Paper 164

A multi-disciplinary and model-based design methodology for high-tech systems[1]

W.P.M.H. Heemels, [a] **E.H. van de Waal,** [b] **G.J. Muller** [a]

[a]*Embedded Systems Institute*
PO Box 513, 5600 MB Eindhoven, Netherlands
maurice.heemels@esi.nl, gerrit.muller@esi.nl

[b]*Imtech ICT Technical Systems*
PO Box 1608, 3800 BP Amersfoort, Netherlands
evert.vandewaal@imtech.nl

Abstract

System architecting for high-tech machines is currently performed by highly experienced individuals, who often use an intuitive approach to their craft. In order to allow better co-operation, and to enable the dissemination of experience and knowledge, a framework is needed in which the methods, modelling formalisms, techniques and tools used by system architects are captured. In this paper, such a framework is proposed. Different steps in a system architecting process are given, with appropriate techniques. Examples are given from a large scale research project investigating the design of a high-speed copier. Finally, the abstract framework is mapped to a dynamic workflow model closely resembling industrial practice.

Introduction

When designing a high-tech system like a copier multiple disciplines need to make the overall design in close co-operation. For instance, the electronic design, mechanical design and software design together need to describe a consistent, functioning machine. The designs are often made in parallel by multiple groups of people, where the communication between these groups is hampered by lack of common understanding, organizational issues, politics and out-of-phase project evolution. A typical example of the latter is that the mechanical design often precedes the electronic design, which on its turn precedes the software design. In addition, the complexity of a copier (typically millions lines of codes, tens of thousands mechanical components like pinches, springs, belts, motors, bolts, etc.) give rise to many cross-disciplinary design decisions. Often choices are made which may have benefits in one discipline but disadvantages in other disciplines. To make a good decision the overall effect of such a choice needs to be evaluated, as early as possible. Therefore, a framework that supports efficient evaluation of design choices over multiple disciplines would be very beneficial.

Evaluation of design choices over multiple disciplines is one of the important features of System Architecting (SA). Typically, SA for high-tech machines is performed by highly experienced individuals, using mostly intuition and 'gut feeling'. The experience of these individuals is hard to transfer, thereby limiting the speed with which companies can develop. This way of working is effective when the project remains small and limited to one location, where a relatively small number of people are involved in the design. However, to enable the co-

---

[1] This work has been carried out as part of the Boderc project under the responsibility of the Embedded Systems Institute. This project is partially supported by the Netherlands Ministry of Economic Affairs under the Senter TS program.

operation for larger projects across multiple sites, a SA framework is needed. Even for smaller projects, such a framework is expected to speed up the design process and to reduce integration time. Moreover, a SA framework that captures the way of working of the experienced architects should enable junior architect to learn the skill of system engineering faster. Hence, in this respect the design methodology has both an educational as an industrial application character.

In SA research, some frameworks have been established (see e.g. (INCOSE, 2004), (Maier and Rechtin, 2002) or Chapter 4 in (Muller, 2004) for an overview). Also, academic research has produced techniques that could be useful in industry. However, these find very limited use, see e.g. (Potts, 1993) and (Muller, 2005a), and the need for multi-disciplinary methodologies is still large as expressed in (Muller, 2005b). In (Muller, 2005b) several reasons are mentioned that hamper the creation of such methodologies. Lack of description and lack of connection of the higher level design methodology to mono-disciplinary methods are just two. The latter one is one of the reasons why the frameworks and methodologies for large scale systems (e.g. aerospace and military) are less useful in the *technical* development and realization of high tech systems like a copier. The former means that although multi-disciplinary methods exist and are in use in the industry in various domains, their use is very implicit – typically 'gut-feeling' based as mentioned before. The consolidation of these industrial methods is very poor. The lack of explicit description means that a lot of open issues remain. Open issues erode the value of these multi-domain methods. To tackle the lack of description and connection to mono-disciplinary techniques, this paper presents an attempt to explicitly describe such a multi-disciplinary methodology and give place to mono-disciplinary design techniques. As (Maier and Rechtin, 2002) state, a high levels of complexity analytical methods are no longer sufficient and heuristics come into play. In this design methodology, heuristics and analytic rigor find their place in the high level method and the mono-disciplinary techniques, respectively. The usefulness of the proposed methodology here is largely due to the connection between the two. Moreover, by making the methodology explicit, discussions should be triggered on the open issues that require future (academic) research.

This paper is based on the findings of the Boderc project (see www.esi.nl/boderc) that was initiated in close cooperation with a copier manufacturer. Besides copiers, the methodology is aimed at what could be called 'high-tech machines' that can be characterised as professional mechatronic systems with sizes between 1 and 20 cubic meters, like electronic microscopes, wafer scanners, MRI scanners, etc. The annotated research goal of Boderc can be found below.

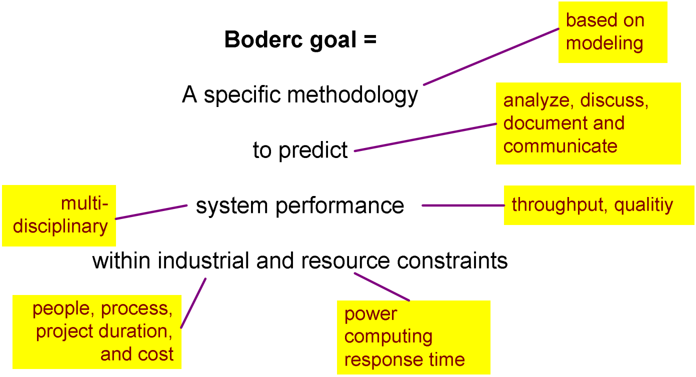


Figure 1: Boderc goal

Design methodology: formalisms, techniques, tools and methods

This paper proposes a *model-based* design methodology that consists of:

Formalisms: *Formalisms* are languages / syntax used for system modeling. Formalisms exist for modeling behavior, but also to formalize system requirements. Instances of formalisms are called *Models*. Examples of formalisms are differential equations, (timed and hybrid) automata, finite state machines, temporal logic and queuing formalisms.

Techniques: *Techniques* are used to retrieve information from models or to transform models. Examples of analysis techniques are model checking, performance analysis and program analysis techniques. Examples of transformation techniques are high-level synthesis and software compilation.

Methods: A *Method* ('reasoning framework') provides guidelines and can be seen as a 'recipe book' how and in which order to apply certain *Formalisms, Techniques* and *Tools* in order to solve the design problem at hand. *Methods* are ways to 'capture' design and reasoning knowledge of experienced modelers and designers. A method indicates for instance decomposition in steps (possibly techniques) and an order in which the step should be performed.

Tools: Software *Tools* support the efficient application of *Formalism, Techniques* and *Methods*.

Boderc design methodology

When developing high-tech machines as described in the introduction, two constraints are paramount: project duration and available man power (see figure 1). These constraints must be deeply ingrained in any successful methodology. To meet these constraints, a careful selection has to be made on how to invest design effort and time. The methodology provides two means:

- Focus the in-depth analysis (via modeling) on the most critical issues, preventing "wasting" effort on less relevant problems. For this, one has to identify the *most essential* conflicts and tensions from the design decisions to be made.
- Using simple models that create insight in a design decision within a reasonable time (hours, weeks), instead of detailed models that requires months or even years to develop. The right level of detail must be chosen, which can range from back-of-the-envelope calculations to very detailed models depending on the accuracy of the answer needed. Stepwise refinement of models can be useful for this (see Figure 2).

Even when using models, physical prototypes are essential because of the confrontation with physical reality, where overlooked issues will inevitably pop up. However, it is difficult to quickly evaluate different designs through physical prototypes because a new prototype is needed for each design. Through analysis of models different designs can be evaluated much faster. As a consequence, both models and prototypes are indispensable.

Figure 2 demonstrates another benefit of the methodology. The methodology gives place to formalisms and techniques (which can be seen as 'plug-ins' in the method) and provides a means to evaluate formalisms/techniques on being effective within industrial constraints. Documenting the conditions under which academic formalisms/techniques (state-of-art) and industrial state-of-practice are applicable and effective and their level of accuracy form valuable information. Moreover, gaps can also be identified that require future (academic) research (e.g. extending state-of-practice and 'industrializing' state-of-the-art academic techniques) to obtain the right abstraction level for industrial practice.

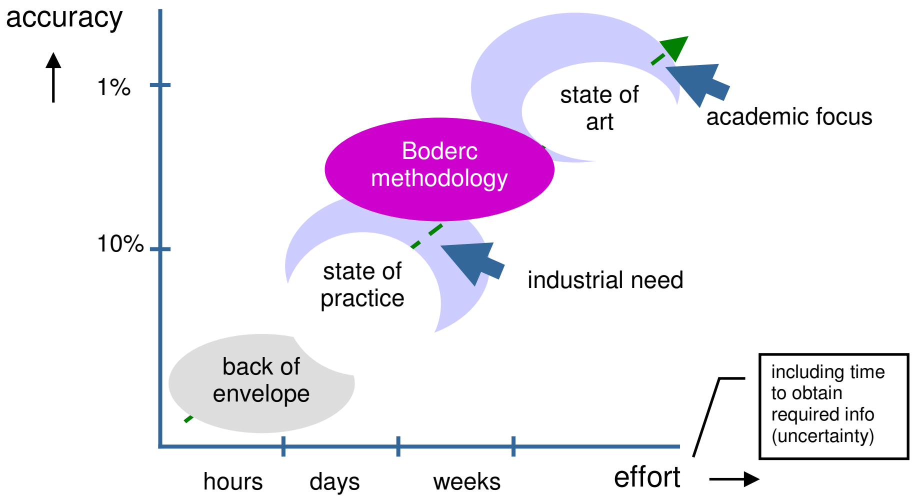


Figure 2: Overview of models at different levels of accuracy (discrepancy between model prediction and reality) and modeling effort required.

Linear stepwise version of the method

The 'method'-part of the methodology is given as the collection of the following steps:

1. **Preparation of the design**
    a. Identify (customer) key drivers and requirements
    b. Identify realization aspects of concern
    c. Consolidate core domain knowledge
2. **Selection of critical design aspects**
    a. Identify tensions and conflicts (qualitative)
    b. Gather facts and identify uncertainties to quantify tensions and conflicts
3. **Evaluation of design aspects**
    a. Build small models (small = hours to weeks of effort)
    b. Perform measurements

These steps are to be used iteratively, so that progressive knowledge can be used. In Figure 10 the iterative nature and the dynamic flow of information between the steps is indicated. Below, the steps and corresponding techniques and formalisms will be explained in more detail. Good visualisation of the outcomes of the steps is important to create insight and overview. The design of a high-volume copier will serve as a means to illustrate the individual steps.

Step 1: Preparation of the design

In step 1, a good understanding of the product to be developed has to be achieved and existing knowledge is gathered to be available for the new design.

Step 1A: Identify (customer) key drivers and requirements

In step 1a, the goal is to identify why a customer (or other stakeholders like the internal business strategist) would want the new product. The main drivers for the stakeholders should be identified and insightfully related to system requirements. This is linked to the product business case. This can be achieved using activities like interviewing marketing experts, interviews and workshops with customers, story telling (Muller, 2004), etc. The results of these activities can then be summarized using a high-level requirements engineering technique. The *key-driver model* has been found to be very useful for this purpose (Muller, 2005).

Example: As part of the Boderc project, a key driver analysis was made of a high-volume copier. The key drivers of the copier were identified and refined into application drivers and finally the system requirements. This analysis is explained further in (Heemels et al, 2006) and a part of the key driver model is shown in the figure below.

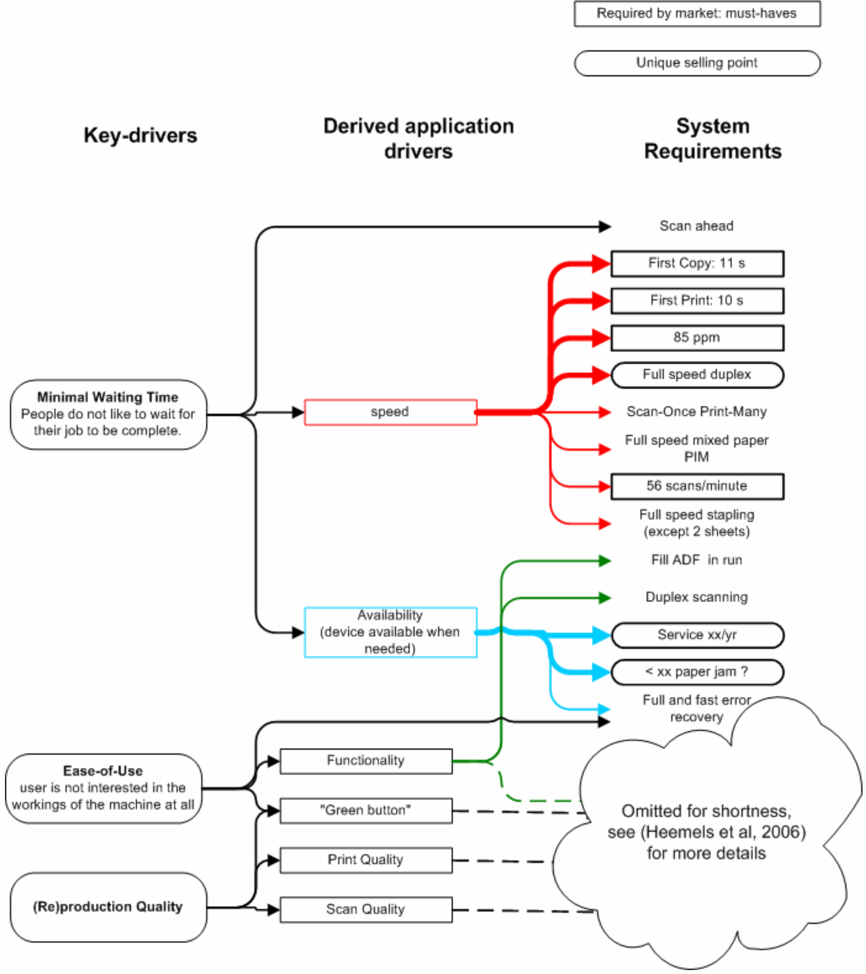


Figure 3: Part of a key driver model for a copier

Step 1b: Identify realization aspects of concern

In step 1b, the goal is to identify which designs aspects of the machine are of concern for its marketing success. In practice, typical the issues or worries that are hot during (coffee or lunch) discussions between the design engineers form good starting points. Some of them might be non-issues caused by non-rational fears, uncertainty, rumours, etc, others might be critical and jeopardizing the success of the product. This has to be found out in steps 2 and 3. In step 1b they are only identified. Currently, there are not many concrete tools and techniques that can be used for this activity; thus this is an interesting area for further research. Methods like 'story telling' (Muller, 2004) and scenario or use case based reasoning (see e.g. (Buhr, 1998)) can be used for this activity. Checklists with problematic issues in previous projects (typically input coming from step 1c) can be used in step 1b. The introduction of new technology or environmental regulations should always be considered with caution.

Example: Based on experience of previous projects and the more stringent power norms nowadays, maximum power usage was an issue in the design of the copier. Also the introduction of stepper motors in the copier is a worry as commonly DC motors were used.

Step 1c: Consolidate core domain knowledge

In step 1c, the goal is to make the most important lessons that were learned during the design of previous machines explicit. In most companies, this knowledge is only known implicitly: it is stored in the minds of key designers. By making this knowledge explicit, a common understanding can be achieved amongst engineers. Capturing the *context* in which a certain design was successful, can be useful to solve similar problems in a same manner in a new machine without much efforts (in figure 10 indicated by 'no-brainers'). It prevents re-inventing the wheel. Going outside the context with a particular solution should be done with caution and would indicate to be part of step 1b. Context is an important factor in design success.

The goals of this step can be achieved by investigating the models, design solutions, methodologies, etc. used in previous designs. Especially designs that were not successful are useful to investigate (see also 1b above). The main question is *why* things were done in a certain way. The results of this investigation must then be summarized, e.g. by identifying design patterns, by writing tutorials and white-papers, determining rules-of-thumb, etc. Of course, part of this information is hopefully consolidated at the end of previous projects, so that this is readily available. Industrial practice often turns out otherwise.

Example: Below are some diagrams showing some core technologies for designing copiers: the main system architecture, the paper-time diagram used for analysis of the timing of print jobs in the paper path, and the main components used in the paper path.

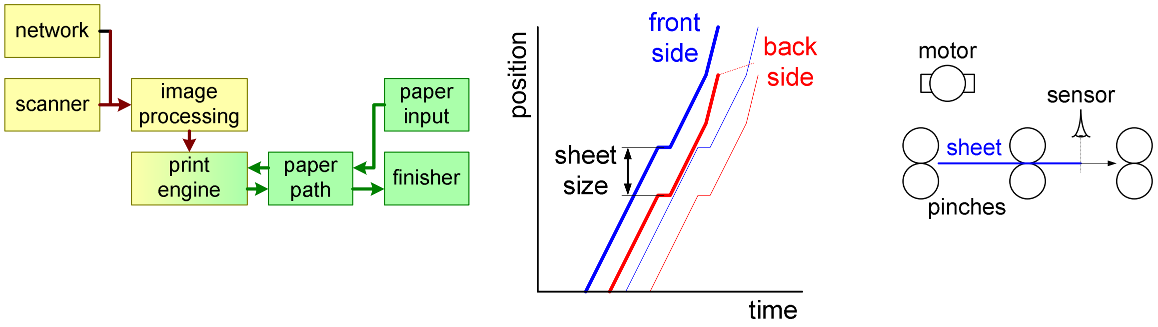


Figure 4: Examples of core domain knowledge for a copier manufacturer

This information is very well known for most experienced copier designers, and these items are always used. However, the familiarity has subtle dangers in that people forget the reason why these technologies are used, what the limitations and advantages are, and when alternatives should be used. Thus there is benefit in formally documenting this knowledge.

Step 2: Selection of critical design aspects

When designing a new product, issues arise constantly. It is imperative to differentiate between important issues, which imply a great risk to the project if not dealt with adequately, and non-issues. Otherwise much time is lost over unimportant issues making development prohibitively expensive. In step 2 the design aspects of concern found in 1b are prioritised by their importance or value for a customer (as analyzed in step 1a), by how challenging the problem is, and how sensitive or vulnerable the overall system is to this challenge.

Step 2a: Identify tensions and conflicts (qualitative)

In step 2a, the goal is to identify qualitatively the design trade-offs and essential tensions that are coupled to a certain design aspect of concern (1b). The fact that a design issue is of concern implies that it must have both benefits and drawbacks (in terms of key drivers and system requirements found in step 1a). Making the tensions and conflicts between benefits and drawbacks explicit allows them to be treated systematically throughout the design process.

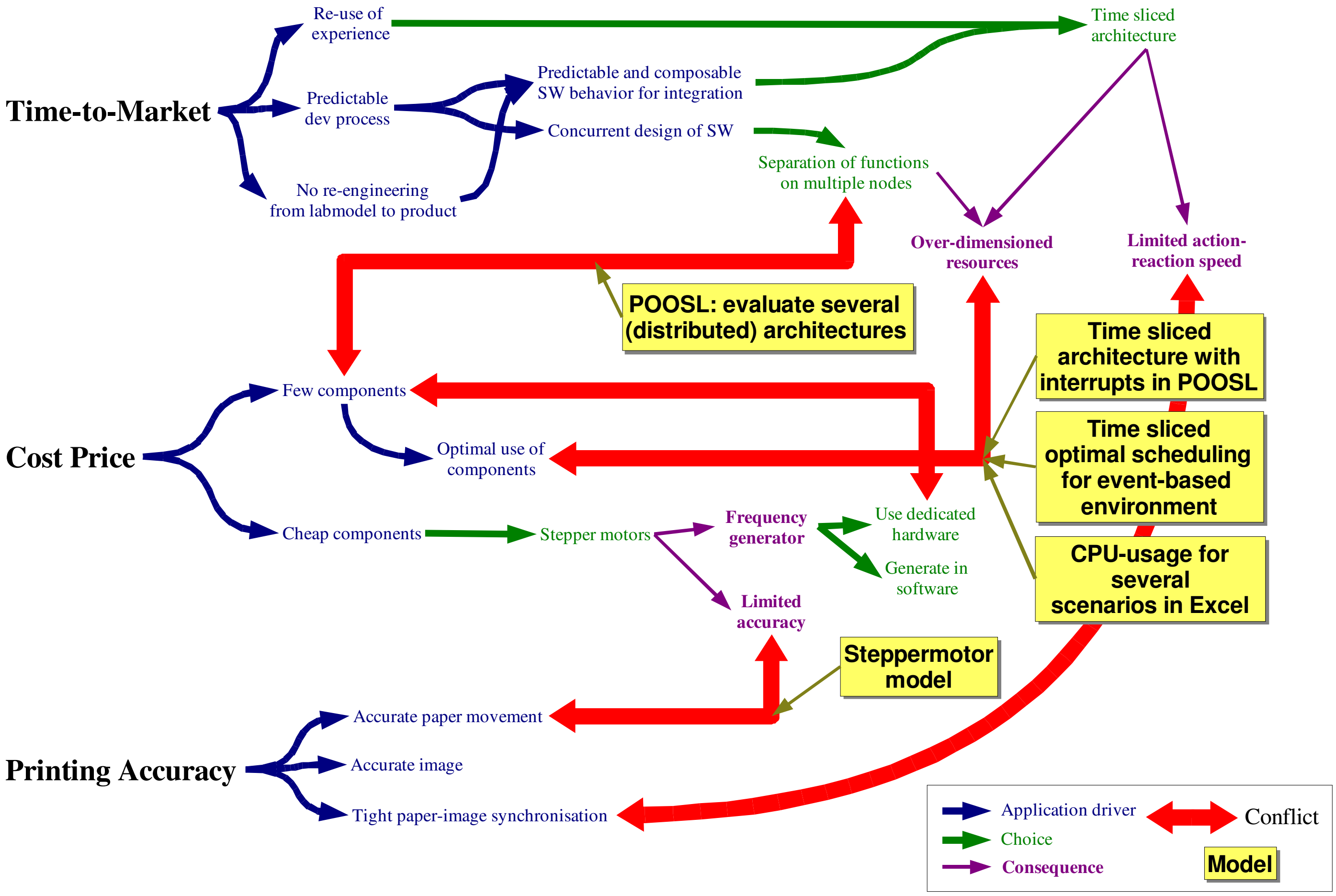


Figure 5: Visualization of threads of reasoning for the control architecture of a copier

A good technique to find these tensions and conflicts is *threads of reasoning*, which investigates where the real tradeoffs are in a design. Concrete design choices are linked to key-drivers and negative side-effects pop-up. In (Sandee et al, 2006) the technique is described and applied for the digital control architecture in a copier. The threads of reasoning diagram for the case study is presented in figure 5. Also the "question generator" (Muller, 2004, Section 9.2.3) supports the exploration of design tensions. Organising workshops and brainstorm sessions is another means. In a workshop, several experts from different disciplines are invited to work together on the main concepts of the machine. They will very quickly find the tensions and conflicts in the design by bringing their own concerns and worries across and connecting them.

Figure 5 is at the heart of the Boderc methodology. For several design choices (e.g. the use of stepper motors instead of DC motors) the relations to the drivers for the copier are displayed. The benefit for the use of steppers is its low cost price, but a drawback is the limited positioning accuracy and thus possible problems for the customer application driver printing accuracy (see figure 3). The conflict between cost price and printing accuracy and several other conflicts are indicated in the figure (see legend for colour use). These require further investigation in step 2b. If the results from 2b are inconclusive, an in-depth study is required: steps 3a and 3b. The rectangles indicate the models that have been used to create more insight in the conflicts

Step 2b: Gather facts and identify uncertainties to quantify tensions and conflicts

In step 2a, tensions and conflicts were identified qualitatively. In step 2b, the goal is to select those tensions and conflicts that require further study. Often, the tensions and conflicts are a result of worries, uncertainty, lack of facts, non-rational fears and turn out to be non-issues. These can often be unmasked by quantifying the issues with rough estimates, using simple facts to weed out the fears, thereby diminishing the worries in the organisation and enabling it to focus on the important issues. However, there will be some issues where there is insufficient knowledge to make an intelligent decision. In those cases, further study is warranted (step 3).

There are many ways to find the facts needed besides using the core domain knowledge of step 1c: an expert can be asked (use the question generator mentioned above), or rough orders of magnitude can be estimated. Also, figures of merit from previous designs can be used. Finally, much knowledge is readily available through existing literature. Facts from all these sources can be used to discard irrelevant conflicts. Note that making quantitative assumption, which all engineers have in their mind, explicit will also reveal the (qualitative) tension. So, step 2b often also precedes step 2a in practice.

Risk assessment (see e.g. Chapter 6 in (Incose, 2004)) is one way to select the tensions that should be addressed more thoroughly as they consider both the impact and the probability of occurrence of a particular issue. Also back-of-the-envelope calculations can be a good starting point as they do not require much effort and time and give first estimates. Iterative refinement to more complicated models as indicated in figure 2 is a good means to progressively analyze a tension. Determining a budget which distributes a resource over different parts of the machine (Freriks et al, 2006) is a formalism that is often used in practice to determine the real magnitude of a problem which is too complex to analyse at the top level. Of course, there is no strict boundary between the current steps 2b and 3a: it is not always clear when to categorize a back-of-the-envelope calculation in step 2 and when to associate a model to step 3a. But in order to keep track of issues, e.g. to allow proper project management, it is helpful to make an *explicit* decision to further study an issue by placing it in step 3a.

Example: In the Boderc project, it was investigated if it were better to use stepper motors or DC motors in a specific copier configuration. For an initial investigation, the thread of reasoning was extended with numerical data on cost price, life time, etc. To assess the consequences of the implementation of steppers for important machine characteristics, a risk assessment matrix model (see (INCOSE, 2004), Chapter 6) was created (figure 6). From this matrix, issues that required in-depth investigation were identified.

	Uncertainties	**Impact**	**Result**
Cost price	**1**	**10**	**10**
Lifetime	**3**	**3**	**9**
Accuracy (reliable)	**10**	**9**	**90**
Ease of design (time)	**7**	**5**	**35**
Noise	**5**	**6**	**30**
Efficiency (power)	**3**	**3**	**9**

Figure 6: Quantified threads of reasoning for the use of stepper motors in a copier

Step 3: Evaluation of design aspects

If the facts in step 2 are not sufficient to make a decision, the issue needs to be evaluated properly. There are two ways to do this: either using a model-based approach or measurements on prototypes. Of course, measurements are always necessary, e.g. to validate models. Depending on the available prototypes, the best way needs to be found to get the answers.

Step 3a: Build small models

In step 3a, the goal is to resolve an open conflict found in step 2 using simple (small) models. As mentioned, models are often very efficient in evaluating design options, as models can be readily modified whereas prototypes are harder to modify. Also models might create a deeper understanding of the relationships in the tension.

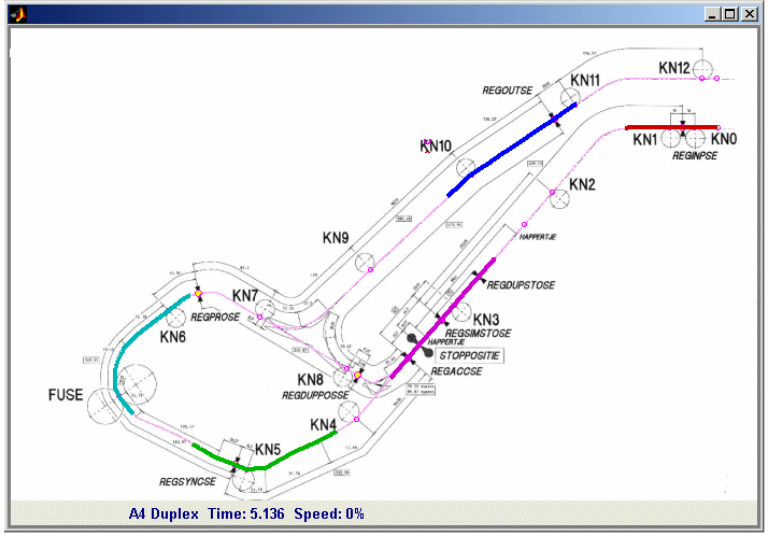
A key issue when using models is which formalism to use to answer the question at hand. Often, model formalisms are suggested in the core domain knowledge gathered in step 1c. If this is not the case, some literature study or research may be required to find the right formalism.

A second key issue is to find the right abstraction level and model boundary to answer the question with the right certainty. The goal is to keep the modelling effort as small as possible.

An interesting question is whether the model is based on theoretical (physical) knowledge (sometimes called first principle or white box modelling), or on empirical facts (regression, identification or black box modelling). Often, a simple model that interpolates measured data can be used for answering questions much quicker than if the model were derived from theory. However, this is case-dependent.

Example: Below are some examples of models used in the Boderc project. Already in Figure 5 some models have been mentioned that were used to analyze specific tensions further. Other modelling formalisms and techniques that have been used include:

- Performance analysis to predict and evaluate the real-time behaviour of the copier control software running on hard-ware platforms (Wandeler et al, 2006) and the analysis of the datapath (the streams of image data that are associated with scan, print and copy jobs).
- Evaluation of real-time embedded systems via coupling of UML tools and Matlab/Simulink to allow simultaneous simulation (Hooman et al, 2004).
- Kinematic models of the sheet flow through the paper path to evaluate copier topology and timing. Strong visualisation and animation complement the models. These are based on 'good weather' conditions: lower level (dynamical) phenomena of motors, slip, jitter and delays in control loops, etc. are not included. See figure 7 for an example of the animations.
- Dynamical models of the complete paper path (including some of the low level phenomena mentioned above) implemented in the simulation environment Matlab/Simulink and Truetime (Van den Bosch et al, 2005). The models focus on timing and power consumption.
- Dynamical models including software execution times of part of the paper path around the fuse, where paper and image meet and accurate synchronisation is needed (Bukkems et al, 2004).
- Power budgets: an example of a visualisation of the power flow through a copier is given in figure 8. See (Freriks et al, 2006) for more details. Thickness of arrows is related to the amount of power flow.



A4 Duplex Time: 5.136 Speed: 0%

Figure 7: Graphical animation for kinematic models of paper transport

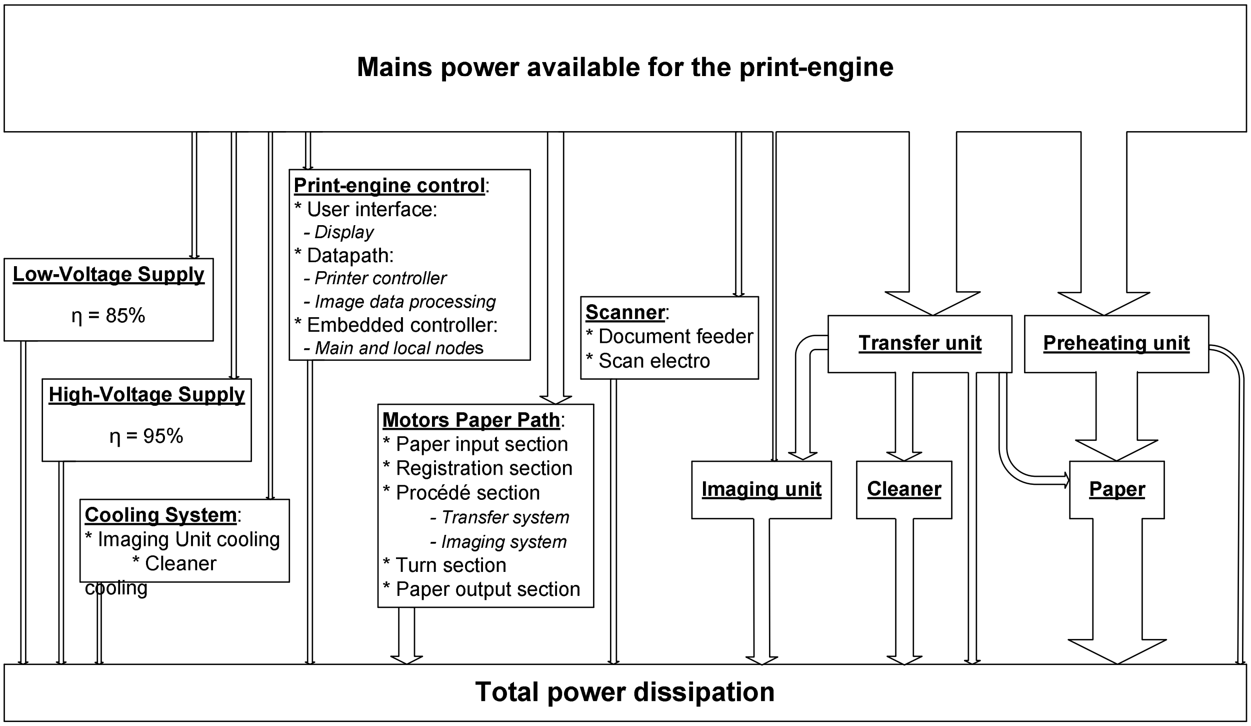


Figure 8: Visualisation of the power flow through a copier

Step 3b: Perform measurements

In step 3b – just as 3a - the goal is to gather the facts with which issues from step 2 can be dismissed. Measurements can play two roles. Either they are used to tune the models to describe practice closely (parameter estimation, identification, model validation) or try to resolve a conflict directly – without a model – by using dedicated experiments. As measurements are from the 'real world', they are usually more authoritative than results from models. However, not every phenomenon can be measured readily, for example because sensors can not be inserted or sensors disturb the phenomenon (think of the Heisenberg principle). It is difficult to determine the effects of parameter variation from measurements. Also, measurements can be faulty. Thus sanity checks are always required.

Often, it is very beneficial to have short iterative loops where measurements and modelling activities follow each other. The measurements show where the models can be improved and the models explain the measurements and show how design choices would influence the results. Models can often capture the relationships between system properties better than a finite number of measurements. Towards the end of a development project more and more the emphasis will shift from modelling (step 3a) towards prototyping and building the actual system (step 3b).

Example: A model was made of the dynamic behaviour of the motors in the paper path (Van den Bosch et al, 2005), as mentioned before. This model was validated with measurements from a real motor in the copier being modelled

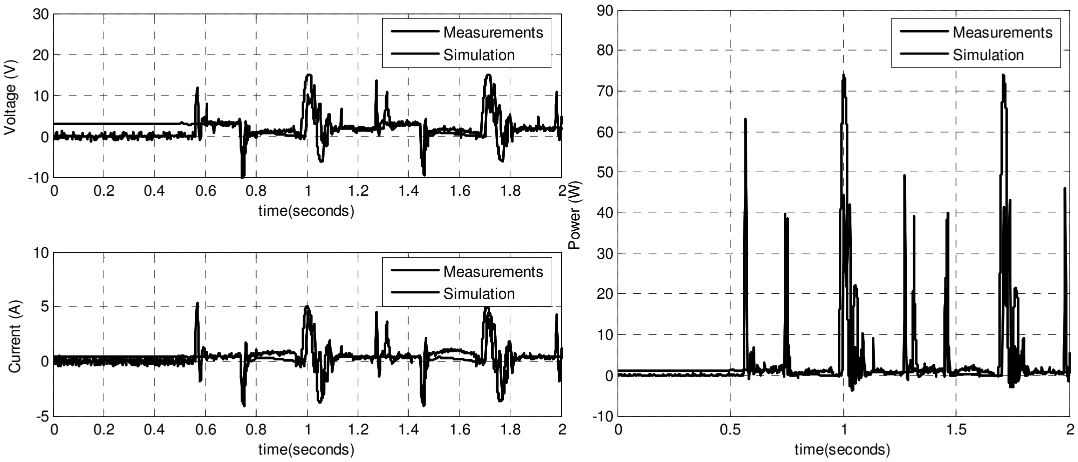


Figure 9: Simulation versus measurements for a single motor in the paper path

Within the Boderc project, also measurements have been performed on hardware platforms to evaluate their real-timing behaviour (e.g. the influence of caching in micro processors).

The method as a structured chart

The nice step-plan shown above is iterative as shown in figure 10 at the end of the paper that contains the same steps, but shows the dynamic flow of information and the making of decisions. For instance, once step 3 has given conclusive answers on a particular issue of concern coming from 1b via step 2, a design decision can be taken. The iteration now proceeds to a next issue of concern. However, also import information obtained during the in-depth study of the previous issues (e.g. data, models, design patterns) should be consolidated in core domain knowledge (1c).

Conclusions

As we are all aware, there is a strong need for multi-disciplinary methodologies that support the system architecting process. In (Muller, 2005b) various reasons are mentioned that hamper the creation of such methodologies. Lack of description and lack of connection to mono-disciplinary techniques are two of them, which we aimed to overcome in this paper. We presented the contours of an emerging design methodology in an explicit manner. The methodology consists of a reasoning framework in the form of a multi-step method, modelling formalisms, analysis techniques and tools. By giving place to modelling and analysis activities, which can be mono-disciplinary, a first step is made in connecting the multi-disciplinary method to mono-disciplinary techniques.

Previous to writing this paper, some steps were taken to validate the methodology. Although various issues remain open, we can already draw the following conclusions:

- The Boderc methodology mimics the way of working of a senior system architect. For instance in (Kostelijk, 2005) the steps of the method can be recognized in the evaluation of an architecture for a DVD hard-disk recorder. As shown by (Kostelijk, 2005), applying the steps of the methodology can prevent system architects from falling prey to ill-founded qualitative reasoning, which can lead to tradeoffs based on incorrect assumptions instead of on quantitative arguments and facts.

- Discussions with junior and senior system architects from Philips revealed that there is a clear recognition of the steps in the method. It matches their way of working and it makes that more explicit. They acknowledged the value of the methodology. Of particular interest for them were the visualisations, e.g. the key driver model in figure 3 and tensions and conflicts in a thread-of-reasoning diagram (figure 5). Documenting design decisions and capturing the main arguments in insightful overviews were considered particularly valuable.
- The application of individual modelling activities (using formalisms and techniques) on particular industrial problems (e.g. paper flow scheduling, stepper motor dynamics analysis, etc) were considered beneficial by the copier manufacturer.

Of course, many issues are still open within this methodology, as in the whole field of multi-disciplinary design. For instance, finding the right level of abstraction for modelling formalisms in an industrial setting is hard. Many academic (mono-disciplinary) formalisms are too complex and many state-of-practice formalisms are too coarse. Finding the right balance between them (see figure 2) is an important issue for future research. Extending the design methodology by further formalisms and tools (especially selecting design aspects of concern in 1b and selecting critical design issues in step 2b) is also open. Hence, by making an attempt to be explicit, this paper hopefully initiates many discussions, allows further validation of the design methodology and advances future SA research.

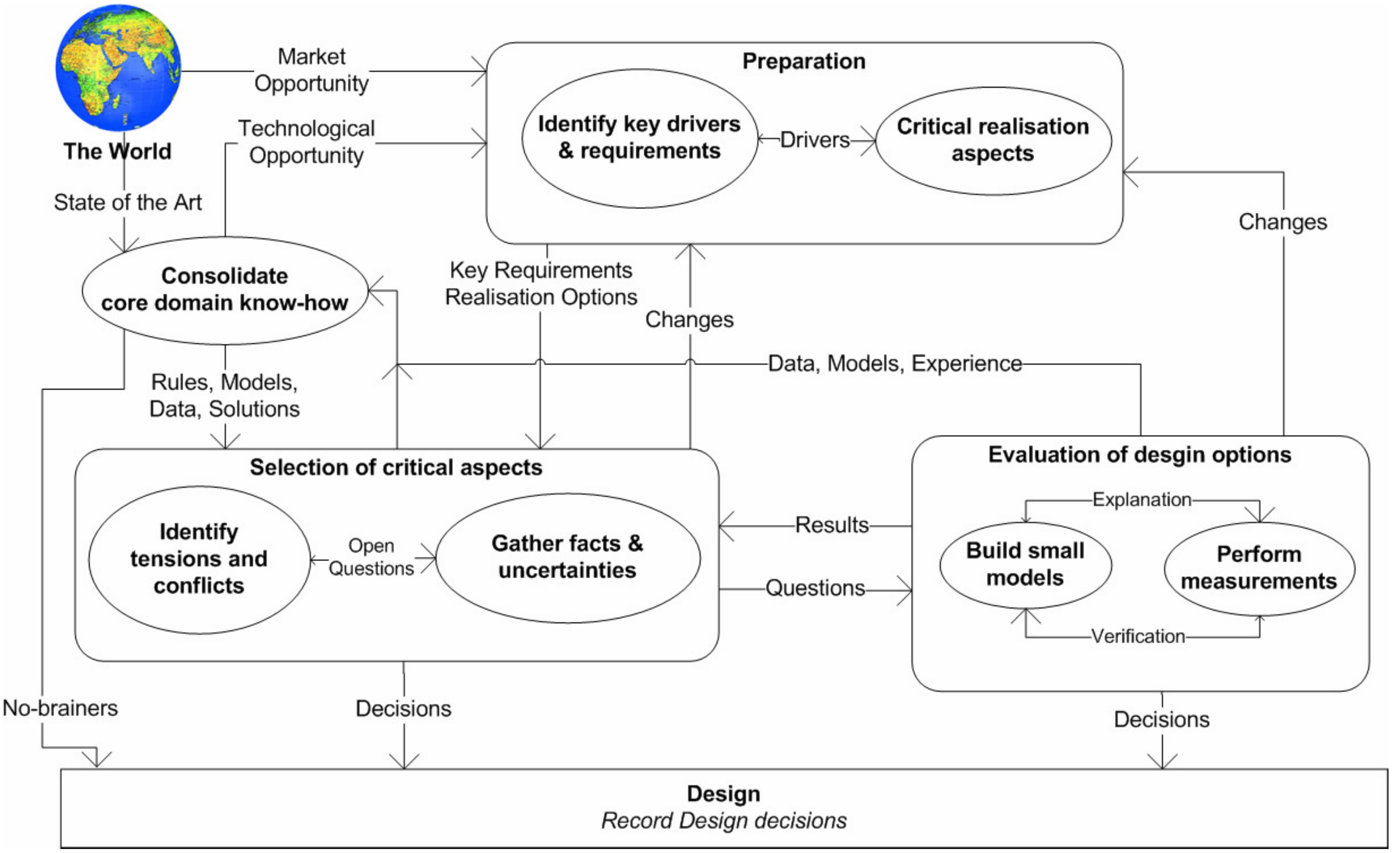


Figure 10: Dynamic flow of information in the method

References

Buhr, R.J.A., "Use Case Maps as Architectural Entities for Complex Systems," *IEEE Transactions on Software Engineering*, Volume 24, Issue 12, pp 1131 - 1155.

Bukkems, B., Sandee, H., Beckers, J., Yuan, Z., Van der Wijst, B., Van de Molengraft, R, "A Case-Study in Multidisciplinary Modeling of Dynamic Embedded Systems", *IEEE Conference on Mechatronics and Robotics 2004*, Aachen, Germany.

Freriks, H.J., Heemels, W.P.M.H., Muller, G., "On the systematic use of budget-based design", submitted, 2006.

Heemels, W.P.M.H., Somers, L., Van den Bosch, P., Muller, G., Yuan, Z., Van der Wijst, B. Van den Brand, A., "The use of the key driver technique in the design of copiers", submitted, 2006.

Hooman, J., Mulyar, N, Posta, L., "Coupling Simulink and UML tools," *Proc. Symposium FORMS/FORMATS 2004*, pp. 304-311, 2004.

International Council on Systems Engineering (INCOSE) Technical board, "System Engineering Handbook", 2004.

Kostelijk, T., "Misleading architecting tradeoffs," *IEEE Computer*, p. 20-26, May 2005.

M.W. Maier, E. Rechtin: The Art of Systems Architecting, second edition, CRC Press, Boca Raton, 2002.

Muller, G., CAFCR: A Multi-view Method for Embedded Systems Architecting. Ph.D. Thesis, http://www.gaudisite.nl/ThesisBook.pdf, 2004.

Muller, G.J., "Industry and Academia: Why Practitioners and Researchers are Disconnected," *Proc. INCOSE Symposium 2005a*, Rochester, NY, USA.

Muller, G.J., "Do useful Multi-Domain Methods exist?," *Conference System Engineering Research (CSER) 2005b*, Hobroken, USA.

Potts, C., "Software-engineering research revisited," *IEEE Software*, Vol. 10, No.5:19–28, 1993.

Sandee, J.H., Heemels, W.P.M.H., Muller, G., Van den Bosch, P.F.A., Verhoef, M.H.G., "Threads of Reasoning: A Case Study in Printer Control," submitted. 2006.

Van den Bosch, P.F.A., Van de Waal, E.H., "A case study of multi-disciplinary modelling using Matlab/Simulink and Truetime", *Proc. INCOSE Symposium 2005*, Rochester, NY, USA.

Wandeler, E., Thiele, L., Verhoef, M., Lieverse, P., "System architecture evaluation using modular performance analysis – a case study", *Intern. journal on software tools technology transfer,* 2006.

Biography

Maurice Heemels received the M.Sc. degree (with honours) in Mathematics and the Ph.D. degree (cum laude) from the dept. of Electrical Engineering (EE) of the TU/e (Technische Universiteit Eindhoven) in 1995 and 1999, respectively. From 2000 until 2004 he worked as assistant professor in the control systems group (EE) of the TU/e. In 2001 he spent three months as a visiting professor at the ETH in Zurich, Switzerland. Since 2004 Dr. Heemels is working at ESI (Embedded Systems Institute) as research fellow. In 2004 he worked at Océ R&D for three months. His research interests include modeling, analysis and control of hybrid and non-smooth dynamical systems and their application to multi-disciplinary design for embedded systems.

Evert van de Waal graduated in 1994 from Twente University, the Netherlands, with an M.Sc. in Electrical Engineering. After performing research at the Industrial Control Centre, University of Strathclyde, Glasgow, he entered industry as a developer of embedded software. Since 2001 he has been working for Imtech ICT as a technical consultant with a focus on developing high-tech machines. Since 2003, he has been involved part-time in the Boderc project.

Gerrit Muller received his M.Sc. 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 System 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 to work on System Architecture methods and to enable education of new System Architects. Special areas of interest are ways to cope with the exponential growth of size and complexity of systems, and the human aspects of systems architecting.