2018): 136–150, Elsevier.

Kronjee, J., Hommersom, A., and Vranken, H. 2018. Discovering software vulnerabilities using data-flow analysis and machine learning. In Proceedings of the International Conference on Availability, Reliability and Security. (Hamburg, Germany, August 27–30, 2018). ARES 2018. ACM, Article 6.

Van Roosmalen, J., Vranken, H., and Van Eekelen, M. 2018. Applying Deep Learning on Packet Flows for Botnet Detection. In *Proceedings of the Symposium on Applied Computing; Computer Security track*. (Pau, France, April 9–13, 2018). SAC 2018. ACM, 1629-1636.

Vranken, H. 2017. Sustainability of bitcoin and blockchains. In: *Current Opinion in Environmental Sustainability 2017*, 28:1-9, Elsevier.

OU research portal profile: https://research.ou.nl/en/persons/harald-vranken-2

Google Scholar profile: https://scholar.google.com/citations?user=J3ZE7nEAAAAJ

ASSISTANT PROFESSORS

dr. Greg Alpár

Research line(s): Security & privacy

dr. Greg Alpár has two Master's degrees (the first one in mathematics and mathematics education from ELTE Budapest, Hungary, and the second one in applied mathematics in communication and information technologies from the TU Eindhoven, The Netherlands) and a PhD degree in Computer Science (Attribute-based Identity Management) from the Radboud University. Greg is an assistant professor at the Faculty of Science of the Open University of the Netherlands. He is also a visiting researcher at the Radboud University. His teaching focusses on computer networking, security and software engineering, while his research on cryptography, privacy and networking. Furthermore, Greg is active in the Comeniusnetwerk in which he is a member due to his innovative education project, the Open Maths.

Selected publications:

Lueks, W., Hampiholi, B., Alpár, G., & Troncoso, C. (2020). Tandem: Securing Keys by Using a Central Server While Preserving Privacy. In *Privacy Enhancing Technologies (PETs)*, 2020(3).

Alpár, G., & van Hoeve, M. (2019). Towards Growth-Mindset Mathematics Teaching in the Netherlands. *Proc. of Learning Innovations and Quality (LINQ)*, vol 2, 1-17.

Lueks, W., Alpár, G., Hoepman, J. H., & Vullers, P. (2017). Fast revocation of attribute-based credentials for both users and verifiers. *Computers & Security*, 67, 308-323.

Hampiholi, B., Alpár, G., van den Broek, F., & Jacobs, B. (2015). Towards practical attribute-based signatures. In *International Conference on Security, Privacy, and Applied Cryptography Engineering* (pp. 310-328). Springer, Cham.

Vullers, P., & Alpár, G. (2013). Efficient selective disclosure on smart cards using idemix. In IFIP Working Conference on Policies and Research in *Identity Management* (pp. 53-67). Springer, Berlin, Heidelberg.

Alpár, G., Batina, L., & Lueks, W. (2012). Designated attribute-based proofs for RFID applications. In *International Workshop on Radio Frequency Identification: Security and Privacy Issues* (pp. 59-75). Springer, Berlin, Heidelberg.

OU research portal profile: https://research.ou.nl/en/persons/g-alpár

dr. Fenia Aivaloglou

Research line(s): Teaching & learning

OU research portal profile: https://research.ou.nl/en/persons/efthimia-aivaloglou-2

dr. Bernard van Gastel Research line(s): Software quality; Security & privacy OU research portal profile: https://research.ou.nl/en/persons/bernard-van-gastel-2	
dr. Stijn de Gouw Research line(s): Software quality dr CPT (Stijn) de Gouw obtained his Master's degree at Leiden University (2009) and a PhD degree (2013) at the Centrum voor Wiskunde & Informatica (CWI) and Leiden University. He has participated in three European projects: as Master student in the EU FP6 project Credo, as PhD-student in the FP7 European HATS project (completed 2013, the project was rated 'excellent') and as workpackage leader and task leader of three tasks in the FP7 project Envisage (completed October 2016, rated 'excellent'). Stijn also was a postdoc in a 2-year NWO PPS between CWI and the company Fredhopper to develop tools for monitoring cloud applications. This evolved into a permanent position as Research Engineer at Fredhopper, where he was responsible for EU research projects. Since 2016 Stijn is Assistant Professor at the Open University (full-time appointment) and guest researcher at CWI. His recent research focuses on developing and applying novel tool-supported analysis techniques for complex case studies, particularly widely used library code. He developed extensions to the KeY Java theorem prover and developed the SAGA tool for run-time monitoring of Java and cloud applications. Stijn was the proof architect in the mechanic verification of TimSort, which resulted in updates to many standard libraries and frameworks. He was also the proof architect of the sorting algorithms Radix Sort and Counting Sort in Java. This marked the first time stability of a Java sorting algorithm was proven. Stijn's teaching activities mainly concern programming languages and the theory of computation. He is a member of the education committee (opleidingscommissie) Bachelor Informatica en Informatiekunde. OU research portal profile: https://research.ou.nl/en/persons/stijn-de-gouw-2 Google scholar profile: https://scholar.google.com/citations?user=a9sw_L0AAAAJ	

dr. Arjen Hommersom

Research line(s): Artificial intelligence

Arjen Hommersom has a Master's degree in computer science (University of Utrecht) and received his PhD degree from the University of Nijmegen for his work on quality assurance of clinical practice guidelines. Since 2015, he is an assistant professor at the Open University, the Netherlands and a researcher at the Institute of Computing and Information Sciences at Radboud University Nijmegen, the Netherlands. His research interests include knowledge representation and reasoning techniques, in particular probabilistic graphical models and probabilistic logic programming, applied in the health domain. He is currently involved in the ZonMW Active4Life project, which focuses on the analysis of large-scale datasets in order to develop new interventions for increasing physical activity among (healthy and diseased) adults.

Key publications:

Bueno, M.L., Hommersom, A.J., Lucas, P.J.F., & Janzing, J. A probabilistic framework for predicting disease dynamics: A case study of psychotic depression. *Journal of biomedical informatics*, 95, 103232, 2019.

Butz, R., Hommersom, A.J., & van Eekelen, M. Explaining the most probable explanation. In *International Conference on Scalable Uncertainty Management*, pages 50–63. Springer, 2018.

Liu, M., Stella, F., and Hommersom, A.J., Lucas, P.J.F., Boer, L, & Bischoff, E. A comparison between discrete and continuous time Bayesian networks in learning from clinical time series data with irregularity. *Artificial intelligence in medicine*, 95, 104–117, 2019.

Amirkhani, H., Rahmati, M., Lucas, P.J.F., & Hommersom, A.J. (2016). Exploiting experts' knowledge for structure learning of Bayesian networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(11), 2154-2170.

Bueno, M.L., Hommersom, A., Lucas, P.J., & Linard, A. (2017). Asymmetric hidden Markov models. *International Journal of Approximate Reasoning*, 88, 169-191.

OU research portal profile: https://research.ou.nl/en/persons/arjen-hommersom-2

Google Scholar profile: https://scholar.google.nl/citations?user=qwYoEBUAAAAJ

dr. ir. Sung-Shik Jongmans

Research line(s): Software quality

Dr. ir. Sung-Shik Jongmans is assistant professor at Open University of the Netherlands and researcher in the Formal Methods group at Centrum Wiskunde & Informatica (CWI), Amsterdam. He has a PhD degree from Leiden University (2016, cum laude) and was visiting researcher at Imperial College London (2017-2019). His research interests include concurrency and programming languages; at the intersection of theory and practice, he currently works on design and implementation of DSL techniques for code generation, optimization, and verification of synchronization and communication protocols. He was awarded an NWO Rubicon grant (2016-2 round) and an NWO Veni grant (2018 round).

Key publications:

Ruben Hamers, Sung-Shik Jongmans. Discourje: Runtime Verification of Communication Protocols in Clojure. *TACAS 2020*. LNCS 12078, pages 266-284.

Sung-Shik Jongmans, Nobuko Yoshida. Exploring Type-Level Bisimilarity towards More Expressive Multiparty Session Types. *ESOP 2020*. LNCS 12075, pages 251-279.

Sung-Shik Jongmans. Toward New Unit-Testing Techniques for Shared-Memory Concurrent Programs. ICECCS 2019. Pages 164-169.

Sung-Shik Jongmans, Arjan Lamers, Marko van Eekelen. SOA and the Button Problem. *FM 2019*. LNCS 11800, pages 689-706.

David Castro, Raymond Hu, Sung-Shik Jongmans, Nicholas Ng, Nobuko Yoshida. Distributed Programming using Role-Parametric Session Types in Go. *POPL 2019*. PACMPL 3, paper 29.

OU research portal profile: https://research.ou.nl/en/persons/sung-shik-jongmans-2

dr. ir. Hugo Jonker

Research line(s): Security & privacy

Hugo Jonker is assistant professor in security and privacy at the Open University of the Netherlands. He is a guest researcher at the Digital Security group of the Radboud University Nijmegen. His research focuses on how security / privacy in the real world both affects and is affected by security / privacy in the virtual world. He has published in a variety of fields, including cryptography, browser fingerprinting, online tracking, privacy and security in e-voting, digital forensics, security of smart contracts, and website security.

Key publications:

A Contemporary Investigation of NTFS File Fragmentation. V. van der Meer, H. Jonker and J. van den Bos. *Digital Investigation*, 2020. To appear.

Shepherd: a generic approach to automating website login. H. Jonker, S. Karsch, B. Krumnow and M. Sleegers. In *Proc. 2nd Workshop on Measurements, Attacks and Defenses for the Web (MADWEB'20)*. IEEE, pp. 1–10, 2020.

Fingerprint surface-based detection of web bot detectors. H. Jonker, B. Krumnow and G. Vlot. In *Proc. 24th European Symposium on Research in Computer Security Part II (ESORICS'19)*. Springer, LNCS 11736, pp. 586–605, 2019.

Investigating Fingerprinters and Fingerprinting-alike Behaviour of Android Applications. C. Ferreira Torres and H. Jonker. In *Proc. 23rd European Symposium on Research in Computer Security (ESORICS'18)*. Springer, LNCS 11023, pp. 60-80, 2018.

Defining Verifiability in e-Auction Protocols. J. Dreier, H.L. Jonker and P. Lafourcade. In *Proc. 8th ACM Symposium on Information, Computer and Communications Security (ASIACCS'13)*. ACM, pp. 547-552, 2013.

OU research portal profile: https://research.ou.nl/en/persons/hugo-jonker-2

Google Scholar profile: https://scholar.google.com/citations?user=0qhFovsAAAAJ

dr. ir. Arjan Kok

Research theme: Teaching & learning

dr. ir. A.J.F. (Arjan) Kok has a Master's degree in Computer science (Delft University of Technology, NL), and a PhD degree in Computer graphics (Delft University of Technology). He is an assistant professor in the Department of Computer Science at the Faculty of Science of the Open University of the Netherlands. His current teaching and research activities are focused on object-oriented design and programming.

OU research portal profile: https://research.ou.nl/en/persons/arjan-kok-2

drs. Josje Lodder

Research line(s): Teaching & learning

Drs. J. S. (Josje Lodder) has a Master's degree in Mathematics (University of Utrecht NL). The main subject of her research is design and use of tools for teaching logic.

Key publications:

Lodder, J., Heeren, B., and Jeuring, J. (2016). A domain reasoner for propositional logic. *Journal of Universal Computer Science*, *22*(8), 1097-1122.

Lodder, J., Heeren, B., and Jeuring, J. (2017). Generating Hints and Feedback for Hilbert-style Axiomatic Proofs. In Caspersen, M. E., Edwards, S. H., Barnes, T., and Garcia, D. D., editors, *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, Seattle, WA, USA, March 8-11, 2017, pages 387-392. ACM.

Lodder, J., Heeren, B., and Jeuring, J. (2019). A comparison of elaborated and restricted feedback in LogEx, a tool for teaching rewriting logical formulae. *Journal of Computer Assisted Learning*, *35*(5):620-632.

Lodder, J., Heeren, B., and Jeuring, J. (2020). Providing Hints, Next Steps and Feedback in a Tutoring System for Structural Induction. *Electronic Proceedings in Theoretical Computer Science*, 313, 17-34.

OU research portal profile: https://research.ou.nl/en/persons/josje-lodder-2

dr. Clara Maathuis

Research line(s): Security & Privacy; Artificial Intelligence

Clara Maathuis is Assistant Professor at OU with technical-military background doing studies in Military Cyber Operations & AI (PhD), Artificial Intelligence (MSc), and Computer Science & Automatic Control Engineering (BSc). Her teaching activities are in the fields of Artificial Intelligence and Cyber Security. She conducts transdisciplinary research in the fields of Artificial Intelligence, Cyber Security, military technologies, social manipulation, law enforcement, and gaming. In the industry, she worked as Senior Software Engineer on data, voice, and automation solutions in telecom and control systems industries.

Maathuis, C. et al. (2020). Decision Support Model for Effects Estimation and Proportionality Assessment for Targeting in Cyber Operations. *Journal of Defence Technology*, Elsevier.

Maathuis, C. et al. (2018). Developing a Cyber Operations Computational Ontology. *Journal of Information Warfare 17*(3), 32-49.

Maathuis, C. et al. (2018). *A Knowledge-Based Model for Assessing the Effects of Cyber Warfare*. 12th NATO Conference on Operations Research and Analysis 2018.

OU research profile: https://research.ou.nl/en/persons/clara-maathuis

Google Scholar profile: https://scholar.google.com/citations?user=WqR3BVwAAAAJ

dr. ir. Martijn van Otterlo

Program line(s): Artificial intelligence

Dr. ir. Martijn van Otterlo is an assistant professor in artificial intelligence (AI) in the computer science department of the Open University (OU, The Netherlands) and a guest researcher at the software science group at the computer science department at Radboud University (Nijmegen). At OU Martijn leads the development of a new master programme in Artificial Intelligence. Martijn obtained his PhD from the University of Twente (Netherlands, 2008) with a dissertation on expressive knowledge representation in machine learning from evaluative feedback (e.g. reinforcement learning). He published two books on such adaptive decision-making algorithms (2009 IOS Press; 2012 Springer, together with Dr. Wiering). Martijn has worked on robotics, vision, language, sensor-based systems, data science, and others, and held academic positions in Freiburg (Germany), Leuven (Belgium), Nijmegen and Amsterdam (The Netherlands). His second research interest, which arose from his expertise in AI, concerns the ethics and implications of adaptive algorithms on privacy, surveillance and society. He is particularly interested in "value alignment": making intelligent machines that respect, understand, and act according to, human values, and in "explainability": making systems explain their decisions in suitable terms. In both these directions he employs the technical tools of reinforcement learning and formal knowledge representation. He has served as committee member and reviewer for dozens of international journals and conferences on machine learning, data science and artificial intelligence, and he is member of the expert panel of the European Science Foundation. Other keywords include photography, Lego, public libraries, geopolitics, popular culture, martial arts: even more information can be found at http://martijnvanotterlo.nl.

OU research portal profile: https://research.ou.nl/en/persons/martijn-van-otterlo

dr. ir. Harrie Passier

Research line(s): Teaching & learning

Harrie Passier is an assistant professor at the Open University of the Netherlands, Beta faculty, department Computer Science. He has experience in teaching object-oriented programming, functional programming, web programming, object-oriented design, software engineering, and didactics of informatics. His research interests are about the use of procedural guidance in OO software development, testing and refactoring. He is a member of the international Computer Science Education Research Conference (CSERC).

Key publications:

Bijlsma, H.J.M. Passier, H.J. Pootjes, and S. Stuurman. Template Method test pattern. in *Information Processing Letters*, *139*, 8-12, 2018

Sylvia Stuurman, Harrie Passier, Erik Barensen, Analyzing Students' Software Redesign Strategies, in Koli Calling '16: *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*, Finland, ACM, 2016

OU research portal profile: https://research.ou.nl/en/persons/harrie-passier-2

dr. Ebrahim Rahimi

Program lines: Teaching & learning

dr. Ebrahim Rahimi has a Master's degree in Software Engineering (Tehran Polytechnic, Iran) and a PhD degree in IT (designing of e-Learning systems) (TU Delft, NL). He is an assistant professor at the Department of Computer Sciences at the Faculty of Science of the Open University of the Netherlands. Before joining the Open University, Ebrahim worked at the Department of Computer Science of the Radboud University on a 3-year post-doctorate NRO-funded research to develop course materials to teach algorithmic thinking high school students.

His teaching activities are mainly focused on object-oriented programming, model-driven development, design patterns, software quality, research methods in computer science, and academic writing. His research activities and interests include e-learning design, personal learning environments, Computation education, design-based education in computer science, computational thinking, pedagogical content knowledge (PCK), and code quality. Further, he cooperates as the scientific reviewer with several scientific journals, including Computer Science Education, Computers & Education, Algorithms, ACM Transactions on Computing Education, Multimodal Technologies and Interaction, Academic Journals, and European Journal of Engineering Education. Ebrahim served as the program chair of the 8th Computer Science Education Research Conference (CSERC' 19) and the co-chair of CSERC' 20.

Key publications:

Rahimi, E., Van den Berg, J., & Veen, W. (2015). Facilitating student-driven constructing of learning environments using Web 2.0 personal learning environments, *Computers & Education*, vol 81, pp. 235–246.

Rahimi, E., Van den Berg, J., & Veen, W. (2014). A learning model for enhancing the student's control in educational process using Web 2.0 personal learning environments, *British journal of educational technology (BJET)*, vol 46, pp. 780–792.

Rahimi, E., van den Berg, J., & Veen, W. (2013). Investigating teachers' perception about the educational benefits of Web 2.0 personal learning environments. *ELearning Papers*, no: 35.

Rahimi, E., van den Berg, J., & Veen, W. (2014). A pedagogy-driven framework for integrating Web 2.0 tools into educational practices and building personal learning environments, *Journal of Literacy and Technology*, vol 15 (2), pp. 54-79. ISSN: 1535-0975.

Rahimi E., Barendsen E., Henze I. (2016) Typifying Informatics Teachers' PCK of Designing Digital Artefacts in Dutch Upper Secondary Education. In: Brodnik A., Tort F. (eds) Informatics in Schools: Improvement of Informatics Knowledge and Perception. ISSEP 2016. *Lecture Notes in Computer Science*, vol 9973. Springer, Cham.

Rahimi, E., Barendsen, E., & Henze, I. (2018, October). An instructional model to link designing and conceptual understanding in secondary computer science education. In *Proceedings of the 13th Workshop in Primary and Secondary Computing Education* (pp. 1-4).

Rahimi E., Barendsen E., Henze I. (2017) Identifying Students' Misconceptions on Basic Algorithmic Concepts Through Flowchart Analysis. In: Dagiene V., Hellas A. (eds). *Informatics in Schools: Focus on Learning Programming*. ISSEP 2017. Lecture Notes in Computer Science, vol 10696. Springer, Cham.

OU research portal profile: https://research.ou.nl/en/persons/ebrahim-rahimi

Google Scholar profile: https://scholar.google.com/citations?user=wRl2ps4AAAAJ

dr. Stefano Schivo

Research line(s): Security & privacy

Dr. Stefano Schivo has a Master's degree in Bioinformatics (University of Trento, IT) and a PhD degree in Information and Communication Technology (University of Trento, IT). He is an assistant professor at the Department of Computer Science at the Faculty of Science of the Open University of the Netherlands. He has been involved in a number of projects ranging from bioinformatics to security, all with the common aspect of applying the strengths of formal methods to other research fields.

Key publications:

Schivo, S., Khurana, S., Govindaraj, K., Scholma, J., Kerkhofs, J., Zhong, L., Huang, X., Pol, J.C. van de, Langerak, R., Wijnen, A.J. van, Geris, L., Karperien, M., Post, J.N. (2020). ECHO, the executable CHOndrocyte: A computational model to study articular chondrocytes in health and disease. *Cellular Signalling*, 68, [109471].

Khurana, S., Bokkers, A., Geijs, D.J., Schivo, S., Karperien, M., & Post, J.N. (2019). In Silico Validation of a cartilage specific circadian clock: mutation of BMAL1 increased MMP expression. *Osteoarthritis and Cartilage*, 27(S1), S193-S194.

Kumar, R., Schivo, S., Ruijters, E.J.J., Yildiz, B.M., Huistra, D.J., Brandt, J., Rensink, A., & Stoelinga, M.I.A. (2018). Effective Analysis of Attack Trees: A Model-Driven Approach. In *Fundamental Approaches to Software Engineering: FASE 2018* (pp. 56-73). Springer. Lecture Notes in Computer Science, Vol. 10802

Ruijters, E., Schivo, S., Stoelinga, M., & Rensink, A. (2017). Uniform analysis of fault trees through model transformations. In *2017 Annual Reliability and Maintainability Symposium (RAMS)* Institute of Electrical and Electronics Engineers Inc.

Schivo, S., Yildiz, B.M., Ruijters, E.J.J., Gerking, C., Kumar, R., Dziwok, S., Rensink, A., & Stoelinga, M.I.A. (2017). How to Efficiently Build a Front-End Tool for UPPAAL: A Model-Driven Approach. In *Dependable Software Engineering. Theories, Tools, and Applications: Third International Symposium*, SETTA 2017, Changsha, China, October 23-25, 2017, Proceedings (pp. 319-336). Springer. Lecture Notes in Computer Science, Vol. 10606

Schivo, S., & Langerak, R. (2017). Discretization of Continuous Dynamical Systems Using UPPAAL. In *ModelEd, TestEd, TrustEd* (pp. 297-315). Springer. Lecture Notes in Computer Science

Schivo, S., Scholma, J., Wanders, B., Urquidi Camacho, R. A., van der Vet, P. E., Karperien, H. B. J., ... Post, J. N. (2014). Modelling biological pathway dynamics with Timed Automata. *IEEE journal of biomedical and health informatics*, *18*(3), 832-839.

OU research portal profile: https://research.ou.nl/en/persons/stefano-schivo

dr. ir. Sylvia Stuurman

Research line(s): Teaching & learning

Key publications:

Sylvia Stuurman, Harrie J.M. Passier, Frédérique Geven, Erik Barendsen. Autism: Implications for Inclusive Education. *Proceedings of the 8th Computer Science Education Research Conference (CSERC) 2019*, pages 15-25, ACM.

Sylvia Stuurman, Harrie Passier, Erik Barendsen. Analyzing Student's Software Redesign Strategies. *Proceedings of the 16th International Conference on Computing Education Research*, Koli Calling, 2016, pages 110-119, ACM.

Harrie J.M. Passier, Sylvia Stuurman, Harold Pootjes. Beautiful Javascript: How to guide students to create good and elegant code. *Proceedings of the fourth Computer Science Education Research Conference 2014*, pages 65-76, ACM.

OU research portal profile: https://research.ou.nl/en/persons/sylvia-stuurman-2

dr. Frank Tempelman

Research line(s): Artificial intelligence

OU research portal profile: https://research.ou.nl/en/persons/frank-tempelman

dr. Freek Verbeek

Research line(s): Software quality

Freek Verbeek received his PhD degree in computer science from the Radboud University Nijmegen in the Netherlands in 2013. Currently, he is assistant professor at the Open University of the Netherland and adjunct assistant professor at Virginia Polytechnic Institute and State University. His research interests include the application of formal methods to practical case-studies. His work has aimed at using combinations of theorem proving techniques, model checking and SMT solving to prove absence of deadlocks in Networks-on-Chips. He has worked in the EUROMILS project where he aimed at using theorem proving to show the absence of unallowed information flows in an industrial operating system for automotive and avionics use-cases. His current research focusses on bottom-up verification of x86 binaries: formal verification of properties such as return-address integrity over a binary without assuming availability of source-code.

Key Publications:

F. Verbeek et al. A Compositional Approach for Verifying Protocols Running on On-Chip Networks. *IEEE Transactions on Computers* 67.7 (2017): 905-919.

F. Verbeek, Joshua Bockenek and Binoy Ravindran. Highly Automated Formal Proofs over Memory Usage of Assembly Code. *Proceedings of TACAS'20*.

F. Verbeek and J. Schmaltz. On Necessary and Sufficient Conditions for Deadlock-Free Routing in Wormhole Networks. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, February 2011.

OU research portal profile: https://research.ou.nl/en/persons/freek-verbeek-2

dr. ir. Alaaeddin Swidan

Research line(s): Teaching & learning

Dr. AAS (Alaaeddin) Swidan a master degree in Management of ICT (Cardiff Metropolitan University, UK) and a PhD degree in Computer Science (Technische Universiteit Delft, NL). I am currently an assistant professor in the Computer Science department at the Faculty of Science of the Open University of the Netherlands. I am also a guest researcher in the Programming Education Research Lab (PERL) at Leiden University (NL). My current teaching activities include teaching software engineering courses such as Software Architecture and Design Patterns. I have developed two research lines in the past years. One relates to the analysis of software artifacts created by end-user programmers, such as spreadsheets. The aim is to improve the quality of such artifacts by adopting methods and techniques from the Software Engineering domain. The second line is focused on Computer Science education for young, school-age learners. The aim is to identify and overcome challenges that those learners face when they start learning how to program. Learning programming at this early ages is known to be very important for developing certain Computational Thinking skills as well as shaping future conceptions about Computer Science in general including misconceptions and career orientation.

Key publications:

Swidan and F. Hermans, "The Effect of Reading Code Aloud on Comprehension: An Empirical Study with School Students," in *Proceedings of the ACM Conference on Global Computing Education, CompEd 2019*, Chengdu, Sichuan, China, May 17-19, 2019, 2019, pp. 178–184

Swidan, F. Hermans, and M. Smit, "Programming Misconceptions for School Students," in *Proceedings of the 2018 ACM Conference on International Computing Education Research, ICER 2018*, Espoo, Finland, August 13-15, 2018, 2018, pp. 151–159.

F. Hermans, A. Swidan, E. Aivaloglou, and M. Smit, "Thinking out of the box: comparing metaphors for variables in programming education," in *Proceedings of the 13th Workshop in Primary and Secondary Computing Education, WiPSCE 2018*, Potsdam, Germany, October 04-06, 2018, 2018, pp. 8:1–8:8.

Swidan and F. Hermans, "Programming Education to Preschoolers: Reflections and Observations from a Field Study," in *Proceedings of the 28th Annual Workshop of the Psychology of Programming Interest Group, PPIG 2017*, Delft, The Netherlands, July 1-3, 2017, 2017, p. 7.

Swidan, A. Serebrenik, and F. Hermans, "How do Scratch Programmers Name Variables and Procedures?," in *17th IEEE International Working Conference on Source Code Analysis and Manipulation, SCAM 2017*, Shanghai, China, September 17-18, 2017, 2017, pp. 51–60.

Swidan, F. Hermans, and R. Koesoemowidjojo, "Improving the Performance of a Large Scale Spreadsheet: A Case Study," in *IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering, SANER 2016*, Suita, Osaka, Japan, March 14-18, 2016 - Volume 1, 2016, pp. 673–677.

Swidan and F. Hermans, "Semi-automatic Extraction of Cross-Table Data from a Set of Spreadsheets," in *End-User Development - 6th International Symposium, IS-EUD 2017*, Eindhoven, The Netherlands, June 13-15, 2017, Proceedings, 2017, vol. 10303, pp. 84–99.

OU research portal profile: https://research.ou.nl/en/persons/alaaeddin-swidan

Google Scholar profile: https://scholar.google.nl/citations?user=NDkRW44AAAAJ&hl=en

LECTURERS

ir. Evert van de Vrie	
dr. Nikè van Vugt-Hage dr. N. (Nikè) van Vugt has a Master's degree in Fundamental Computer Science and a PhD degree in Mathematics and Natural sciences. The subject of her PhD thesis was 'Models of Molecular Computing': using formal language theory to describe certain natural processes that might be used for computing. She does not work as a researcher anymore, but sometimes contributes to projects of others (SERF). OU research portal profile: https://research.ou.nl/en/persons/nike-van-vugt-hage-2	
drs. Petra Leeuwestein-Verbeek	

RESEARCHERS

dr. Pekka Aho Research line(s): Software quality Dr. Pekka Aho received his MSc and PhD degrees on computer engineering from University of Oulu, Finland. He is a senior researcher working on international industry-academia collaboration projects related to automated software testing. He is supervising BSc and MSc graduation projects with Prof. Dr. Tanja Vos related to automated testing and open source TESTAR tool. Google Scholar profile: https://scholar.google.fi/citations?user=u0AgPdYAAAAJ	
Daniele Di Mitri, MSc Research line(s): Teaching & learning OU research portal profile: https://research.ou.nl/en/persons/daniele-di-mitri-2	

dr. Fabian van den Broek

Research line(s): Security & privacy

Fabian van den Broek is a researcher at the computer science department of the Open University and a guest researcher at the digital security group of the Radboud university. His research focuses on privacy and authentication and on maturing the IRMA (I Reveal My Attributes) technology. The IRMA technology offers a way to authenticate users based
on a subset of their attributes, which results in a very flexible and privacy-friendly eco system. This work is in part financed by the Privacy by Design Foundation.

Fabian finished his PhD (Radboud University) on the security of wireless communication technologies such as GSM, GPRS and UMTS. For his PhD he looked into both theoretical and practical questions on security and privacy in the access part of mobile technologies. He also looked at the security of systems utilizing mobile communication technology within their architecture and he proposed improvements to the wireless
standards which prevent fake-base station attacks (IMSI catchers).

After his Phd Fabian was involved in the OYOI (Own Your Own Identity) project were the IRMA technology was successfully moved from smart cards to mobile phones. He also participated in the European C-DAX project, working on the security aspects of C-DAX, an ICN (Information-Centric Networking) architecture for smart grids.

Key publications:

"IRMA: practical, decentralized and privacy-friendly identity management using smartphones", Gergely Alpár, Fabian van den Broek, Brinda Hampiholi, Bart Jacobs, Wouter Lueks and Sietse Ringers. *HotPETs 2017, 10th Workshop on Hot Topics in Privacy Enhancing Technologies*, Minneapolis, USA 2017.

"Defeating IMSI Catchers", Fabian van den Broek, Roel Verdult and Joeri de Ruiter. *CCS 2015, ACM Conference on Computer and Communications Security 2015*, 340-351, ACM, 2015.

OU research portal profile: https://research.ou.nl/en/persons/fabian-van-den-broek-2

Nico Naus, MSc

Research line(s): Software quality

OU research portal profile: https://research.ou.nl/en/persons/nico-naus

dr. Sietse Ringers

Research line(s): Security & privacy

OU research portal profile: https://research.ou.nl/en/persons/sietse-ringers-2

PHD STUDENTS

Luc Edixhoven, MSc Research line(s): Software quality Luc Edixhoven graduated with an MSc in Computer Science from Leiden University in 2019, and started as a PhD student at the Open University later that year under the supervision of Sung-Shik Jongmans. OU research portal profile: https://research.ou.nl/en/persons/luc-edixhoven	
Olivia Rodríguez Valdés, MSc Research line(s): Software quality Olivia Rodriguez Valdes graduated with an MSc in Computer Science from the University of Havana (2017). Between 2017 and 2019, she worked as an teaching assistant at Technological University of Havana, in the courses of Introduction to Programming and Mathematical Analysis to the faculties of Telecommunications Engineering and Electrical Engineering, respectively. She is currently a PhD Candidate at Open Universiteit, researching in the field of Artificial Intelligence applied to Automatic Testing. OU research portal profile: https://research.ou.nl/en/persons/olivia-rodr%C3%ADguez-valdés	
Timmy Weerwag, MSc Research line(s): Software quality OU research portal profile: https://research.ou.nl/en/persons/timmy-weerwag	

EXTERNAL PHD STUDENTS

Raphaela Butz, MSc (co-supervised by Arjen Hommersom)
Xi Cheng, MSc (supervised by Stefano Bromuri)
Ed van Doorn, MSc (supervised by Sylvia Stuurman and Marko van Eekelen)
Francesco Gibellini, MSc (supervised by Stefano Bromuri)
Hans-Dieter Hiep, MSc (co-supervised by Stijn de Gouw)
Hieke Keuning, MSc (supervised by Johan Jeuring and Bastiaan Heeren)
Benjamin Krumnow, MSc (supervised by Hugo Jonker and Harald Vranken)
Arjan Lamers, MSc (supervised by Marko van Eekelen and Sung-Shik Jongmans)
ir. Vincent van der Meer (supervised by Hugo Jonker and Marko van Eekelen)
Vreda Pieterse, MSc (supervised by Marko van Eekelen)
Andreas Schuderer, magister (supervised by Stefano Bromuri)
Martijn Stegeman, MSc (supervised by Erik Barendsen)
Simone Tummers, MSc (co-supervised by Arjen Hommersom)
Koen Weterings, MSC (supervised by Stefano Bromuri and Marko van Eekelen).