Chapter 2
Co-Modeling of Embedded Networks Using SystemC and SDL

Sergey Balandin
Nokia Research Center, Finland

Michel Gillet
Nokia Research Center, Finland

Irina Lavrovskaya
St. Petersburg State University of Aerospace Instrumentation, Russia

Valentin Olenev
St. Petersburg State University of Aerospace Instrumentation, Russia

Alexey Rabin
St. Petersburg State University of Aerospace Instrumentation, Russia

Alexander Stepanov
St. Petersburg State University of Aerospace Instrumentation, Russia

ABSTRACT

Today, SDL and SystemC are two very popular languages for embedded systems modeling. SDL has specific advanced features that make it good for reflection of the multi-object systems and interactions between modules. It is also good for system model validation. The SystemC models are better suitable for tracing internal functions of the modeled modules. The hypothetical possibility of combined use of these two languages promises a number of benefits for researchers. This article specifically addresses and discusses the integration of SDL and SystemC modeling environments, exchange the data and control information between the SDL and SystemC sub-modules and the real-time co-modeling aspects of the integrated SDL/SystemC system. As a result, the mechanisms of SDL/SystemC co-modeling is presented and illustrated for an embedded network protocols co-modeling case study. The article gives an overview and description of a co-modeling solution for embedded networks protocols simulation based on experience and previous publications and research.

DOI: 10.4018/978-1-4666-2776-5.ch002
INTRODUCTION

The embedded systems design is a labor-consuming and complicated process. Modeling is used to simplify it and speed it up. This article gives an overview of a modeling application in the general embedded systems design flow and shows scenarios, in which modeling is especially efficient to be adopted into the development process. We selected two widely used modeling languages SDL and SystemC and started with an overview and comparison of their key features (Gipper, 2007; Fonseca, 2008). The comparison shows that both languages have a number of disadvantages that could be compensated by the joint use of SystemC/SDL. Thus, we propose three approaches for the SystemC and SDL co-modeling and compare them. Analysis of all factors shows that one of the approaches is better for the co-modeling application and we propose two ways for implementation of the approach. In the end, we give an example practical application for the introduced way of co-modeling.

EMBEDDED SYSTEMS DESIGN FLOW

An embedded system is a combination of computer hardware and software (could have fixed capability or be programmable) that is specifically designed for a particular function. This makes it a system dedicated for an application(s), its specific parts or parts of a larger system. An embedded system is built to control a function or range of functions and usually have strict real-time computational restrictions (Heath, 2003; Barr & Massa, 2006; Kamal, 2008).

Embedded systems span all aspects of modern life and there are many examples of their use. They are cash dispenses, payment terminals, handheld computers, mobile phones, telecommunication equipment and so on.

Many different activities are required to carry a complex electronic system from initial idea to the physical implementation. Performance modeling helps to understand and establish the major functional and nonfunctional characteristics of the product. Functional modeling results in a specification of the functional behavior of the product. Design and synthesis refine the product specification into a sequence of design descriptions that contain progressively more design decisions and implementation details. Validation and verification hopefully ensure that the final implementation behaves as specified. All these activities operate on models and not on the real physical object. One obvious reason for using a model is that the real product is not available before the development task is completed (Jantsch, 2004).

But the embedded systems design encounters a number of difficulties caused by increasing complexity of projects, increasing requirements to products reliability, power consumption and demand to speed-up the project design phase. The modern approach to the system design is illustrated by Figure 1 and includes the following stages:

- **Conceptual System Design**: The primary goal of this stage is the formulation of the main system design choice, analysis of the system design and development of the specification draft;
- **Specification**: This stage is targeted to get the final version of the system specification and benefits from having the model done in a high-level language, usually in C/C++, SystemC, SDL;
- **Logical Design**: Converting the executable project specification to the register transfer level, i.e., the result is a specification implementation in Verilog/VHDL and further at the gate level. In this context can be also called – architectural design;