Metamodeling for model-based system design

Metamodeling is defined, according to Wikipedia, as the construction of a collection of concepts within a certain domain. A model is an abstraction of phenomena in the real world; a metamodel is yet another abstraction that highlights the properties of the model itself. Metamodeling originated in the field of mathematical logic, and software engineering research has been very actively using this approach for model-driven software design. It has been increasingly used in electronic system design in recent years as a result of our never-ending need to move the abstraction level of design ever higher. Metamodeling tools and methodologies have recently been employed in creating design frameworks for SoCs. Some experts believe that there will be increasing use of metamodels for supporting automated system integration, system test generation, and verification because these models provide information that enable tools to efficiently explore system design alternatives as well as generate verification test benches.

In the May/June issue, we have a special section on this emerging topic. The special section consists of two overview articles. The first, by Alberto Sangiovanni-Vincentelli and colleagues, shows how the metamodeling concept can be used to compare different models of computation, to provide mathematical machinery for proving design properties, and to support platform-based design. The second article by Roberto Passerone and colleagues reviews the role that metamodels have played and continues to play in several industrial and research projects across Europe. I would like to take this opportunity to thank our guest editors, Sandeep Shukla and Alberto Sangiovanni-Vincentelli, for their excellent efforts in introducing this subject to our readers.

This issue also concludes the theme "IEEE 1500 and Its Usage," with three additional articles on this topic. These three articles, which complement the coverage in the January/February 2009 issue, discuss an automated test synthesis methodology and tool suite to build IEEE-1500-wrapped cores in a production environment; an IEEE 1500 verification framework based on a commercial functional verification suite; and an in-depth comparison of two widely known strategies for testing SoCs in terms of test data volume.

Also included in this issue are two general-interest articles. The article by Ishwar Parulkar and Babu Turumella presents a silicon debug approach for a high-performance server chipset that includes a chip-multithreaded Sparc microprocessor. The other, by Vladimir Živkovic and colleagues, discusses test development for embedded mixed-signal and RF modules, and it uses two SoC examples to illustrate the benefits and performance of their approach.

I hope you enjoy this issue. If you have any feedback, please share it with us.

Tim Cheng
Editor in Chief
IEEE Design & Test