Survey Report on Model Driven Engineering in Semiconductor Domain

R Swaminathan
Director - Technology
CircuitSutra Technologies

Dec 5, 2017

The following report is based on an email survey conducted by CircuitSutra in Nov 2017. Close to eighty people were contacted and sixteen replied, giving a response rate of about 20%. The respondents' composition is six Engineers, four each of Architects and Managers and two MD/CEOs. Eleven of them belong to Semiconductor field, two from EDA and one each from Tier-I supplier and Deep Learning field.

Disclaimer: *This is not a scientific survey, but a quick dipstick survey drawn from LinkedIn Contacts, who are working in the Design and Development in Semiconductor and allied domains.*

Introduction

Systems engineering is a multidisciplinary approach for developing solutions to complex engineering problems. The continuing increase in system complexity is demanding more rigorous and formalized systems engineering practices. In response to this demand, along with advancements in computer technology, the practice of systems engineering is undergoing **a fundamental transition from a document-based approach to a model-based approach**. In a model-based approach, the emphasis shifts from producing and controlling documentation about the system, to producing and controlling a coherent model of the system. **Model Driven Engineering** (MDE) can help to manage complexity, while at the same time improve design quality and cycle time, improve communications among a diverse development team, and facilitate knowledge capture and design evolution.

> *"Computer Science is a science of abstraction – creating the right model for a problem and devising the appropriate mechanizable techniques to solve it." - Alfred Aho.*

Visual Modeling formalizes real world products and processes to a graphical representation. The additional complication for development of real world engineering systems is to satisfy physical constraints on behavior. As future engineering systems will be more complex than today, designers would need to be more productive just to keep the duration and economics of design development in check. Major increases in design productivity have been achieved by solving problems at higher levels of abstraction and maximizing opportunities for adaption by delaying decisions on implementation for as long as possible.

Two of the most popular Visual Modelling languages are UML[1] (Unified Modeling Language) and SysML[2] (System Modeling Language). The goals of both these languages are to provide users with ready to use, expressive visual language (notation) so that they can describe and exchange meaningful models.

[1] https://www.tutorialspoint.com/uml/ [Accessed: Nov 13, 2017]
[2] http://www.omgsysml.org/INCOSE-OMGSysML-Tutorial-Final-090901.pdf [Accessed: Nov 13, 2017]

Modeling solution is a combination of modeling language(s), methodology and tool(s) that together provide a productive infrastructure for applying model driven development in the context of an organization.

UML is the de facto standard for visualizing, specifying, constructing and documenting software systems. It includes 13 different diagrams. **SysML**, implemented as a UML 2.0 profile, is dedicated for modeling complex systems that may include hardware, software, information, personnel, procedures and facilities. It includes 9 diagrams. Note that SysML is neither a modeling methodology nor a framework.

One in two respondents were aware of the term “MDE”, and one in four have used UML. SysML, the younger sibling of UML, was used by one in five.

Model Use cases

Thirteen of sixteen respondents use high-level Abstract Models in their work. If we try to drill down on the kind of use cases these models are used for, we arrive at the following graph shown in Figure 1Model Use cases (Percentage).

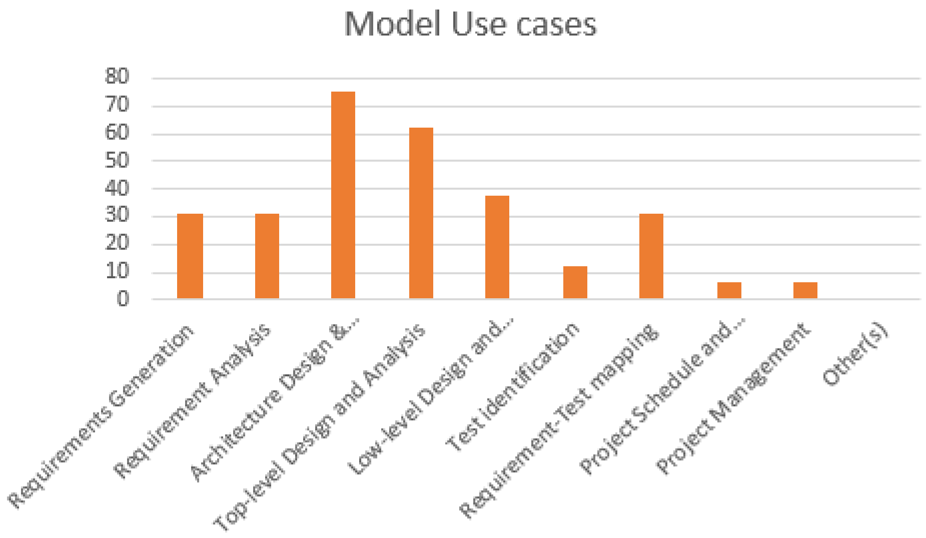

Figure 1Model Use cases (Percentage)

“Much of the essence of building a program is in fact debugging of the specification.” – Fred Brooks.

About one in three respondents use high-level abstract models for Requirements Generation and Analysis. The usage peaks for Architecture Design and Analysis with nearly three in four people investing in models. Top level Design and Analysis is also strong with over 60% using models. This is consistent with the sweet spot of the high-level abstract models based on C/C++/SystemC etc. When we move to low-level design and analysis the usage of models drops to just above 30% indicating that now the action moves to HDLs like System Verilog, VHDL etc. Testing finds a marginal role with models hovering a shade above 10%; but the Requirements to Test mapping still has a good play at above 30%. The Functional to Test mapping is a must-have in many industry standards like CMMI, ISO, etc. and these models have a handy way to map and verify that nothing slides through the gaps.

Management of Projects and Project Schedules using models hasn’t picked up much traction. These are probably handled through more traditional tools like MS Project or Agile methodology.

MDE Tools

The following Figure 2MDE Tools (Percentage) shows the popular tools used for MDE.

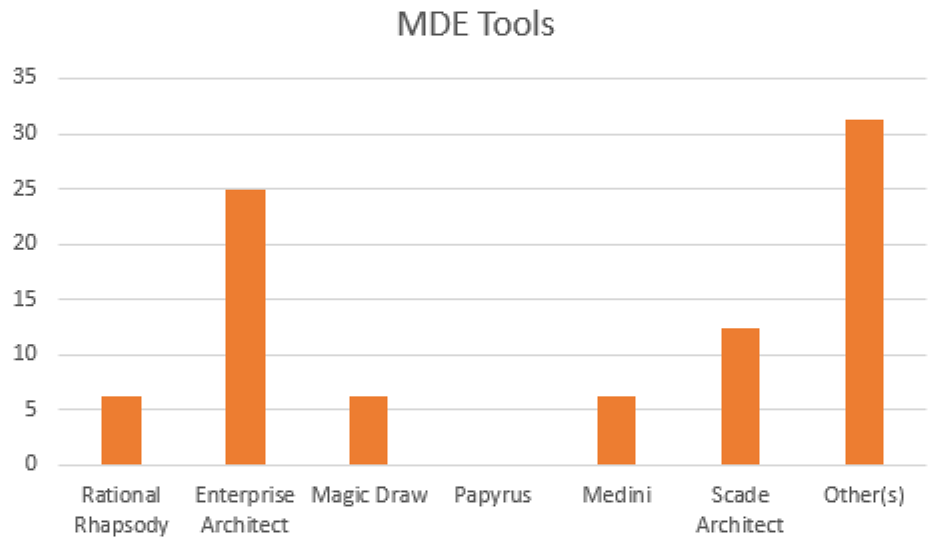

Figure 2MDE Tools (Percentage)

> “Man is a tool using animal. Without tools he is nothing. With tools he is all.” – Thomas Carlyle.

Commercial tools seem to outweigh open source options (like Papyrus). One strong possibility is because the companies need assured support for feature requests and bugs. Of the commercial tools, Enterprise Architect seems to be the preferred option, with the second favorite Scade Architect coming with about half the popularity of the leader. Enterprise ready and feature rich (UML 2.0/SysML) tools like Rational Rhapsody and Magic Draw don’t seem to be very popular in this segment.

A surprising observation about this Manhattan is that the Other(s) category seems to dominate. While this segment is a catch all, ‘Inhouse’ tools dominates here. This could only mean that while MDE is found to be useful, in fact useful enough to deploy internal resources to work on tools, the commercial offerings are found lacking in features, or positioned at a price point that is not justifiable for them.

Here is a list of reasons given for using/not-using MDE tools:

+ Traceable and graphical decomposition and documentation of system
+ Architecture Design and Analysis (2 respondents)
+ Easier to create and maintain Automation
+ As Input to Requirements list.
– Not aware (2 respondents)
– Our interest mostly lies in more accurate analysis of system traffics, we doubt MDE tools achieve the target accuracy

Current Gaps

Our survey also tried to find out about the biggest gaps in the MDE tools as perceived by the participants. Here is how they responded:

- Connection to executable specification/simulation and low-level flows.
- We need to have language which can generate low level model from high level abstraction. The reverse flow is also very interesting.
- We would like to see higher abstracted model from design for redesign of the architecture.
- Model for architecture definition and then generating specification out of it. There is a gap between the architecture and specification stage.
- Use case is one important part of Architecture study. We should have flow where the use case is mapped with the model and flexibility provided on this. Basically, Task mapping especially chaining the task.
- Bridges between tools to maximize reuse.
- Matlab/Simulink Vs HDL there is a big disconnect.
- Power, Timing, Floor-Planning can have standard at Model level.

A few of the above concerns like Use case mapping to model and Requirements to documentation generation are already addressed by SysML. Methodology and tool interoperability seem to be showing up as two top challenges from the list above.

About one in ten respondents would like to see new standards to address some of the perceived limitations and three in ten are willing to participate in any standards bodies and/or review new standards, if needed.

Conclusion

In summary, Model Driven Engineering (MDE) is gaining interest in the Semiconductor Design and Verification field especially in the Requirements to High-level design continuum. Its usage in low-level design and project management is not very high. Commercial UML2.0 and SysML languages and tools are used with Enterprise Architect and Inhouse tools taking the lion's share. Lack of awareness, methodology challenges and tool interoperability challenges are the prime road blocks in wider adoption and usage of MDE tools in this domain.

The author hopes that this survey has shed some light on the status of MDE in the field of Semiconductor Design and Development, and encourages Engineers and Tool vendors to address some gaps highlighted in the user survey.

Note: Any comments or suggestions on this report can be emailed to the author at r.swaminathan@circuitsutra.com .