

Software Technology Research Plan 2010-2015

School of Computer Science, Open Universiteit

May 9, 2011

1 Research Lines

The software technology domain distinguishes the following two research lines.

- **Software Technology for Teaching and Learning.** This line has a significant history at the Open Universiteit. It is supervised by Prof. Johan Jeuring.
- **Software Technology for Quality Improvement.** This line just started at the Open Universiteit in 2009. It is supervised by Prof. Marko van Eekelen.

There is not a strict separation between the researchers that contribute to the two lines. Researchers can be in just one line but also in two lines.

2 Software Technology for Teaching and Learning

2.1 Introduction

There is an increasing need for research and expertise at the academic level on the role software technology plays in teaching and learning. E-learning applications, intelligent tutoring systems, collaboration environments, serious games, and many more software artefacts are used in teaching. The Software Technology group investigates the relation between software technology and teaching and learning.

Tools for learning are developing and spreading fast, but are often hampered by technological limitations and problems. For example, many tools that support students with solving mathematical exercises do not give feedback to students when they make an error. And hardly any tool reports common errors or problems of students back to the teacher. Fundamental concepts from computer science can make an important contribution to technology for learning. Technology for learning has received a lot of attention in recent years, and there are a number of funding agencies and/or programmes, such as the Dutch Kennisnet en SURF, and the European eContentPlus programme with a focus on further developing technology for teaching and learning.

2.2 Context

The focus of our research is on tools that support learning, in particular learning of procedural skills. In many subjects students have to acquire procedural skills. Problems in mathematics are often solved using a standard procedure, such as for example solving a system of linear equations by subtracting equations from top to bottom, and then substituting variables from bottom to top. Problems in computer science, physics, chemistry, electrotechnics, and other disciplines, often require procedural skills as well. Furthermore, procedural skills appear at any educational level.

VanLehn [26] (p. 7) says the following about learning procedural skills

“Learning of subtraction procedures (and perhaps other procedures as well) is driven not by natural-language recipes but instead by examples of the execution of the to-be-acquired procedure.”

Students infer a procedure inductively from examples. Of course, many of them make mistakes when inferring a procedure, because examples almost never completely and uniquely determine a procedure. Anderson [1] formulates a similar hypothesis for learning procedural skills.

It is important to see the procedure itself after practicing ([2], p. 65):

“Research studies generally provide strong support for the benefits of helping students represent their experiences at levels of abstraction that transcend the specificity of particular contexts and examples.”

and to see different procedures in action ([2], p. 100):

“The broader the range of strategies that children know and can appreciate where they apply, the more precisely they can shape their approaches to the demands of particular circumstances.”

Combining and automatising procedures, and obtaining a deeper understanding of them is challenging for education ([2], p. 139):

“A challenge for the design of knowledge-centered environments is to strike the appropriate balance between activities designed to promote understanding and those designed to promote the automaticity of skills necessary to function effectively without being overwhelmed by attentional requirements.”

Feedback plays an essential role in learning procedural skills ([2], p. 174):

“Undergraduate engineering students were instructed to write qualitative strategies for solving problems before attempting to solve them. The strategies consisted of a coherent verbal description of how a problem could be solved: the major principle to be applied; the justification for why the principle was applicable; and the procedures for applying the principle (what, why, and how). They performed significantly better. [...] These examples demonstrate the importance of deliberate practice and of having a “coach” who provides feedback for ways of optimizing performance.”

In our research we try to use concepts from software technology to improve learning, in particular learning procedural skills. An important focus area is diagnosing student errors, and giving feedback to students when solving exercises stepwise. In our research we address two of the five factors underpinning successful learning [17]: learning by doing, and learning through feedback.

2.3 History

In 2003 the software technology group started with research in the context of the research plan 'Software technology for e-learning' (STEL). We have worked on several aspects related to exercise assistants: tools that support solving exercises. In particular, we have focussed on reasoning about the domains of the exercise assistants, and on how we can give better feedback to students using exercise assistants. We have worked on:

- providing feedback to students applying rules in exercise assistants [16, 13].
- developing a language for specifying strategies for solving exercises [10].
- developing technology for tracking the steps a student takes towards a solution [8].
- developing ways to efficiently model mathematical rewrite rules and strategies using our formalism [9].
- developing services so that external tools can use our feedback services [6].
- using strategies for recognizing steps in the process of constructing functional programs [5].
- formulating our software in terms of generic programs to make adding a new domain as simple as possible [11].

Some of these research projects are finished to a large extent, such as the language for specifying strategies for solving exercises, the technology for tracking the steps a student takes towards a solution, and the development of feedback services. Other projects need more work in the coming years.

The research within STEL has been funded by the School of Computer Science, by the Open University research initiative and by two externally funded projects: Intelligent Feedback (SURF), and NKBW2 (SURF). People that have been involved are:

- Prof. dr. Johan Jeuring (research leader, funded by SURF: intelligent feedback and the school, 2003 - now).
- Dr. Bastiaan Heeren (funded by SURF: intelligent feedback and NKBW2, and by the school, 2007 - now).
- Drs. Arthur van Leeuwen (funded by SURF: intelligent feedback, 2007).
- Drs. Josje Lodder (funded by SURF: intelligent feedback and the school, 2006 - now).
- Ir. Harrie Passier (funded by the school, 2003 - 2008).
- Ir. Sylvia Stuurman (funded by the school, 2003 - 2008).

- Ir. Alex Gerdes (funded by the OU, 2007 - 2011).

Per May 1, 2009 the European eContentPlus project Math-Bridge has started, in which we are a partner. Until February 1, 2012 we have to invest almost 4 person-years in this project. Our main goals within this project are to further develop our domain reasoners, and to develop an on-line mathematics bridging course.

We have twice arranged a course on e-learning in the capita selecta series of courses in the computer science master programme.

2.4 Research method

The research method we apply in our research is to take advanced software technology to improve the state-of-the-art in tools supporting learning and teaching. Often we have to further develop the software technology to realise the improvements.

For example, we use strategies as developed in the compiler construction, rewriting, and proof construction fields to represent procedures used in solving exercises. This makes it easier to develop worked-out examples, to provide hints, to estimate the number of steps needed to solve an exercise, to exactly describe the knowledge represented within a procedure, etc. We use parsing technology to recognize steps made by a student. Our approach is new in the learning technology world, and has led to much better diagnoses than other tools can provide. As a consequence, our feedback services are now used by the Freudenthal Institute applets, the MathDox player (TU/e), and ActiveMath.

To support the easy construction of domain reasoners from strategies and rules we had to develop new generic rewriting theory [15]. This was a good example of how the development of learning tools stimulates research within software technology.

We perform our research in collaboration with several external partners:

- the Information and Computing Sciences department of Utrecht University, at which Jeuring also has an appointment.
- the Freudenthal Institute of Utrecht University, in particular Peter Boon en Paul Drijvers. We deliver domain reasoners for some of the applets in the DWO (Digitale Wiskunde Omgeving).
- the DFKI (German Research Centre for Artificial Intelligence), Saarbrücken, Germany, in particular Erica Melis, Paul Libbrecht, en George Goguaдзе. We deliver domain reasoners for the ActiveMath environment.
- the TU/e, in particular Hans Cuypers and Arjeh Cohen. We deliver domain reasoners for MathDox. Jeuring is member of the steering committee of a Surf project led by Arjeh Cohen.
- the TUD, in particular Erik Jansen. We developed our strategy language in cooperation with the TUD.

2.5 Research plans 2010 - 2015

In the coming years we want to both broaden and deepen our research on software technology for teaching and learning. We have short-term research plans for the NKBW2 and Math-Bridge projects. Here we want to:

- further develop our domain reasoners to support many mathematical domains.
- investigate how the several domains, rules and strategies should be separated in order to make maintaining and adapting exercise assistants as easy as possible.
- develop a framework in which teachers can easily add new exercises, (buggy) rules, strategies, and even domains.
- rewrite the exercise assistant as a generic program to simplify maintenance and adding domains.
- analyse the behaviour of students. We can use the information obtained to investigate how much a student knows, recognize buggy rules, and report the behaviour of a group of students to a teacher.

These two projects are of a more applied nature. Besides these projects we want to study more fundamental problems within our domain, such as:

- Developing intelligent tutoring tools for learning how to program. These tools are also based on strategies, but instead of rewriting, the basic components in the strategies are refinement rules. Intelligent programming tutors will challenge our parsing technology because of the possibly large sets of correct answers.
- Studying the fundamental properties of strategies. Can we describe fundamental properties or categories of strategies, similar to how properties for and categories of languages are formulated? What are the languages represented by strategies?
- Using error diagnosis for determining how much a student knows. This information can be used to update a student model, to test a student's knowledge, or to select the next task for a student. Our diagnoses are much more precise than most diagnoses offered by existing testing tools and tools that maintain student models. Using this more detailed information, we think we can implement tests that give more precise results, improve student models, and provide better task selection.

Strategies play a fundamental role in knowledge and learning. For example, the purpose of serious games is often to practice one or more strategies. And theorem provers use tactics (a kind of strategies) to prove theorems. There are many relations between our work and the use of strategies in these fields. Our long-term research plan is to investigate the role strategies play in such fields, and how we can use or adapt our technologies to improve learning in these fields.

We expect that the tools we develop will be useful for teaching purposes too: for example, our intelligent programming tutor can be partly used for the automatic correction of programs too, and a plagiarism checker is within reach as well. The precise possibilities and problems have to be investigated.

Our domain reasoners are used in multiple exercise assistants, using various data formats, and offering different services. We continuously investigate and add new domains. Some domains are built on top of multiple other domains. This situation leads to several software engineering problems, such as:

- How can we ensure the correctness and consistency of our code base?
- How do we deal with backwards compatibility, scalability, stability aspects.
- How do we deal with the various formats for services?
- How can we support datamining of our log-files to analyse student behaviour?
- How can we use our services in other ways than via external exercise assistants, for example, via a web-application?

Although these software engineering problems are not our focus in our research, we will address them when necessary.

Of course not all of the plans described above can be realised in the little time we have available besides the time for the funded research projects. We intend to apply for money from the EU, STW, NWO, and maybe other funding agencies to fund our work on some of the research problems described above.

3 Software Technology for Quality Improvement

In this research line, started in 2010, the focus is on the assessment and improvement of the quality of software products. Many kinds of properties are studied: functional properties (i.e. does the program have the right input-output correspondence) and non-functional properties (e.g. memory consumption, absence of deadlock, and sustainability).

This line does not have education and learning as initial, main focus. However, applications of the work for education will be actively investigated. Not only via incorporating the results in student research and in courses for students but also in Ph.D. schools and in publishing educational material and new methods for teaching the latest research results effectively.

3.1 Software Analysis

Prof. Marko van Eekelen performs research on verification of both functional and non-functional properties. Functional properties concern properties that define, considering a given input, which kinds of output are correct. Non-functional properties concern properties about other aspects of the program. Examples of that are resource consumption analysis (e.g. size analysis, heap analysis and loop bound inference) and security analyses (e.g. for smart meters and smart grids). This research is mainly performed in cooperation with several researchers at the Radboud University Nijmegen. In this research formal methods are used, e.g. type systems, model based testing, model checking and theorem proving. Van Eekelen is leading several projects. Since 2009, he is 0.3 affiliated with Open University and 0.7 with Radboud University Nijmegen. The projects on Software Analysis listed below are

all part of the Radboud University Nijmegen research. Students at the Open University can also perform research on subjects that are part of the Nijmegen funded research projects. It is the intention that new Open University research lines will be complimentary to the Radboud university research.

AHA - Amortised Heap Space Analysis The goal of this NWO project is to analyse heap space consumption for functional programs and for object-oriented programs [24]. Several papers have been written on size analysis (among others [18, 22, 21]) which forms an important pre-requisite for other forms of resource analysis. The project will end in 2011.

CHARTER - Critical and High Assurance Requirements Transformed through Engineering Rigour

This European Artemis Project (total cost 5M Euro, 2009-2012) aims to ease, accelerate, and cost-reduce the certification of critical embedded systems by melding real-time Java Model Driven Development, rule-based compilation and formal verification. Industrial partners are companies in the areas of Health-care, Surveillance, Automotive and Avionics where software is safety critical and hence correctness is of utmost importance. The Open Group (U.K.), Aicas (Germany), Artisan (U.K.), NLR (NL), Impronova (Sweden), Chalmers (Sweden), University of Twente (NL), Radboud University Nijmegen (NL), Lero (Ireland), Luminis (NL) and QRTech (Sweden). Prof. Van Eekelen is leading the work package on verification. His research contribution is in developing memory consumption analysis techniques for Real-Time Java [20].

Secure Metering The Sentinels STW project Secure Metering (2009-2012), led by Prof. van Eekelen, aims to develop secure, privacy friendly techniques for metering [4], both in the context of smart meter and in the context of road pricing. The project is supported by the industrial net operator Alliander and the government institution RDW (Rijksdienst voor het Wegverkeer).

GoGreen The GoGreen IOP GenCom Project (2011-2015) aims to study and develop methodologies and techniques to enable energy neutral 'green' houses using a secure and privacy friendly energy management system. The project is supported by IOP GenCom Agentschap NL. Partners are Universiteit Twente, Novay, Saxion Hogeschool, Technische Universiteit Delft, PTC and Ambient Systems. Leader of the project is Universiteit Twente.

LaQuSo - Formal Verification In this project formal verification research results are applied in real life industrial case studies [25, 3, 14]. This is supported by the Laboratory for Quality Software (LaQuSo), a joint initiative of Technical University Eindhoven and Radboud University Nijmegen. Prof. van Eekelen is also scientific director for the Nijmegen section of LaQuSo.

3.2 Formal Verification of Networks-on-chip

Julien Schmaltz is full-time employed by the Open Universiteit since 2009. He has several research projects in the area of verification and validation.

FVDAM Project - Networks-on-chip verification This research is the topic of the thesis work

of Freek Verbeek. The work is done in cooperation with Prof. van Eekelen and Prof. Vaandrager. This project is about the application of formal methods to Networks-on-Chips (NoCs) [19]. Our focus of interest is the use of interactive theorem proving - the ACL2 system - to prove global properties of high-level descriptions of NoCs. This project is focusing on deadlocks and related properties. We are (1) developing necessary and sufficient conditions that guarantee a deadlock-free network, (2) developing efficient algorithms to automatically check that a network satisfies such conditions, and (3) integrating these results in the GeNoC environment. We have made progress in all these points. We have formalized a condition for deterministic routing in wormhole and store-and-forward networks [31]. This constitutes a formalization of the condition originally proposed in a seminal paper by Dally and Seitz. We have extended this condition to adaptive routing in store-and-forward networks[30]. This effort is basically the formalization of Duato's theory. We designed an algorithm checking this condition in linear time. We have integrated part of these results in the GeNoC environment to enable the proof that all messages injected in a network eventually evacuate at their expected destination [29, 28]. We have formalized a condition and an associated algorithm for adaptive routing in wormhole networks [27]. We consider our verified algorithms as specifications for more efficient implementations [23]. Our plan is to develop efficient implementations of our algorithms and apply them to realistic case-studies. Our objective is to obtain a theory for deadlock-free networks and associated efficient algorithms. In the future of the project, we will integrate this theory and these algorithms in the GeNoC verification environment. Our objective will be to extend GeNoC with continuous injections of messages and bounds on the time needed to evacuate networks.

QUASIMODO Project - Quantitative analysis of embedded systems This European project (STREP FP7) ended in December 2010. This is the continuation of activities started before joining the Open University of the Netherlands. During this time, we started theoretical and practical research in the domain of testing real-timed systems. In 2010 we worked on the following points:

- Theory for testing real-timed systems. We developed a theory for model-based testing and real-timed systems.
- Theory for testing symbolic timed automata. We developed the model of Symbolic Timed Automata and an associated testing relation. This was the M.Sc. thesis work of S. von Styp-Rekowski and was performed in collaboration with H. Bohnenkamp and J.-P. Katoen from RWTH Aachen. The work is presented in [32].
- Applications of real-timed model-based testing. Model-based testing is a recent technology that is gaining popularity. Its application to real-timed systems is a current hot topic. We are applying this technique to two case-studies. The first one consists in a Wireless Sensor Network developed by the CHESSE company. The second one is the DNSSEC protocol. From these applications, we may derive interesting assignments for the course "Software Verification and Validation" at the Open University.

This work involves also several researchers and students at Nijmegen.

3.3 Research Method

As much as possible formal methods are used. New areas are generally first studied in a fundamental context and then the results are applied to real life industrial problems. In this way, fundamental results are achieved that have an impact in society. New results are also incorporated in education, e.g. via Master thesis work of Open University students and in the OU course Software Verification and Validation.

Prof. van Eekelen and Dr. Schmalz have their office at the Digital Security Section of the Institute for Computing and Information Sciences (iCIS) of the Radboud University. In this way their OU research can be performed in the context of the excellent quality iCIS research environment. The research of iCIS ranked first in the recent Dutch national computer science research assessment. In that assessment the Digital Security section got the highest possible marks on all aspects.

3.4 History

This research line just started in 2010.

3.5 Research Plans 2010-2015

Many activities have recently started or are about to start. During the period 2010-2015 it will become clear which research activities grow into proper research lines. Below we list the current plans.

Apart from finishing/continuing the projects that are described above, plans are being developed for creating an open university line of research which can be applied both in industry and in teaching. The main topic of the research will be on *Analysis and Verification of Non-Functional Properties*.

- **Analysis of Sustainability** Prof. van Eekelen is starting research on sustainability as joint work with Anda Counotte-Potman, Harald Vranken and Julien Schmalz. Part of this work may be the development of a sustainability maturity model for knowledge intensive organisations and research towards analysis (and reduction) of energy consumption of software. Two other options are to study energy benefits of virtualisation [33, 7, 12] and to study the energy consumption of software.
- **Verification of Network on Chip Properties** In the beginning of 2011 Dr. Schmalz received a prestigious 3-year grant from Intel to intensify the research on Networks on Chip in order to develop a formal verification environment to prove deadlock freedom, livelock freedom and memory consistency properties using the GeNoC methodology.
- **Verification of Software Product Lines** Furthermore, the formal verification research we would like to extend from Networks on Chip to Software Product Lines. In our design of GeNoC we studied the properties that all networks should satisfy. Our network model is build as the composition of different components. From properties of the components we derived properties of the networks. In this new project, we would

like to research if we can develop a similar framework for another application domain. Our intended domain is Software Product Lines (SPLs). SPLs define a set of assets from which products are derived. This derivation is regulated by interconnection rules, patterns and variation points. The point is that an SPL proposes a regulated framework to build new systems. They are common points to all products. There must be properties that all the products from an SPL should satisfy. The goal of this project would be to define a formal model of an SPL by identifying common constituents, global properties satisfied by a family of products, and sufficient proof obligations on the constituents.

- **Analysis of Security Properties** It is the intention to start a research line on security with Harald Vranken (with as possible focus points virtualisation and cybercrime; the latter in cooperation with Professor Wouter Stol from the faculty of Management Sciences of the Open Universiteit).
- **Analysis of Usage of Software** First thoughts are to start studying the usage of software in the following way. From within the software data are collected. These data are interpreted, analysed and visualised. The results of the analysis can give an indication about the functional aspects of the software usage but it may also be that they give an indication about non-functional aspects such as security issues or memory/energy consumption. Based upon these results, advice can be given to the user (who can be in a teaching context or in an industrial context) to improve the usage of the software or to the designer/developer to improve the software itself such that its usage can be improved. In this context Sylvia Stuurman and Marko van Eekelen are planning to start research to monitor the usage of StOER/OpenU, an experimental environment that is meant to act as a portal for Open University Computer Science students providing information and supporting the communication between students and teachers.
- **Computer Science Education Methodology** There is the possibility of starting a research line in Computer Science Educational Methodology. First results in the latter have been achieved by Bastiaan Heeren and Harrie Passier in developing (and evaluating) a new method to teach XML design in a more formal way (using regular expressions).

In order to promote the Open Universiteit research activities it is the intention to organize scientific events on a regular basis. This has resulted in two events in 2011:

- In 2011 Marko van Eekelen is co-chair of Computer Science Educational Research (CSERC2011¹, held in Heerlen, April 2011) together Peter Sloep (OU) and Gerrit van der Veer (OU).
- Julien Schmaltz and Marko van Eekelen are co-chairs for Interactive Theorem Proving 2011 (ITP2011², to be held in Nijmegen, August 2011) together with Herman Geuvers (RU) and Freek Wiedijk (RU).

¹<http://www.ou.nl/cserc/>

²<http://itp.cs.ru.nl/>

References

- [1] John R. Anderson. *Rules of the Mind*. Lawrence Erlbaum Associates, 1993.
- [2] National Research Council. *How People Learn – Brain, Mind, Experience, and School*. National Academy Press, 2000.
- [3] Schierboom Erik, Alejandro Tamalet, Hendrik Tews, Marko van Eekelen, and Sjaak Smetsers. Preemption abstraction: A lightweight approach to modelling concurrency. In *Formal Methods for Industrial Critical Systems (FMICS 2009)*, volume 5825 of *LNCS*, pages 149–164. Springer, 2009.
- [4] Flavio D. Garcia and Bart Jacobs. Privacy-friendly energy-metering via homomorphic encryption. In J. Cuellar et al., editor, *In 6th Workshop on Security and Trust Management (STM 2010)*, Lecture Notes in Computer Science. Springer Verlag, 2010.
- [5] Alex Gerdes, Bastiaan Heeren, and Johan Jeuring. Constructing strategies for programming. In José Cordeiro, Boris Shishkov, Alexander Verbraeck, and Markus Helfert, editors, *Proceedings 1st International Conference on Computer Supported Education, CSEDU '09*, 2009.
- [6] Alex Gerdes, Bastiaan Heeren, Johan Jeuring, and Sylvia Stuurman. Feedback services for exercise assistants. In Dan Remenyi, editor, *Proceedings of the 7th European Conference on e-Learning*, pages 402–410. Academic Publishing Limited, 2008.
- [7] J. Haag, T. Horsmann, S. Karsch, and H. Vranken. A Distributed Virtual Computer Security Lab with Central Authority. In *Intrnational Computer Science Education Research Conference (CSERC2011)*. ACM Digital Library, 2011. Heerlen, The Netherlands 7 - 8 April, 2011.
- [8] Bastiaan Heeren and Johan Jeuring. Recognizing strategies. In Aart Middeldorp, editor, *WRS 2008: Reduction Strategies in Rewriting and Programming, 8th International Workshop*, 2008.
- [9] Bastiaan Heeren and Johan Jeuring. Canonical forms in interactive exercise assistants. In Claudio Sacerdoti Cohen and Stephen Watt, editors, *Proceedings of Mathematical Knowledge management, MKM 2009*, Lecture Notes in Computer Science. Springer, 2009. Also available as Technical Report, ICS Utrecht University, UU-CS-2009-0??
- [10] Bastiaan Heeren, Johan Jeuring, Arthur van Leeuwen, and Alex Gerdes. Specifying strategies for exercises. In Serge Autexier, John Campbell, Julio Rubio, Sorge Volker, Masakazu Suzuki, and Freek Wiedijk, editors, *Intelligent Computer Mathematics*, volume 5144 of *Lecture Notes in Computer Science*. Springer, 2008. Also available as Technical Report, ICS Utrecht University, UU-CS-2008-001.
- [11] Johan Jeuring, Harrie Passier, and Sylvia Stuurman. A generic framework for developing exercise assistants. In *Proceedings of the 8th International Conference on Information Technology Based Higher Education and Training, ITHET 2007, Kumamoto City, Japan, July*, 2007. Also available as Technical Report, ICS Utrecht University, UU-CS-2007-017.
- [12] J. Kamer and H. Vranken. The impact of server virtualization on ITIL processes. In *In-*

- ternational Conference on Cloud Computing and Services Science (CLOSER2011)*, 2011. Noordwijkerhout, The Netherlands, 6 - 9 Ma, 2011, publisher = ACM Digital Library.
- [13] Josje Lodder, Johan Jeuring, and Harrie Passier. An interactive tool for manipulating logical formulae. In M. Manzano, B. Pérez Lancho, and Gil A., editors, *Proceedings of the Second International Congress on Tools for Teaching Logic*, 2006.
 - [14] Ken Madlener, Sjaak Smetsers, and Marko C. J. D. van Eekelen. A formal verification study on the rotterdam storm surge barrier. In Jin Song Dong and Huibiao Zhu, editors, *ICFEM*, volume 6447 of *Lecture Notes in Computer Science*, pages 287–302. Springer, 2010.
 - [15] Thomas van Noort, Alexey Rodriguez Yakushev, Stefan Holdermans, Johan Jeuring, and Bastiaan Heeren. A lightweight approach to datatype-generic rewriting. In *WGP '08: Proceedings of the ACM SIGPLAN workshop on Generic programming*, pages 13–24. ACM, 2008.
 - [16] Harrie Passier and Johan Jeuring. Feedback in an interactive equation solver. In M. Seppälä, S. Xambo, and O. Caprotti, editors, *Proceedings of the Web Advanced Learning Conference and Exhibition, WebALT 2006*, pages 53–68. Oy WebALT Inc., 2006.
 - [17] Phil Race. *Making Learning Happen – A Guide for Post-Compulsory Education*. Sage Publications, 2005.
 - [18] Simona Ronchi Della Rocca, editor. *Typed Lambda Calculi and Applications, 8th International Conference, TLCA 2007, Paris, France, June 26-28, 2007, Proceedings*, volume 4583 of *Lecture Notes in Computer Science*. Springer, 2007.
 - [19] J. Schmaltz, F. Verbeek, and T. van den Broek. Formal validation and verification of networks-on-chips: Status and perspective. In *Designing Correct Circuits (DCC'10)*, pages 1–15, Paphos, Cyprus, March 20–21 2010. part of ETAPS 2011.
 - [20] Olha Shkaravska, Rody Kersten, and Marko Van Eekelen. Test-based inference of polynomial loop-bound functions. In Andreas Krall and Hanspeter Mössenböck, editors, *PPPJ'10: Proceedings of the 8th International Conference on the Principles and Practice of Programming in Java*, ACM Digital Proceedings Series, pages 99–108, 2010.
 - [21] Olha Shkaravska, Marko C. J. D. van Eekelen, and Ron van Kesteren. Polynomial size analysis of first-order shapely functions. *Logical Methods in Computer Science*, 5, issue 2, paper 10:1–35, 2009. Special Issue with Selected Papers from TLCA 2007.
 - [22] Alejandro Tamalet, Olha Shkaravska, and Marko van Eekelen. Size analysis of algebraic data types. In Peter Achten, Pieter Koopman, and Marco T. Morazán, editors, *Trends in Functional Programming*, volume 9 of *Trends in Functional Programming*, pages 33–48. Intellect, 2009. ISBN 978-1-84150-277-9.
 - [23] T. van den Broek and J. Schmaltz. A generic implementation model for the verification of networks-on-chips. In S. Ray and D. Russinoff, editors, *8th Intl. Workshop on the ACL2 Theorem Prover and Its Application*, pages 130–134, Northeastern Univ., Boston MA, USA, 2009. ACM.
 - [24] M. van Eekelen, O. Shkaravska, R. van Kesteren, B. Jacobs, E. Poll, and S. Smetsers. AHA: Amortized Heap Space Usage Analysis. In Marco Morazán, editor, *Trends in*

Functional Programming Volume 8: Selected Papers of the 8th International Symposium on Trends in Functional Programming (TFP07), New York, USA, pages 36–53. Intellect Publishers, UK, 2008.

- [25] Bernard van Gastel, Leonard Lensink, Sjaak Smetsers, and Marko van Eekelen. Deadlock and starvation free reentrant readers-writers—a case study combining model checking with theorem proving. *Science of Computer Programming*, 76(2):82–99, 2011. Selected papers from the workshops on Formal Methods for Industrial Critical Systems - FMICS 2007 + FMICS 2008.
- [26] Kurt VanLehn. *Mind Bugs – The Origins of Procedural Misconceptions*. MIT Press, 1990.
- [27] F. Verbeek and J. Schmaltz. Automatic verification for deadlock in adaptive routing wormhole networks. In *5th International Symposium on Networks-on-Chip (NOCS'11)*, Pittsburgh, PA, USA, May 1–4. IEEE. To Appear.
- [28] F. Verbeek and J. Schmaltz. A formal proof of a necessary and sufficient condition for deadlock-free adaptive networks. In M. Kaufmann and L. Paulson, editors, *Proc. of 1st International Conference on Interactive Theorem Proving (ITP'10)*, number 6172 in LNCS, pages 67–82, Edinburgh, July 11–14 2010. Springer. Part of FloC 2010.
- [29] F. Verbeek and J. Schmaltz. Formal validation of networks-on-chips: Deadlock and evacuation. In *Proc. of International Conference on Design, Automation and Test Europe (DATE'10)*, pages 1701–1706, Dresden, Germany, March 8–12 2010. ACM/IEEE.
- [30] F. Verbeek and J. Schmaltz. A comment on “a necessary and sufficient condition for deadlock-free adaptive routing in wormhole networks”. *IEEE Transactions on Parallel and Distributed Systems*, 2011. (PrePrints) available on-line at IEEE TPDS.
- [31] F. Verbeek and J. Schmaltz. Proof pearl: A formal proof of dally and seitz necessary and sufficient condition for deadlock-free routing in interconnection networks. *Journal of Automated Reasoning*, 2011. To Appear. Available on-line at Springer.
- [32] S. von Styp, H. C. Bohnenkamp, and J. Schmaltz. A conformance testing relation for symbolic timed automata. In *8th International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS'10)*, volume 6246 of LNCS, pages 243–255, Vienna, Austria, September 8–10 2010.
- [33] H. Vranken, J. Haag, Horsmann T., and S. Karsch. A Distributed Virtual Computer Security Lab. In *International Conference on Computer Supported Education (CSEDU2011)*. SciTePress Digital Library, 2011. Noordwijkerhout, The Netherlands, 6 - 9 May, 2011.