Proceedings of the Estonian Academy of Sciences,
2013, **62**, 1, 1–2
doi: 10.3176/proc.2013.1.01
Available online at www.eap.ee/proceedings

EDITORIAL

Preface

The series of symposia under the title "Programming Languages and Software Tools (SPLST)" started in Szeged, Hungary, in 1989. It was originally intended to provide an opportunity to computer scientists from Hungary, Finland, and Estonia to meet regularly and introduce their current work to each other. Informally the conferences were even called Finno-Ugric symposia on languages and tools and it was hoped that these events might help the computer scientists from two Eastern European countries to get integrated via Finland with the more developed Europe. Now, when the three states have joined the EU and the words "Finno-Ugric" have been removed from the title, we are looking at an all-European series of conferences organized biannually in these countries. Another tradition of these symposia is that the majority of participants are young scientists or PhD students who enjoy the opportunity to meet their colleagues of the same age, discuss scientific problems, listen to invited talks of experienced researchers, and receive rather detailed reviews as feedback to their submitted papers.

The topics of SPLST have reflected actual trends in the research on software development and undergone several changes during its history. When it started, computing was still mainly an academic activity and software construction was based mostly on compiling techniques. Shortly after that software development transformed into an industrial activity that brought to the forefront software production lines, modelling of the software process, software architecture, software reuse, testing, and so on. Another revolution in computing was induced by spreading of the internet and World Wide Web that for IT systems opened opportunities like never before, but also raised new problems like cyber crime and the need to deal with security aspects in computer systems. New cooperative and distance development methods appeared and have evolved up to nowadays cloud computing facilities. Complexity and worldwide linkage has highlighted terms like trustworthy software systems, safety, quality of service (performance, reliability, availability), security, and privacy. Software development now includes directions like aspects, agile techniques, and model-driven software engineering. This forms a new context for traditional topics of SPLST: programming languages, tools, and environments.

This special issue of the *Proceedings of the Estonian Academy of Sciences* contains seven papers that were first presented at the 12th symposium in the series, held in Tallinn on 5–7 October 2011. Those seven papers were selected by the Program Committee from all papers given at the conference to be invited for submission to this issue and were subsequently subjected to the standard refereeing process of the journal.

A traditional SPLST topic – techniques and tools for software engineering – is represented here by the article "DPF Workbench: a multi-level language workbench for MDE", written by Yngve Lamo and his co-authors. The paper provides a fully diagrammatic specification language to develop domain-specific metamodels and their transformations facilitating generation of software from these models. This approach contributes to modern model-driven engineering paradigm as well as visual specification languages that allows domain experts to develop graphical models via drawing schemes at different levels of abstraction. The tool supports automatic validation of these specifications and conformance between modelling levels.

Software architecture matters are touched upon in several papers. Most directly these questions are treated in the paper "Interleaving human and search-based software architecture design" by Sriharsha Vathsavayi et al. Fragments of the structure of the system under development are evolved automatically rather than designed by means of a classical method. Techniques of genetic algorithms used here are inspired by the process of evolution in the nature.

Another architecture-oriented paper "Implementing artificial intelligence: a generic approach with software support" by Teemu J. Heinimäki and Juha-Matti Vanha-

tupa describes an approach for rapid scripting and testing software agents. The authors believe that the method allows implementation of components of Artificial Intelligence systems. They demonstrate the approach by scripting characters that comprise a virtual world in computer games.

The paper on architecture above uses a method that mimicks biological evolution. A similar heuristic approach is used in the paper by Maragarita Spichakova "An approach to the inference of finite state machines based on a gravitationally-inspired search algorithm" for generation of Moore machines, a version of final state machines with output. In this case the inference algorithm she uses is inspired by Newton's gravitational law.

New architectures of computers and distributed systems encourage people to reformulate or reinterpret well-known concepts and knowledge. Antti Valmari asks in his paper: "Does the Shannon bound really apply to all data structures?" Roughly speaking, this bound determines a minimum amount of memory needed to store a given information. Originally this theorem was given for the memory consisting of a single series of bits. The author shows that it is problematic to apply such an information-theoretic bound to a memory with complex structures. By the way, this article created long and eager discussions between the author and reviewers that resulted in a number of revisions of the paper and, I believe, improved it as well as brought satisfaction and fun to the parties of the debate.

Modern software processes still require much basic research on verification and modelling methods. Model checking is a most popular approach for generating safety-critical software. The paper "Bounded saturation-based CTL model checking" by András Vörös et al. combines different model checking techniques to obtain a new quality in safety. The paper by Gabriella Tóth and co-authors "Adjusting effort estimation using micro-productivity profiles" concentrates on software processes from another viewpoint, namely how to organize a software project and how to measure or estimate the productivity of the team. The formalism used and the experiments commit the intuitive expectation that a series of changes require usually more resources than implementing the same changes in a single step.

The organizers of the symposium and the guest editor of this issue thank all reviewers and the Program Committee for thorough evaluation of the papers. I would also like to thank the Estonian Centre of Excellence in Computer Science, funded mainly by the European Regional Development Fund, for sponsoring SPLST'11 and all people from the host organization, Institute of Cybernetics, who assisted in preparation of this symposium.

Jaan Penjam
Guest editor