Modeling Hierarchy, Coping with the Dynamic Range from Design Details up to Business Metrics;

Illustrated by a Semiconductor Case.

Gerrit Muller
Embedded Systems Institute
Gerrit.muller@embeddedsystems.nl

Abstract. A system functions as part of a broader enterprise. For the design of a system understanding is required of its purpose within the enterprise, as well as of its internal functioning. Models are a means to create and capture understanding. Many different models are needed during the design of a system, from broad enterprise models down to detailed implementation models of components or functions. In this article we show the hierarchy of models, their relations and the level of detail in these models. We use the semiconductor industry as an example.

Introduction. System design decisions are based on the purpose of the system in the system context and on the solution options available in the design space. For many systems the ultimate goal and the ultimate atomic design decision are completely different worlds, light years separated from each other. Nevertheless, a seemingly detailed design decision may have impact on the ultimate value proposition.

Semiconductor case. We will use the semiconductor industry as an example to illustrate this dynamic range. The value chain of the semiconductor industry ends at the consumer who is using electronic appliances to be entertained with content or to do useful things by using services on the electronic appliances. At the other end we will focus on the equipment supply chain, where high-tech components are designed and manufactured. These high-tech components are used in one of the critical production machines in a wafer fab. The wafer fab is the production facility of chips that are used in the electronic appliances. This full chain is shown in Figure 1. Note that we have selected a sub (sub) set of this chain. In the later diagrams a number of related branches will show up.

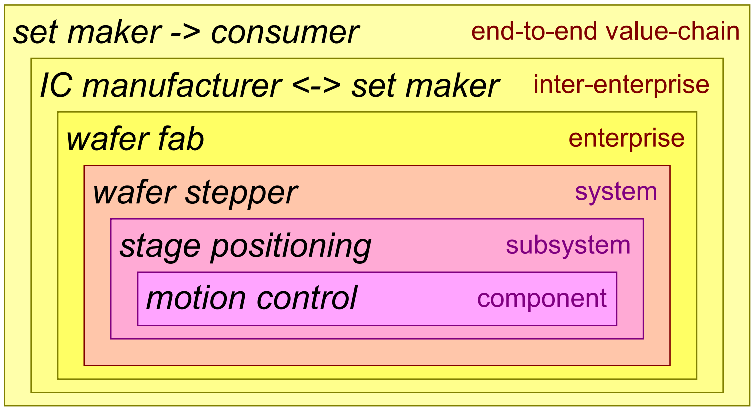


Figure 1. Hierarchical Levels in Semiconductor Industry

Engineering Principle: Decoupling. The decoupling principle in engineering is that complex problems are simplified by layering and decomposition. Layers and components of a composition are chosen such that coupling is minimal between layers and components. We use a layer or a component by defining an abstracted specification, hiding it internal complexity. In other words the purpose of this engineering strategy is that we don't have to know the internals of layers and components.

The role of systems engineering and architecting. Systems engineers and architects are responsible for the system design. In that role they are also responsible for the *decoupling* mentioned above. Thanks to the decoupling principle, most engineers can live and work in a well-defined context. Unfortunately, the abstraction of component or layer internals is never completely ideal. Imperfect abstractions result in coupling between components and layers.

Systems engineering best practice: identification of cross-cutting concerns and key-drivers. Another responsibility of the systems engineers and architects is the integral performance and behavior of the system in its intended context. This responsibility is the counterpart of decomposition: integration or synthesis. Systems architects and engineers have to be able to reason how components and layers operate together and how the system will operate in its context. Systems people are often involved with cross-boundary issues. The essence of their job is to prevent these issues by design, while at the same moment we know that imperfections will always cause these cross-boundary issues. The systems architect identifies crosscutting concerns to be able to make a system design that facilitates these crosscutting concerns. The systems architect also identifies key-drivers of the system of interest and of the customer. These key-drivers are used to focus on the main specification and design issues. In practice key-drivers are often the starting points of crosscutting concerns.

Semiconductor case: from equipment to consumer. Figure 2 shows the end-to-end value chain of the semiconductor industry. In fact two repeatable chains are present:

- **Material flow**: Raw materials transformed into chips by the IC manufacturer. A set maker produces electronic appliances, finally used by consumers, uses these chips.
- **Content flow**: Content that is created and published. Finally consumers use the content via a distribution channel.

As an example we take a portable multi-media player. From consumer point of view main concerns are content quality and availability, and battery life-time (how long can I play without reloading the batteries). The content provider is mainly concerned with protection of the content by means of Digital Rights Management (DRM). DRM is perceived as the solution against unlimited and unpaid copying of contents. The set maker has to satisfy both consumer and content provider, meaning that both DRM and power consumption of the appliance are main concerns. The set maker is also very cost sensitive, caused by a very competitive market that threaten the margin of set makers. The IC manufacturer is striving for high performance chips, with affordable yield and high productivity. This triangle directly corresponds to the financial results of the IC manufacturer. The equipment supplier has to deliver very accurate systems to enable high performing chip manufacturing with high yield, and high throughput system enabling high fab productivity. The competitiveness of the IC manufacturer depends on its ability to design and manufacture high performance (DRM), low power (battery-life), low cost (yield and productivity) ICs.

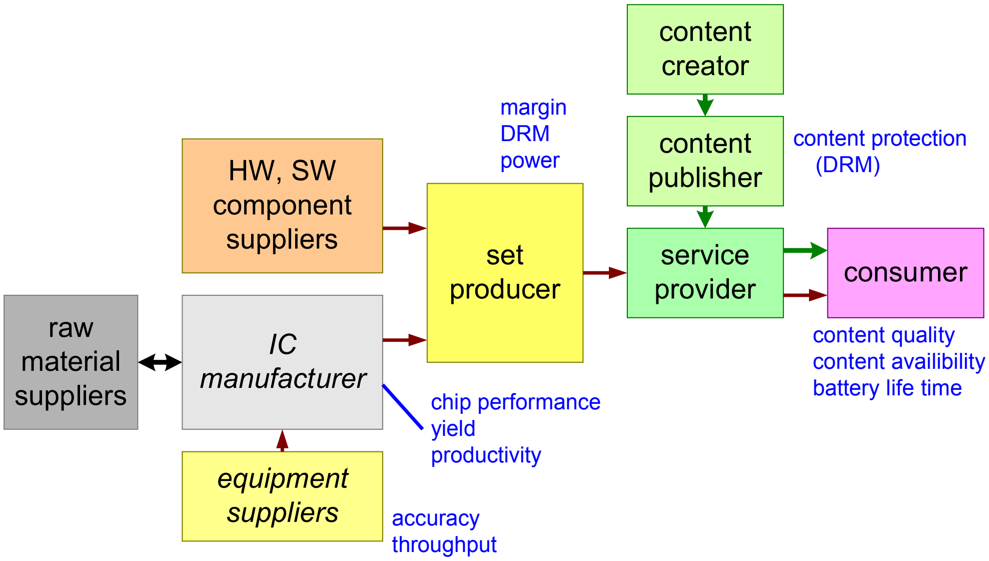


Figure 2. Example of a simplified end-to-end value-chain

Ecology of enterprises. If we zoom in on the IC manufacturer, we become aware of an entire ecology of enterprises that organically interact to produce chips for the set-makers. In Figure 3 we show this inter-enterprise level. The pink cloud indicates the working area of the IC manufacturer. The cloud symbol is used, because the exact boundaries of the IC manufacturer depend on the company. TSMC, Micron, Intel, to name a few, have all different company boundaries. However, all these IC manufacturing companies operate expensive fabs (costing more than 4G$ nowadays). The generic flow as shown in this figure is also the same for all chip producers. The entire chain from logic design to chip packaging is critical for the competitive factors mentioned the end-to-end value-chain.

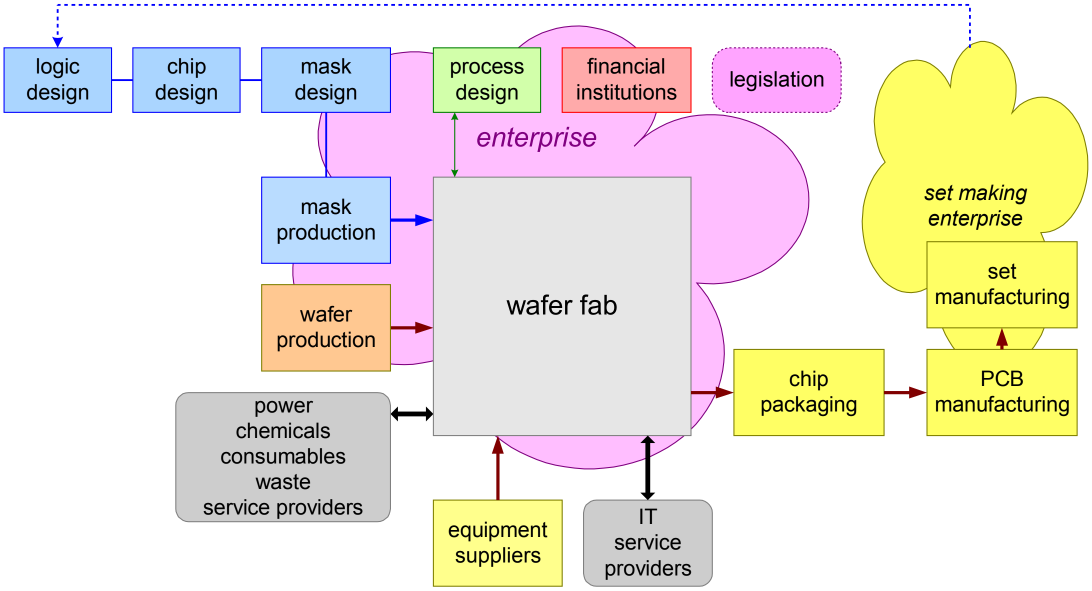


Figure 3. Inter-Enterprise: Ecology of enterprises

Chip producer as enterprise. Zooming in on the next layer we enter the IC manufacturing company, where the wafer fab is central. Within the wafer fab we encounter many completely different functions or disciplines: human resources, finance and administration, logistics, process control, IT, facility management, and of course the core production line from wafers to chips. This layer is visualized in Figure 4. Within the wafer fab critical functions to achieve yield, low power and high performance chips are present. These functions are mostly realized by the following systems in the fab: *exposure*, *metrology* and *advanced process control*.

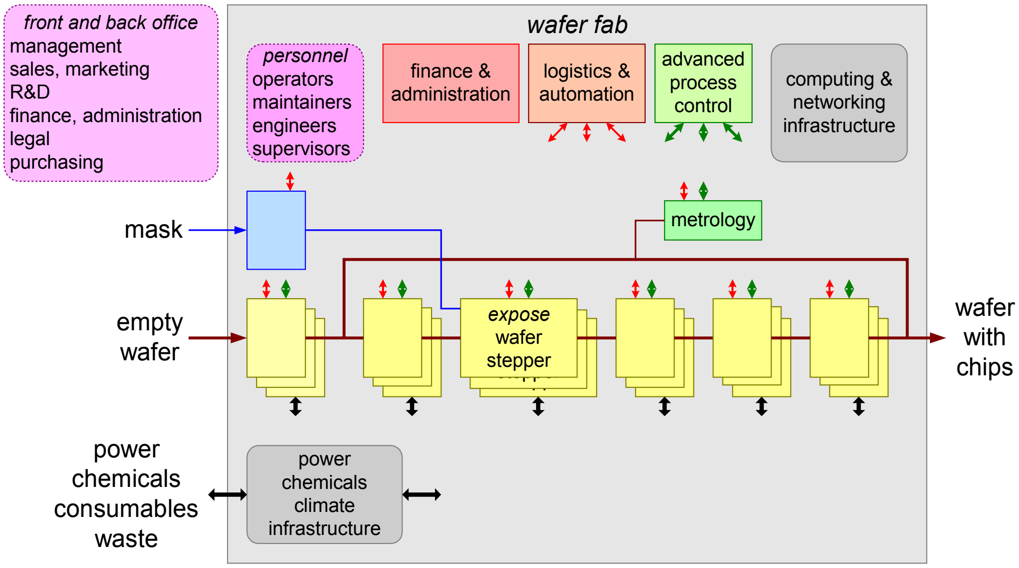


Figure 4. IC manufacturing Enterprise

Wafer stepper. The lithography system is one of the most critical systems in the wafer production line. The popular name for a lithographic system is wafer stepper. The waferstepper exposes the wafer step by step, hundreds of exposures per wafer per wafer production cycle. A wafer stepper is in itself a very complex high-tech system. This system with outer dimensions of meters exposes structures on a wafer with nanometer precision. Not only is the system extremely accurate, it is also extremely fast: hundreds of wafers are exposed per hour.

The key performance parameters of a wafer stepper are *overlay*, *critical dimension* (directly related to transistor gate size), and *productivity*. A direct relation exists between the key performance parameter of the waferstepper and the competitive success factors mentioned in the end-to-end value-chain. For example *critical dimension* has a strong impact on chip performance and relates also to power consumption.

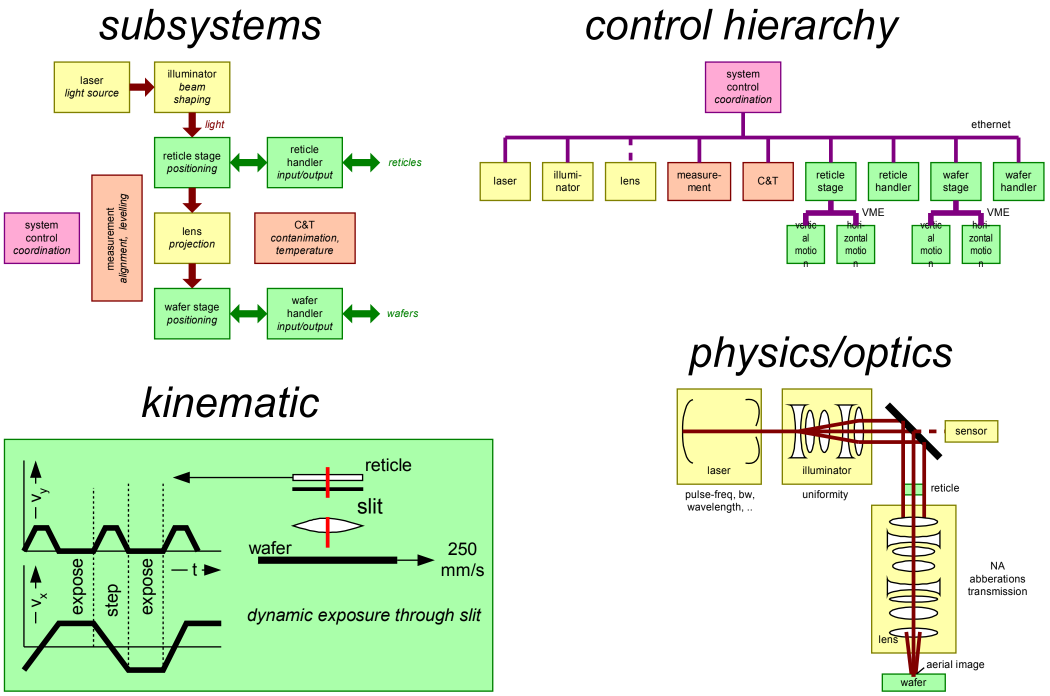


Figure 5. Wafer stepper system views

Figure 5 zooms in on the waferstepper itself and shows four different views of the system internals. The first view shows the further decomposition in subsystems. Also shown are the control hierarchy, the basic kinematic performance and the optical path through the system. The kinematic and the optical performance determine the system level image quality (accuracy) and the system level throughput. The *critical dimension*, key performance parameter of a wafer stepper, is a complex function of kinematic and dynamic behavior, and the optical characteristics of the waferstepper.

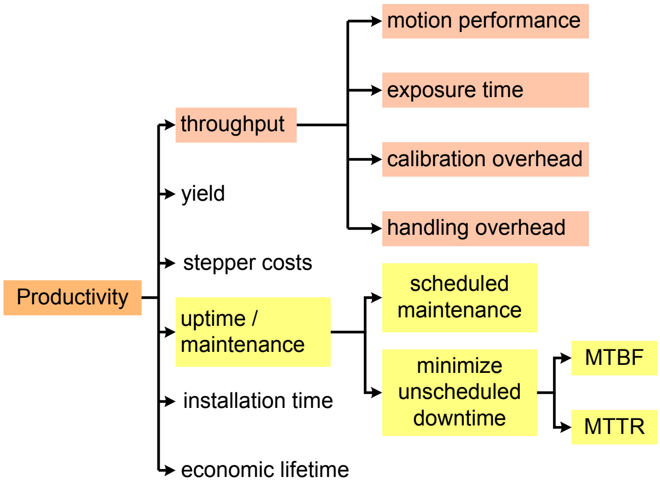


Figure 6. Productivity Key Driver Decomposition

Figure 6 shows the decomposition of customer level productivity. Via system throughput it is shown that motion performance is one of the critical contributions to productivity.

Key-drivers and crosscutting concerns. We encountered DRM as content provider key-driver and battery life time as consumer key-driver. Figure 7 shows the relations of the key drivers at the different scopes discussed so far. This graph shows that cost, performance and power are crosscutting concerns throughout the value chain.

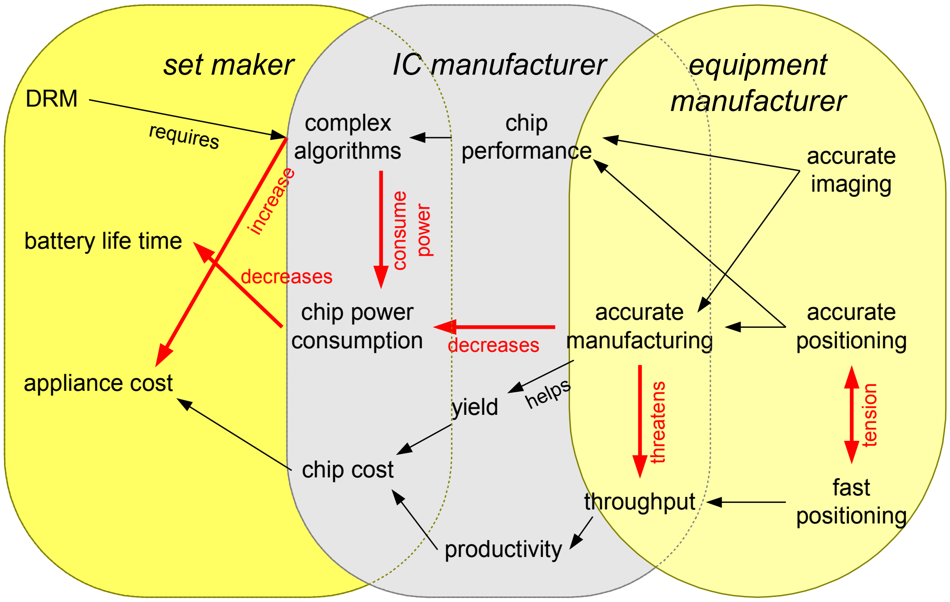


Figure 7. Relating key-drivers from consumer and content industry to key-drivers of IC manufacturer and equipment manufacturer.

Motion control. In the previous section we showed that *overlay*, *critical dimension* and *productivity* are key performance parameters of the wafer stepper system. One of the crucial subsystems for these parameters is the wafer stage. A motion controller controls the wafer stage. Figure 8 shows the wafer stage positioning and a few of its main principles, such as fast synchronous feedback to control 6 degrees of freedom.

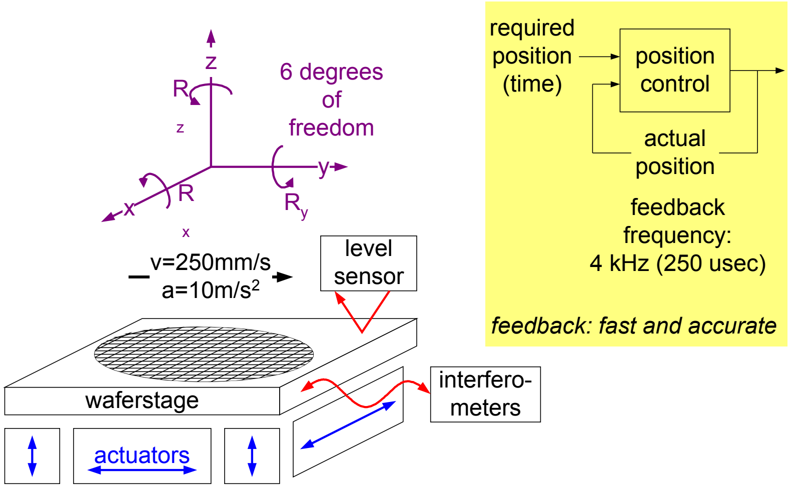


Figure 8. Wafer stage positioning

Motion control platform. The motion control function runs on a dedicated processor platform. The performance and accuracy of the motion control function depend on the capabilities of this platform. One of the key design parameters at this level is the feedback frequency. The platform itself is again layered: computing hardware, with a small operating kernel. On top of the operating kernel support functions are build and finally some control algorithm performs the end-to-end control function. The feedback frequency depends on the performance characteristics of this hardware and software platform.

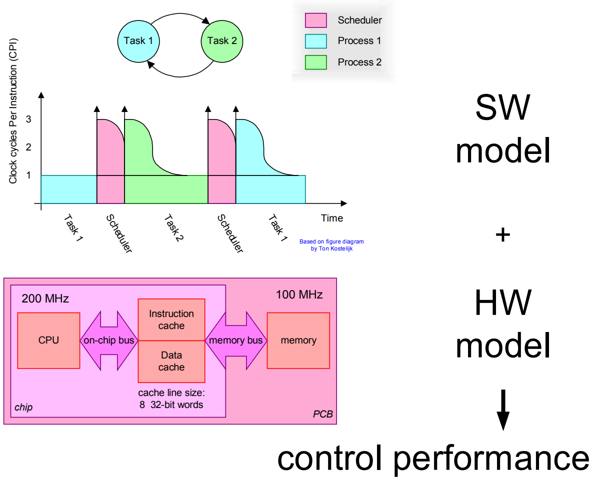


Figure 9. Performance of motion control platform modeled as interaction between software and hardware.

Figure 9 shows a SW model and HW model that together are used to model control performance. It also shows quantification of these models that can be obtained by measuring layers in HW and SW. The control algorithm performance depends on hardware characteristics, such as cache and instruction set, but also on efficiency of the higher layers, such as interrupt handling time and compiler efficiency.

Revisiting the role of systems people. We have journeyed from a very broad and complex consumer and producer world down to a highly detailed technical world of caches and compilers. We postulated that the systems designer strives for decoupling: cache performance of the motion control platform ought to be decoupled from consumer electronic appliance experience. For most detailed design choices the chosen abstractions and decoupling mechanisms work well. However, once in a while a design choice is highly critical and has a big impact on next layers.

For example the power consumption of individual transistors is very critical for the overall electronics design of electrical circuits. At the same time this transistor-level power consumption is extremely sensitive for the imaging accuracy. Sub-nanometer variations may change transistor performance with a factor, due to leakage problems. Power consumption of appliances at consumer level is one of the critical success factors. In this example power consumption is so critical that it may violate our decoupling principle. Or to put this in a different perspective: very detailed design choices in the wafer stepper may result in critical competitive advantage at electronic appliance design level.

Modeling and quantification. We have merely touched upon the layers of the semiconductor case, the format of this paper does not allow for a more in depth description. In real life we also have to deal with severe time and cost constraints. As systems designer we have to understand layers above and under our own layer, in order to make the right abstractions. To achieve decoupling, we have to create understanding across the boundaries. The means to create this understanding are modeling and quantification. Models are abstractions that describe behavior and functionality of part of the problem space. For example we can make a productivity model of a wafer stepper in a fab context, and we can also model the throughput within the wafer stepper as function of subsystem performance.

Crucial to making these models deployable is to make the models concrete with quantifications. For example a typical wafer fab performs 60,000 wafer starts per month. Every wafer is typically 40 times exposed during the production cycle. These quantified models enable further reasoning by using more business type models, translating productivity in turnover and profit.

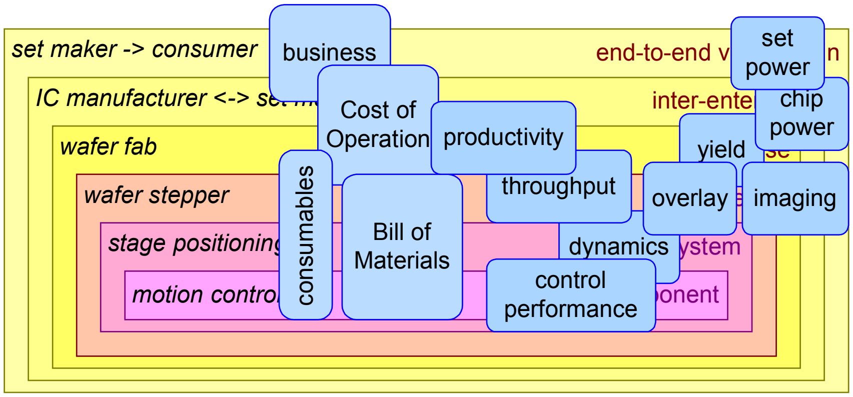


Figure 10. Related crosscutting models at different levels

Figure 10 shows how the key-drivers and crosscutting concerns discussed so far result in models at different levels in the abstraction hierarchy. The challenge is to identify a minimum set of models across the hierarchy that is needed to create a competitive solution. Figure 11 provides a number of criteria to identify these models. The main message in this figure is that a model has a clear goal: to be able to analyze system requirements in relation to design choices. To provide more guidance criteria are provide to select relevant design choices and requirements. The last element of guidance is the tension. Useful models often facilitate the analysis of choices or requirements that have some form of tension.

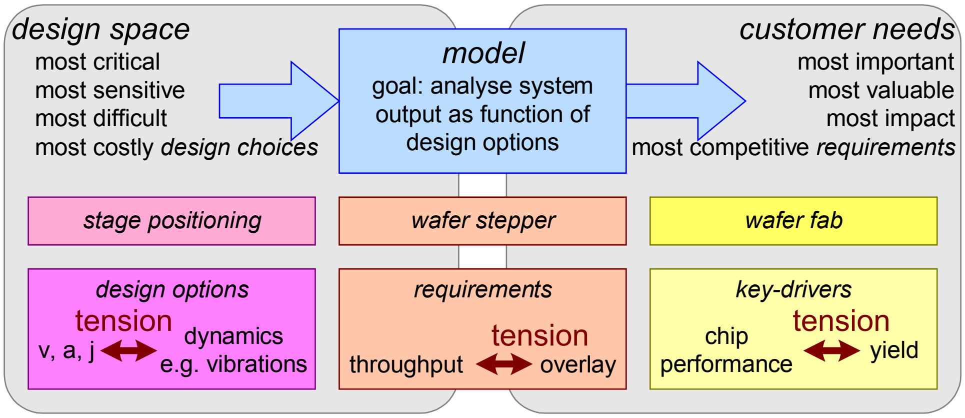
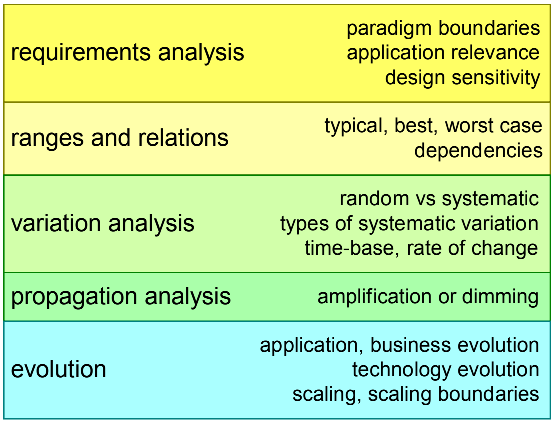


Figure 11. Models to be made help to analyse and understand the most important, valuable, impact, and competitive requirements of the system in relations to the most critical , sensitive, difficult and costly design choices.

Figure 12 shows that quantification in itself opens a whole world of new questions: different types of analysis, putting number into perspective with ranges and credibility and the evolution of numbers over time. This insight results in a further refinement of the guidance. What exactly do we need to analyze: the feasibility of a requirement, the allowed variation, worst case behavior, or the evolution over time? In general we recommend to start as simple as possible and to extend the model when needed.



requirements analysis	paradigm boundaries application relevance design sensitivity
ranges and relations	typical, best, worst case dependencies
variation analysis	random vs systematic types of systematic variation time-base, rate of change
propagation analysis	amplification or dimming
evolution	application, business evolution technology evolution scaling, scaling boundaries

Figure 12. Quantification issues.

Summary and Conclusion. We have very quickly shown the many layers of abstraction in one of the chains of the semiconductor industry. Systems engineers and architects strive to decouple layers as much as possible. However, at the same time they have the responsibility to understand critical relations across boundaries. For that purpose a systems architect will model a few layers above the system of interest and a few layers within the system of interest. The means to select the models to be made are: key-drivers and crosscutting concerns. By articulating the goal of a model in terms of inputs (relevant design choices), outputs (relevant requirements), and tensions a model can be kept as simple as possible. Layering of models (e.g. Figure 10) also helps to keep individual models as simple as possible.

The increased scope of projects increases the systems engineering challenge: how to keep complexity manageable by abstraction and decoupling and when to zoom in or out to find the appropriate solution? The direction we provided in this paper is to use abstraction and simplification: modeling! As guidance we provided criteria to minimize the modeling effort by focusing on the key-issues.

Author Biography

Gerrit Muller.

Gerrit Muller received his Master's degree in Physics from the University of Amsterdam in 1979. He worked from 1980 until 1997 at Philips Medical Systems as system architect. From 1997 to 1999 he was manager Systems Engineering at ASML. From 1999 - 2002 he worked at Philips Research. Since 2003 he is working as senior research fellow at ESI (Embedded Systems Institute). In June 2004 he received his doctorate. The main focus of his work at ESI is on System Architecture methods and on education of future System Architects. Special areas of interest are:

- Ways to cope with the exponential growth of size and complexity of systems. Examples of methods to address the growing complexity are product lines and composable architectures.

- The human aspects of systems architecting (which in itself is a crucial factor in coping with the above mentioned growth)

All information (System Architecture articles, course material, curriculum vitae) can be found at:

Gaudí systems architecting http://www.gaudisite.nl/