C/C++-Based System Design
Introduction

This part of the book contains the five most interesting contributions selected from the C/C++-Based System Design Thematic Area (CSD-TA) workshop, which was part of the FDL'05 conference. The CSD-TA addresses language-based hardware/software system methodologies and tools for modeling, simulating, evaluating the performance, and analyzing hardware/software systems. The focus of this thematic area is on research approaches applying C/C++-based languages like SystemC, but other languages are explicitly welcomed as well. Topics of interest also include Real-Time Operating System (RTOS) and embedded software aspects.

Through its open and extensible language architecture, SystemC provides an efficient framework for new research approaches addressing the design of complex electronic systems. On this foundation, Chapters 3–7 present methodologies and extensions for SystemC to enlarge its scope and expressiveness. Chapters 3–7 address very different design challenges such as behavioral separation in protocol-dominated systems, checking of timing properties, hardware/software integration, and mixing synchronous with untimed models of computation (MoC).

In the first chapter of this part, “Behaviour separation: a high-level methodology applicable in the SystemC environment” (Chapter 3), Giovanni et al. describe an approach to foster reuse when modeling design components communicating through protocols. Their technique enables an efficient reuse of parts of components sharing one protocol by separating the fixed protocol behavior from the device-specific behavior. The chapter illustrates the applicability and efficiency of the presented methodology using an Advanced Microcontroller Bus Architecture (AMBA) bus system master device design example.

In “Mixing synchronous reactive and untimed MoCs in SystemC” (Chapter 4), Fernando Herrara and Eugenio Villar present a methodology that allows to mix models of computations for describing components in a heterogeneous system. In particular, the work approaches the integration of the synchronous, reactive, and the untimed model of computation in the SystemC language framework. The chapter describes a concrete SystemC-based
mechanism applying so-called border channels and border processes to inter-
face synchronous, reactive, and untimed models.

The third chapter by André et al., “Interface-centric abstraction level for rapid hardware/software integration” (Chapter 5), describes a portable set of abstract operative system (OS) primitives realized as an application program-
ing interface (API) facilitating hardware/software codesign in SystemC. The API covers all aspects of typical operation systems, e.g., process management, communication, synchronization, and timing. For each OS primitive, they pro-
vide a hardware and a software implementation alternative. With these alterna-
tives, the designer can easily evaluate different alternative system partitionings.

Roland et al. received the FDL’05 Best Paper Award for their contribution,
“Efficient and customizable integration of temporal properties into SystemC” (Chapter 6). The work presents an extension of SystemC with cus-
tomizable temporal properties. The two main aspects discussed in this chapter are the way to specify temporal properties in SystemC models and the mechan-
ism to check these properties automatically. Both aspects are illustrated by modeling examples and performance-evaluation results.

The heterogeneity and complexity of electronic systems necessitates integration of different MoC into a semantically well-defined and efficient modeling environment. In “UMoC++: A C++-based multi-MoC modeling environment” (Chapter 7), Deepak et al. describe basic techniques to inte-
grate generic MoCs taken from functional frameworks, such as ForSyDe and SML-Sys, into an efficient framework based on an imperative language. By doing so, the presented environment called UMoC++ promises to improve the efficiency while maintaining the well-defined semantic of the heterogeneous system model.

It is my hope that this short introduction will draw your attention to the very interesting work presented in the following chapters. As they outline the scope of the CSD-TA quite well, they might also encourage you to attend or even contribute to one of the following CSD workshops at the FDL conference.

Frank Oppenheimer
OFFIS e. V.
R&D division Embedded Hardware/Software-Systems
Oldenburg, Germany, February 2006