For nearly 25 years, the standardization of design and verification languages has been an accepted part of the electronics industry. Although many early electronic-design-automation (EDA) standards dealt primarily with the interchange between tools (for example, EDIF and GDS II), the majority of standards in use today define how design and verification engineers provide input to EDA tools. Widely used examples include the following: SystemVerilog, Verilog, Verilog-AMS, VHDL, and SystemC for designs and models; SystemVerilog and e for verification environments; SystemVerilog Assertions (SVA) and Property Specification Language (PSL) for assertions; and Common Power Format (CPF) and Universal Power Format (UPF) for power intent.

Some of these standards began as proprietary formats that were then standardized by the IEEE, Accellera, Si2, or other organizations. Others originated directly within these standards bodies. Either way, once a standard is available, users tend to gravitate toward it quickly. Design and verification engineers take it for granted that all EDA vendors will provide support as organizations define new standards and existing ones evolve. Naturally, there's always an adoption curve as vendors implement support and users come up to speed on each new language. But everyone involved in EDA accepts the need for language standards and ultimately embraces them.

Until very recently, however, methodology wasn’t subject to standardization. Engineers who learn standard languages quickly realize that there’s more to being successful than just knowing the constructs in the Language Reference Manual (LRM). Designers need to learn how to specify their designs in a style that works well for synthesis and results in an implementation that meets their area, timing, and power requirements. Verification engineers need a great deal of guidance in order to construct environments that are efficient and effective. Design and verification reuse from project to project—or even between companies—places further demands on methodology.

Methodology evolution

Recognizing that successful usage goes beyond the constructs defined in the LRM, some standards bodies also provide users guides. These guides offer more detailed code examples while discussing basic issues of methodology. Particularly when it came to verification, however, no existing standard even attempted to tackle the complete scope of the problem. The result was that some end users defined their own verification methodologies while the rest adopted the methodology developed by their primary EDA providers. In the specific case of SystemVerilog, the three major simulator vendors each developed their own independent verification methodology.

The history of verification methodology begins with Verification Advisor (vAdvisor), a collection of best practices for the e verification language developed in 2000 by Verisity Design (now part of Cadence). This led directly to the e Reuse Methodology (eRM) in 2002. Although the e language was standardized a few years later, the eRM wasn’t. Shortly thereafter, Synopsys introduced the Reference Verification Methodology (RVM) using its OpenVera verification language. Neither the methodology nor the language was ever standardized. In 2005, Synopsys published the Verification Methodology Manual (VMM) for SystemVerilog. In 2006, Mentor Graphics introduced the Advanced Verification Methodology (AVM), which was based on both SystemVerilog and SystemC. Cadence spanned all three verification and modeling languages by expanding the eRM into the Universal Reuse Methodology (URM) in 2007.

Although each of the three vendors’ methodologies had success within their own customer base, each ran only on its own simulator. There was no attempt at cross-vendor support or any form of standardization. The first key step in this direction happened in early 2008, when Cadence and Mentor delivered the first SystemVerilog version of the Open Verification Methodology (OVM). Like its AVM and URM predecessors, the OVM was provided as open source. But it was the first methodology to be tested and guaranteed to work on multiple simulators. Although the OVM has enjoyed
wide adoption, it can be regarded only as a de facto standard because it wasn’t directly endorsed by any organization. This all changed with the 2010 introduction of the Universal Verification Methodology (UVM), which evolved directly from the OVM (see Figure 1).

HISTORY OF THE UVM

The first serious industry attempt to standardize a verification methodology began in 2008, when the Verification IP Technical Sub-Committee (VIP-TSC) was formed within Accellera. It was already clear that the OVM would be very successful. Given its longer history, there were many VMM users as well. Thus, the initial focus for the TSC was to figure out how VMM-based VIP could function in an OVM verification environment (and vice versa). Cadence, Mentor, and Synopsys worked with representatives of numerous user companies to define and validate a standard interoperability solution to link the two methodologies.

The second focus for the Accellera TSC, which is still in progress, is to define the UVM—a single standard verification methodology supported by all major simulation vendors and meeting the diverse needs of the user community. After an extensive process of due diligence, in which many members presented their requirements for such a common methodology, the committee voted to base the UVM very closely on the 2.1.1 release of the OVM. The initial release, UVM 1.0 EA, came out in May of this year. It contains a few small additions beyond OVM 2.1.1, but otherwise is exactly the same methodology and building-block class library (with the “ovm_” references changed to “uvm_”). Thus, the initial UVM release is ready for production use. In fact, it has already been adopted by many projects.

Current Accellera plans call for a “UVM 1.0” release before the end of 2010. Exactly what additional features will be included is the subject of much debate at TSC meetings. But the chairs have pledged strong backward compatibility to ensure that current users can upgrade easily. The future of the UVM is bright with many possibilities for expansion over time. Cadence expanded the OVM to include e verification environments and SystemC models. It has pledged to do the same for the UVM for possible consideration by Accellera. Other potential areas for extension include low-power verification, mixed-signal simulation, simulation acceleration, and hardware-software co-verification.

ASPECTS OF UVM STANDARDIZATION

As the first methodology being standardized by any organization, the UVM is breaking new ground. However, its standardization process is a natural outcome of previous EDA standards. It’s important to distinguish the three key aspects of the UVM provided by Accellera: the building-block class library, Class Library Reference, and User Guide. Strictly speaking, the “standard” consists of only the Reference, as it defines the interfaces and functionality of the class library elements. Figure 2 shows an example of a UVM-compliant verification environment in which the components and the interfaces among them are defined by the library.

Although the User Guide provides important supplemental information to construct such environments, it isn’t technically part of the standard. The library itself is regarded by Accellera as a reference implementation. Any other proper implementation of the Class Library Reference also would be considered UVM-compliant. Many users run the
Accellera UVM library unmodified. Because it's provided as open source under the popular Apache 2.0 license, however, users can enhance it. In addition, EDA vendors can tweak it for maximum performance with their tools. This is similar to the approach that Accellera took with its Open Verification Library (OVL), for which the manual defines the standard but reference implementations are provided.

The imminent standardization of the Universal Verification Methodology (UVM) within Accellera is a major step forward for the EDA industry. This is the first time that EDA standards have been extended beyond interfaces and languages to encompass a full methodology with a supporting library and documentation. However, this novel step doesn't mean that there's any risk in the immediate adoption of the UVM. The well-proven OVM base and the rich history of EDA standards development built a firm foundation upon which Accellera could take this next step. In doing so, it will ensure success for VIP-TSC and UVM users alike.

Thomas L. Anderson is group director, verification product management at Cadence, where his responsibilities include all verification software products. Anderson has presented more than 100 conference talks and published more than 150 papers and technical articles. He holds an MS in electrical engineering and computer science from MIT and a BS in computer systems engineering from the University of Massachusetts at Amherst.