The Informal Nature of Systems Engineering

Abstract
This is a position paper about the relation between Formal Methods and Systems Engineering for complex computerized systems. We will argue that Formal Methods are well suited to prescribed homogeneous domains, and that systems engineering, which integrates more specialized engineering disciplines, is inherently much more informal.

We will use the waferstepper as a typical complex computerized system, the case is described at the beginning. Next we explain the discipline of Systems Engineering. In a short intermezzo the overloaded meaning of the word “formal” is discussed. The real positioning is given in two steps: first we elaborate the informal nature of Systems Engineering and then we discuss the relation to Formal Methods.

Case
The development of a new generation of wafersteppers, the ASML Twinscan family, is used as illustrative case. A waferstepper is a large (ca. 30 m^3), costly (ca 10M$) and complex system used in the lithographic process of a semiconductor fab. The main function of the system is to replicate a circuit pattern many times on a wafer, by stepwise exposing the wafer. The early wafersteppers exposed fields sequentially, where the wafer is not moving during the exposure. The actions are then simply to move, stop, and expose. The most recent wafersteppers use the scanning principle. Scanning is based on exposure through a slit, while the reticule with the original and the wafer move harmoniously. Time and position are much more directly coupled in the scanning exposure.

The key drivers of lithography are Critical Dimension (CD) control, overlay, and productivity. CD control is the variation of the line width or gate size, for 130nm line width a typical CD control value is 10 nm. Less variation is better, minor variations may cause a significant power consumption problem in the final integrated circuit. Overlay is typical 45nm for 130nm line width. Smaller overlay values are better, allowing a denser design and hence more chips per wafer. Productivity for these systems is expressed in terms of exposed wafers per hour, typical 100 300mm wafers/hr. The productivity of the waferstepper is directly related to the cost effectiveness of the semiconductor fab, the value of the waferstepper is more or less proportional with the productivity.

Characteristic for the semiconductor equipment market is the fast evolution, expressed in Moore’s law. The exponential performance improvements dictated by this law translate in exponential improvements of CD control and overlay. A twofold improvement is required every four years.

The customer level performance is achieved by budgeting the most important performance targets. For example the overlay budget for wafersteppers in 1997 was decomposed in 5 decomposition steps in individual contributions. For instance tracking error contributions and stability requirements are specified in nanometers. Such a budget is an abstraction of the actual machine behavior. A typical overlay budget contains about

25 numbers, while hundreds of components and parameters have somehow impact on the final overlay.

Systems Engineering

The Systems Engineering discipline is an integrating discipline. Systems Engineering integrates and guides mono-disciplines, such as mechanical engineering, electrical engineering, and software engineering, to create reliable systems. The Systems Engineering discipline comprehends multiple approaches:

- well defined formalized Systems Engineering methods
- strong process focused
- “common sense”, based on human experience and intelligence

A balance of these three approaches yields successful products. In this document we will discuss this balance and especially the, often underrated, informal side of Systems Engineering.

What is “formal”?

Industrial discussions about the use of *formal* methods often derail due to the ambiguity of the word *formal* itself. In industrial context formality is often used in organizational sense: what are the formalized processes, responsibilities, roles, et cetera. Formalized processes facilitate well-known problems of heterogeneous nature. The scientific based *formal* methods use the mathematical sense of formality. Science based formal methods facilitate specialized well-known problems of homogeneous nature. These formal methods provide proven solutions to problems fitting in the limited specialized area covered by the method. For instance Rate Monotonic Scheduling guarantees real-time performance for repetitious tasks with well-defined processing times and deadlines.
The Systems Engineering community is strongly focused on (formal) processes. However, most system level problems are ill defined and very heterogeneous. The overlay specification of 45nm, for example, sounds quite well defined. However, this specification is only valid in unique well-defined measurement circumstances. The realized overlay in actual production lots is a function of hundreds or thousands of parameters. Customer satisfaction is determined by actual overlay performance, not by the artificially defined acceptance specification. We will discuss the consequences of these characteristics in relation with *formal* methods.

The Informal Side of Systems Engineering

The key performance of the waferstepper, in terms of CD control, overlay and productivity and the design choices depend on many context aspects, such as the production environment, the business, the human stakeholders, and the many involved technical disciplines.
The yield and productivity of a lithography cell depends on the waferstepper, but also on many other aspects in the context of the waferstepper. For example, the wafer and the reticule themselves influence the performance as well as the measurement, processing and logistics of wafers and reticules.
In the business context a balancing act is performed between yield and CD control with a significant impact on the final chip performance (power and speed). The business context is a complex playing field with many players, such as equipments vendors, system

integrators, lease companies, fab designers, consultants, mask makers, resist makers, and wafer makers and many different kinds of customers: design houses, foundries, and vertical integrated companies.

The human context is full of stakeholders, both internal as well as external. All stakeholders have their particular concerns, interests, rhythms, and contributions. The design emerges from a complex psychosocial interaction between all these stakeholders. Problems arising from the complexity of this context become visible in a rather late stage of development: during integration or worse in the customer's fab. The dynamics, the uncertainties, the unknowns and the heterogeneity of these systems engineering aspects do not fit with rigorous formal methods. Informal, "common sense" and experience-based methods are used mostly here.

Where do Systems Engineering and Formal Methods meet?

In industrial practice some huge gaps exist between tools and methods of the involved disciplines. Worse is that the involved engineers are often unaware of these gaps and use their own frame of reference in the discussion with other disciplines. For example, software people claim to have a proven implementation, but at the same time cannot answer the simple, but crucial, question how much time is needed per function. The ideal situation would be that disciplines have sufficient mutual understanding to communicate and cooperate. The gap between Systems Engineering and Formal Methods in Software Engineering in industrial practice is rather large at this moment.

Conventional disciplines, such as mechanical engineering, electrical engineering, and computer science, have a rich collection of formalisms, techniques, tools and methods. The elements in this collection work on well defined problems in a well-defined manner. In product creation less well-defined problems occur when multiple disciplines jointly realize some functionality. Techniques, tools and methods exist at the multi-disciplinary level. These techniques, tools and methods are less well defined than at the mono-disciplinary level. When the focus is limited to a single objective the problems and means are well defined, but soft. At the system level, where multiple objectives have to be achieved simultaneously, the problems are ill defined and the methods become rather soft. The natural habitat of formal methods is in the category of well-defined problems, while Systems Engineering is heavily involved with the ill-defined problems, with multiple objectives and many contributing disciplines. Systems Engineering and formal methods can be complementary, when formal methods remove risks of well-defined problems.

A system can be described and analyzed at different levels of abstraction. The static description of today's embedded systems contains tens of millions of details, such as lines of code, components, and connections. The challenge of product creation is to translate a few key requirements in several design steps in the tens of millions details at the lowest level of abstraction. The most detailed design steps are mono disciplinary, for example transforming an interface and behavior specification of a class into hundreds to thousands lines of code. However, at a higher abstraction level design trade-offs are made to allocate functionality to technologies and components, typical multi-disciplinary design. System engineers have the responsibility for the integral system performance and functionality: the integration of multi-disciplinary components and subsystems into a system. For example the system engineer reasons at the highest abstraction level about exposure in terms of a light source, reticule, lens and wafer, and about system functionality in terms of 3 key parameters: overlay, CD control, and productivity. The higher-level abstraction is transformed into models and budgets with tens of contributing elements. Finally, the waferstepper contains 10 million lines of code to realize the required system behavior and performance.

In the following postulates we position formal methods in relation to systems engineering. The purpose of this positioning is to create mutual understanding of the contribution of these disciplines.

Postulate 1: Formal Methods in industrial context work only at the more detailed mono-disciplinary abstraction levels, with well-defined problems. Examples are communication protocols and scheduling strategies.

Postulate 2: Inventors of formal methods are capable to apply their personal strengths also at a much higher abstraction level. These inventors are: analytical, structural, firm of principle, and consistent. The formal methods themselves do not really contribute to the Systems Engineering means; the personal strength of formal people can contribute.

The research field of multi-disciplinary design is tackled with three research approaches. The *scientific* approach is to extend the existing body of knowledge with small increments. Every increment is well founded. A lot of the methods and techniques available in the existing body of knowledge can be used with adaptations at the multi-disciplinary design level. We call this approach *borrow & adapt*. The third approach is *heuristic*: observe the system engineers in the industrial context and make the implicit experience explicit. Based on the consolidation of the state-of-practice many research questions can be formulated. Such a consolidation starts with observations and descriptions, and in the long term, after a lot of research, will be turned into well-structured methods with clear fundaments.

Conclusions

Systems engineering takes place in a very heterogeneous environment, Systems engineering is the art of ignoring details. Formal Methods provide a systematic and accurate approach, and works on well-defined homogeneous problems.

Systems engineering can use formal thinking: *borrow & adapt*. An example is System Level modeling; systematic and structured like formal methods, but not proven or very accurate due to the inherent uncertainties at system level Formal methods, applied at specific homogeneous niches, provide input to Systems Engineering work at multi-disciplinary level. Systems Engineering sets, the other way around, the boundaries for the application of Formal Methods for partial system problems.