

Executive Summary:

Dagstuhl Seminar 10191 on Program Composition and Optimization: Autotuning, Scheduling, Metaprogramming and Beyond, 9-12 May 2010

Christoph W. Kessler¹, Welf Löwe², David Padua³, and Markus Püschel⁴

¹ Linköping University, Linköping, Sweden, chrke@ida.liu.se

² Linnaeus University, Växjö, Sweden, welf.lowe@lnu.se

³ University of Illinois at Urbana Champaign, USA, padua@uiuc.edu

⁴ Carnegie Mellon University, USA, pueschel@ece.cmu.edu

1 Seminar Motivation

Components are a well-proven means of handling software complexity. Reusable components and software composition support the construction of large and reliable software systems from pre-defined and tested partial solutions. When maximizing reusability, we end up with components that are very general and do not fit one particular scenario perfectly. Therefore, adaptation, especially optimization, is established as a technique to deal with such mismatches.

A main motivation for us to initiate this Dagstuhl seminar was our observation that composition and optimization are discussed almost in isolation in two different scientific communities. Composition is understood as a problem in software engineering for general-purpose computing; techniques include meta-/generative programming, configuration and (self-)adaptation, and architecture and component systems. Optimization is typically understood as a problem of high-performance and stream computing, and compiler construction; techniques include auto-tuning, scheduling, parallelization, just-in-time compilation, runtime optimizations, and performance prediction.

The goal of this Dagstuhl seminar was thus to bring together outstanding researchers from these two communities. We focused on a common topic of interest: the *optimized generation and composition of programs from components*, which includes the automatic adaptation to advanced execution platforms. Hence, the topic addresses solutions for both sequential and parallel components and target systems. This issue requires the integration of techniques from both subdomains.

A longer introduction to the main issues in composition and optimization is given in a separate paper.

2 Seminar Organization and Results

As a baseline, the seminar reviewed key results achieved so far in the two communities. In the discussion sessions, we combined expert insight from both groups to identify common issues and point out challenges for advancing the state of the art in composition and optimization.

The seminar started with a round of short self-presentations by all 26 participants. Based on the 17 presentation proposals submitted by the participants in advance, the 2.5 days of the seminar were organized into three main topic areas: *Software composition* (monday), *Autotuning* (tuesday) and *Multicore issues* (wednesday). Each topic area was opened by a survey presentation of 30 minutes, followed by sessions with 30-minutes technical presentations, and concluded with a panel discussion each.

After some initial effort of understanding specific terminology and concepts used and problems discussed by the respective other group, there were lively and productive discussions across the two communities. In particular, we collected, during discussions, a longer list of research challenges and open problems in optimized composition, which we intend to condense into a common paper in the next months.

At the end of the seminar, we received positive feedback from several of the participants indicating that the seminar took up relevant topics, provided sufficient room for discussion in spite of the tight schedule of a short seminar, and inspired concrete plans for cooperations both within and beyond community boundaries.

In summary, we conclude that the seminar was constructive and largely met its goals. Dagstuhl provided a productive and interactive atmosphere that stimulated a cross-fertilization between the two groups. We consider the seminar a first step in a beginning process of getting the two communities closer to each other.