Institut für Informatik

Ludwig Maximilians Universität München LMU

Prof. Dr. Rolf Hennicker · Institute of Informatics
LMU Munich, · Oettingenstraße 67 · 80538 München

Professor Jan Trlifaj
Vice-Dean
Faculty of Mathematics and Physics
Charles University
M266, Ke Karlovu 3

CZ - 16 Praha 2

Prof. Dr. Rolf Hennicker
Software and Computational Systems Lab
Institute of Informatics
LMU Munich

Oettingenstr. 67, 80538 München
Germany
Tel.: 089/2180-9184
Fax: 089/2180-9175

hennicker@ifi.lmu.de

Munich, May 29, 2019

Report on the Habilitation Thesis

"Verification of Software"

submitted by

Jan Kofron Ph.D.
Charles University Prague

For the production of reliable software, which is particularly important for safety critical systems, it is indispensable to apply best practices of software engineering in all phases of the software lifecycle. This subsumes specification techniques to formulate desired properties of a system as well as methods to check properties of system designs and implementations. A crucial role for these tasks is played by formal specification and verification techniques which set the frame for the habilitation thesis of Jan Kofron.

The thesis starts with a short introduction to the field (chapter 1) and an overview on motivations and contributions of the scientific work of Jan Kofron in the areas of software behaviour specification (chapter 2) and code verification (chapter 3). The major scientific results Jan Kofron has achieved since 2005 are presented in the subsequent chapters by a collection of nine selected research papers. These are published in relevant international journals (Journal of Computer and Information Science, Information and Software Technology, Formal Aspects of Computing (two times)) and in the proceedings of peer-reviewed international software engineering conferences (among them ACM Symposium on Applied Computing, SEFM, FASE, and the main European Conference on Object-Oriented Programming ECOOP). Thematically the thesis consists of two major parts dealing with software behaviour specifications (chapters 4-7) and code verification (chapters 8-12).

In the area of software architectures and behaviour specifications Jan Kofron follows the ideas of the software component model developed in the software engineering group at Charles

University which relies on (hierarchical) components and behaviour protocols. Crucial aspects to produce correct component systems concern behavioural compatibility of connected components ("horizontal" dimension) and compliance of hierarchically structured components with an overall behaviour specification ("vertical" dimension). In each case, compliance checking may suffer under the problem of state explosion. To tackle this issue chapter 4 proposes and implements a novel space representation, relying on so-called parse tree automata, which is more efficient for behaviour protocols of "practical size". Another approach to the same problem is pursued in chapter 5 published by Jan Kofron as a single author. It proposes a translation of behaviour protocols to the input language Promela of the established model checker Spin. An evaluation of the new approach shows that compliance checking is now still "an order of magnitude faster" than before.

Significant conceptual work on component behaviour specifications is presented in chapters 6 and 7. Chapter 6 uses Extended Behavior Protocols (EBPs) which allow to include component state variables in specifications. It is shown how these can be utilised to model variability of software architectures and thus product lines. A short mentioning of the contents of chapter 6 in the overview chapter 2 would have been welcome. Chapter 7 deals with the problem that specification formats may be too far away from the notations used in the daily life of programmers. Therefore a comprehensive framework, called Threaded Behavior Protocols (TBPs), is investigated being syntactically more close to constructs of real imperative programming languages but still expressive enough for specifying behaviours of multi-threaded software components. The TBP framework is underpinned with a formal semantics which provides a solid basis for notions of compliance and refinement.

The second part of the habilitation thesis is devoted to verification on the code level where Jan Kofron has provided significant contributions in the areas of explicit model checking, static analysis of programs and symbolic verification. The size of needed space and time to perform a verification task is a crucial concern again. Chapter 8 proposes, implements and evaluates two analysis techniques for finding dead variables, i.e. variables which do not influence the further program execution, in multi-threaded Java programs. It is shown by a benchmark analysis that both techniques, though related to different parameters, improve significantly the performance of on-the-fly explicit model checking.

Chapters 9 and 10 deal with static analysis of programs for web development written in dynamic languages like PHP and JavaScript. A static analysis framework is presented to detect vulnerabilities of web applications. The framework is carefully related to the literature and experiments are performed to compare it with other state of the art analysis tools. The results of the comparison show that the framework found more real problems while at the same time less false positives are reported.

The last two technical chapters 11 and 12 consider symbolic methods for code verification. The idea of symbolic methods is to represent concisely very large (possibly infinite) sets of states. Thereby a particular role is played by Craig interpolants known from mathematical logic and used for abstractions of programs. Chapter 11 shows that Craig interpolants may not be sufficient. Therefore a generalisation of Craig interpolants is proposed, called Partial Variable Assignment Interpolants (PVAIs). It is shown how PVAIs can be computed and their strength is analysed. A tool that implements the construction of PVAIs is described in chapter 12 together with an evaluation showing the performance advantages of the interpolant computation.

The thesis ends with a short summary and with an interesting discussion of a still missing brick concerning the transition from specification level to real programs.

Evaluation: The habilitation thesis of Jan Kofron shows his broad knowledge and his creativity for finding solutions in the field of software verification. This covers verification on the specification level (chapters 4 and 5) as well as a big variety of significant results for different methods of code verification (chapters 8-12). He is also an expert in developing tool support. Moreover, there is considerable amount on conceptual work on component-based systems shown in chapter 6 and 7 which is relevant and of high interest for the international community in component-based software engineering. Jan Kofron is without doubts an excellent scientist and I believe that his present and future work will have a significant impact on new and sustainable developments. His international visibility is underpinned by several workshops and conferences for which he has served as PC member or PC co-chair over the last years. In summary, I recommend strongly the acceptance of the habilitation thesis and the promotion of Jan Kofron to become an Associate Professor.

(Prof. Dr. Rolf Hennicker)