RISK ADJUSTED, CONCURRENT DEVELOPMENT OF MICROSYSTEMS AND RECONFIGURABLE MANUFACTURING SYSTEMS

By
Erik Puik

A thesis submitted in partial fulfilment of the requirements for the degree of
Doctor of Philosophy in Engineering

Doctoral Committee:
Professor Nam Pyo Suh
Professor Tony McNally
Professor Richard McMahon

Supervisor:
Professor Darek Ceglarek

Warwick Manufacturing Group
University of Warwick
Supported by the HU University of Applied Sciences Utrecht
November 2017

ABSTRACT

Controlling uncertainties is a challenging aspect in design and manufacturing of microsystems. As microsystems are characterised by features in the micro domain, product development and manufacturing processes are applied at the boundaries of their operational areas. In combination with many disciplines (mechanical, electrical, software, chemical etc.) and little standardisation, it causes microsystems development to be more time and cost intensive than products in the macro domain. Development of microsystems benefits from a concurrent approach of product and production design.

Uncertainties may be addressed by application of methods for systems engineering (engineering design). Systems engineering applies models for the analysis of projects, usually a linear set of gates that need to be closed successively as the project evolves. Over the last ten years, models with an iterative approach of design and testing, gained in popularity due to their more agile characteristic that performs better in fast changing markets. Microsystems development benefits from the linear approach that performs well for their structured project control, but because of the high market dynamics, agile methods will speed up the process, which results in faster market introduction, advances the product life cycle, and increases return on investments.

Currently, there are no known systems engineering models that combine linear and iterative monitoring of projects to gain the best of both methods, especially not in combination with the capability of concurrently monitoring the development of product and production design. This thesis investigates how existing ways of system engineering can be combined to: (RQ1) enable iterative and linear modelling of microsystems development, and (RQ2) merge these qualities into a combined model to monitor the development process concurrently. The first problem is addressed by (RQ1):

i. Modelling development progression by execution of iterative cycles that alternately perform functional system decomposition and functional gating;
ii. This iterative model is elevated with the method of Axiomatic Design to enable concurrent system decomposition. Implementation of elements from the V-Modell XT enable functional gating to index the concurrent development process;

iii. The 'Theory of Complexity' of Axiomatic Design is applied to realise an intelligent, knowledge based, gating function to be used as a continuous maturity measure;

The results show that linear and iterative models can be merged successfully. With some extensions, the Theory of Complexity of Axiomatic Design can indeed be used for continuous monitoring of product and process development. The thus-obtained maturity measure can be applied for the analysis of project decisions. This was successfully done for retrospective analysis of two cases.

To merge the qualities of analyses 'i to iii' into a combined model to monitor the development process concurrently, three tools for application have been developed (RQ2):

iv. The first is a method for visualisation of the intelligent gating function, based on analysis 'iii'. The method applies a newly developed 'Maturity Diagram' that plots the Design Axioms as continuous parameters;

v. The second is a method for assessment of reconfigurable manufacturing systems based on analysis 'ii'. The method estimates the investigations needed to (re)configure a product specific manufacturing system;

vi. The third is a tool for roadmapping and monitoring that combines outcomes of analyses 'i, ii, and iii'. This model is called 'Constituent Roadmap' and it is based on: (a) an iterative approach, (b) concurrent decomposition, (c) the advanced gating function, and (d) knowledge application to the product and process design.

The Constituent Roadmap was applied for the development of a 'smart dust' sensor system. It was found to structure knowledge development and application. This increases the chances to satisfy the functional requirements of the design. In parallel, it functions as a communications tool between designers and managers.

Together, a reasonably complete picture has emerged how the design of microsystems and their production means can be modelled, and how uncertainties may be categorised so they can be addressed in the best order.

THESIS HIGHLIGHTS

H.1 Motivation of the Research

- Microsystems are products with features in the micrometre range, comprising different technologies, like mechanics, electronics, software, optics, chemistry, or biomedical technology. Micro systems are usually applied as sensors in cars, cell-phones, and many household appliances or as actuators in beamers, switches, or photo cameras;
- Microsystems are difficult to design and to manufacture. Small tolerances are needed to realise the small feature sizes, and production technologies are required to meet these tolerances. Therefore, product design and manufacturing are developed concurrently;
- Time to market is an issue when developing microsystems. When delayed, customers may switch to competing resources. This not only leads to delayed break-even point for investments, but also to less market penetration and substantial loss of turnover;
- To organise the product development- and industrialisation-processes, industry applies ‘system engineering’ models to model the development process. As such, product design and manufacturing are monitored in a concurrent fashion;
- The system engineering models that are currently available, find their origin in the development of software systems. These models are not optimised for the microsystems and their manufacturing means; especially in the early phase of product design, reliable models for the development of microsystems are sparsely available;
- In the development process of system engineering models for product development, there are generally two streams of thought: (i) so called ‘Linear’ models that describe and follow the development process as a sequence of stages to complete, and (ii) ‘Iterative’ models, that stimulate development in cycles to divide the development process in many small stages;
- Microsystem development benefits of both models, as well the linear models to follow the project on a macro scale, as the iterative models that provide quick

feedback with the ability to regularly synchronise the product and manufacturing processes;

- In this thesis, the combined models are developed, executed, and evaluated. This is done by defining the different dynamical characteristics of the microsystem development process, as well for iterative as linear models.

H.2 Part 1: Modelling

- The basic model that is applied in Chapter 3 of this thesis is called the ‘Micro System Development’ (μSD) Framework. This framework focuses on two important functionalities of the development process: (i) the process of ‘Functional Decomposition’ to break-down difficult problems into small pieces of technology that each can be overseen, and (ii) the process of ‘Functional Gating’ to control and structure the development process on a macro scale by successively closing a number of gates that indicate development stadia of the project;
- The second step that will be explained in Chapter 4, is to upgrade the μSD Framework with improved capability for decomposition and gating, to optimally support the concurrent development process of microsystems. This model is called the ‘Concurrent Micro System Development’ (CμSD) Framework. This upgraded framework applies the methodology of Axiomatic Design to implement full concurrency;
- The third step is explained in Chapter 5 and continues in expanding the μSD Framework, however in this chapter, the gating function is implemented differently. For the μSD and the CμSD Framework, the gating functions are based on the extent to which decomposition is completed. In the model of chapter 5 this is implemented with a process that is called ‘Intelligent Gating’. Intelligent gating applies a measure of ‘Information in Design’ according to the method of Shannon as was developed for the communication theory.

H.3 Part 2: Applications of Risk Based, Concurrent Systems Engineering Models for Microsystem Development

- In addition to these three models, the thesis also describes three applications that show how the models may be applied in practice. Chapter 6 explains how the progression in a microsystem development project may be plotted in a newly developed diagram; the 'Axiomatic Maturity Diagram'. This diagram can not only be used to monitor the progression in the project, but also to recover from errors made in the project since it has the capability to analyse and visually reveal the project-errors;
- Chapter 7 also applies the intelligent way of gating; however, the implementation is different. It presents a way to determine lead-time for development and remaining work that needs to be done to complete the project. The activities are weighted by an analysis based on axiomatic design as was developed for the CμSD framework in Chapter 4;
- Finally, a combination of the system engineering models that were developed in this thesis are combined to an overall method for the development of microsystems in Chapter 8. The method is called 'Constituent Roadmap of Product Design' and it gathers the μSD framework, intelligent gating, and a new measure called the 'Check Matrix' that can be used to track the status of the development process.

H.4 Research Results

- A number of six case studies are executed throughout the thesis with the models that were developed. These cases showed the possibility to execute the development of microsystems iteratively, concurrently, and functionally gated. In this way, a combination is obtained between agile feedback on the development process, concurrent approach of product design and manufacturing means, and rigorous way follow the development process in a sequence of stages;
- This thesis delivers a number of academic contributions. The μSD combines the capability of quick feedback from iterative methods and the rigidity of linear methods. The CμSD adds use-cases for 'Iterative Concurrent Development', and

the µSD with intelligent gating offers a totally new approach to Functional Gating. New and effective decomposition of 'Information in design' provides three kinds of information with typical characteristics that need all three need to be approached in a specific way and that can be visualised in the new axiomatic maturity diagram that also visually reveals errors in the project. Finally, the constituent roadmap of product design implements the ability to transparently track the knowledge of the designer using the check matrix. Combined, microsystem development may benefit from these models.

ACKNOWLEDGEMENTS

"… Jij doet aan breinpluk, maar dit is de tijd van het ipsen en daarom verpop ik mijn denksels…" (Toonder, 1972).

I would like to think this thesis is a product of my mental faculties, however, it is not. Obviously as the declaration on the next page states, I wrote the thesis myself, but that's about it. In general, humans are forged by their environment, and it makes this thesis a gathering of what came on my path, the people in my environment, and the literature I happened to find. The good part is that colleagues, family, and others in our environment are of influence too, and by being a teacher myself, I pay it forward in my attempts to help my students; if I try hard, and with a little luck, I may succeed and make it stick. While finalising this thesis, I would like to thank others for their assistance; they forwarded their knowledge, had a little luck and made it stick with me. Though by no means perfect, I consider this thesis my best accomplishment so far.

I would like to thank my advisor professor Darek Ceglarek for spending so much time helping me when I was visiting Warwick University. Also, I really would like to thank my examiner professor Nam Suh, who came all the way from Boston for the viva, and professors Tony McNally and Richard McMahon for their role in this process as well.

Many thanks to my co-authors too. I am not going to name them all. Thanks for helping me, sharing your thoughts and contributions to the papers. Your legacy is not only in the papers, but also in the thesis.

Another important role was played by many colleagues of the HU University of Applied Sciences to instigate and support me in this process. Finishing this PHD study is a long-cherished wish of me and their unintentional pressure on me to perform well was just what I needed, to start the whole process as well as to finish it! I am grateful to HU as organisation to support me financially and make it all possible.

Finally, the person I would like to thank most is my wife Lindy, she brought many sacrifices and on top of that, she had to listen to me complaining when things went bad, when papers were not accepted and needed to be rewritten. She hardly complained, even when we skipped our holidays more than once. Without that support, I don't think we would have succeeded.

DECLARATION

This thesis is presented in accordance with the regulations for the degree of Doctor of Philosophy. It has been written by myself and has not been submitted anywhere else. The work in this thesis has been undertaken by me except where otherwise stated.

TABLE OF CONTENTS

LIST OF TABLES

TABLE OF FIGURES

LIST OF PUBLICATIONS

Journal Papers

1. Puik, E. C. N., Telgen, D., Moergestel, L., van, & Ceglarek, D. (2016). Assessment of reconfiguration schemes for Reconfigurable Manufacturing Systems based on resources and lead time. *Robotics and Computer-Integrated Manufacturing*. http://doi.org/10.1016/j.rcim.2015.12.011

Journal Papers (draft)

1. A Design Progression Rationale for Development of Products and their Reconfigurable Manufacturing Systems, to be submitted to International Journal of Project Management
2. A Method for Indexing Product/Production Development Progression, Applied on Reconfigurable Manufacturing Systems, to be submitted to the Journal of Manufacturing Technology Management
3. The IV Model: An Iterative, Risk Based Development Method, Applied to Reconfigurable Manufacturing Systems, to be submitted to the Journal of Systems Architecture

Book Chapters

1. Puik, E. C. N., & Ceglarek, D. (2016). A Different Consideration on Information and Complexity in Axiomatic Design. In N. P. Suh & A. M. Farid (Eds.), *Axiomatic Design in Large Systems: Complex Products, Buildings Manufacturing Systems* (1st ed.). http://doi.org/10.1007/978-3-319-32388-6

Conference Papers

1. Gielen, P., Sillen, R., & Puik, E. C. N. (2012). Low cost environmentally friendly ultrasonic embossed electronic circuit board (pp. 1–7). Presented at the 2012 4th Electronic System-Integration Technology Conference, IEEE. doi.org/10.1109/estc.2012.6542058
2. Puik, E. C. N., Smulders, E., Gerritsen, J., Huijgevoort, B. V., & Ceglarek, D. (2013). A Method for Indexing Axiomatic Independence Applied to Reconfigurable Manufacturing Systems. In M. K. Thompson (Ed.), (1st ed., Vol. 7, pp. 186–194). Presented at the 7th International Conference on Axiomatic Design, Worcester. www.axiomaticdesign.com/technology/icad/icad2013/27-Puik-et-al-eProceedings.pdf

3. Puik, E. C. N., Telgen, D., Moergestel, L. van, & Ceglarek, D. (2013). Structured Analysis of Reconfigurable Manufacturing Systems. 23rd International Conference on Flexible Automation and Intelligent Manufacturing (FAIM). doi.org/10.1007/978-3-319-00557-7_12
4. Puik, E. C. N., Telgen, D., Ceglarek, D., & Moergestel, L., van. (2013). Qualitative product/process modelling for reconfigurable manufacturing systems. *2013 IEEE International Symposium on Assembly and Manufacturing (ISAM)* (pp. 214–218). IEEE. doi.org/10.1109/ISAM.2013.6643530
5. Puik, E. C. N., Gielen, P., Telgen, D., Moergestel, L., van, & Ceglarek, D. (2014). A Generic Systems Engineering Method for Concurrent Development of Products and Manufacturing Equipment (Vol. 435, pp. 139–146). Berlin, Heidelberg: Springer Berlin Heidelberg. doi.org/10.1007/978-3-662-45586-9_18
6. Puik, E. C. N., & Ceglarek, D. (2014). A Review on Information in Design, Presented at the 8th International Conference on Axiomatic Design, Lisboa. www.axiomaticdesign.com/technology/icad/icad2014/9-Puik-et-al-paper.pdf
7. Puik, E. C. N., & Ceglarek, D. (2014). A Theory of Maturity, Presented at the 8th International Conference on Axiomatic Design, Lisboa. www.axiomaticdesign.com/technology/icad/icad2014/17-Puik-et-al-paper.pdf
8. Puik, E. C. N., Telgen, D., Moergestel, L., van, & Ceglarek, D. (2014). Classification of reconfiguration resources and lead time for reconfigurable manufacturing systems (pp. 857–862). Presented at the International Conference on Flexible Automation and Intelligent Manufacturing FAIM, DEStech Publications, Inc. doi.org/10.14809/faim.2014.0857
9. Puik, E. C. N., & Ceglarek, D. (2015). The Quality of a Design Will Not Exceed the Knowledge of its Designer; an Analysis Based on Axiomatic Information and the Cynefin Framework (Vol. 34, pp. 19–24). Presented at the 9th International Conference on Axiomatic Design ICAD2015. doi.org/10.1016/j.procir.2015.07.040
10. Puik, E. C. N., & Ceglarek, D. (2015). Axiomatic Product Design in Three Stages: A Constituent Roadmap That Visualises the Status of the Design Process by Tracking the Knowledge of the Designer (pp. V015T19A003–V015T19A003). Presented at the ASME 2015 International Mechanical Engineering Congress and Exposition, ASME. doi.org/10.1115/IMECE2015-50993
11. Puik, E. C. N., Foley, J. T., & Ceglarek, D. (2016). Ignorance is Bliss: sudden appearance of previously unrecognized information and its effect. In A. Liu (Ed.), (p. 8). Presented at the 10th International Conference on Axiomatic Design ICAD2016, Xi'an, (Puik et al., 2016)
12. Foley, J. T., Puik, E. C. N., & Cochran, D. S. (2016). Desirable Complexity. In A. Liu (Ed.), (p. 6). Presented at the 10th International Conference on Axiomatic Design

ICAD2016, Xi'an, (Foley et al., 2016)

13. Puik, E. C. N., van Duijn J., Ceglarek D.J. (2017). Guidelines for Application of the Constituent Roadmap of Product Design Based on Axiomatic Design. (p. 6). Presented at the 11th International Conference on Axiomatic Design ICAD2017, Iaci, (Puik et al., 2017b)
14. Foley, J. T., Puik, E. C. N., & Cochran, D. S. (2017). The Faucet Reloaded: Improving Axiomatic Design by Example (p. 6). Presented at the 11th International Conference on Axiomatic Design ICAD2017, Iaci, (Foley et al., 2017)

ABBREVIATIONS

AD:	Axiomatic Design
AMS:	Adaptable Manufacturing Systems
CA:	Customer Attributes
DMS:	Dedicated Manufacturing Systems
DP:	Design Parameter
FMEA:	Failure Modes and Effect Analysis
FMS:	Flexible Manufacturing Systems
FR:	Functional Requirement
MG:	Maturity Grid
PRINCE(2):	PRojects IN Controlled Environments (2nd generation)
PV:	Process Variable
RMS:	Reconfigurable Manufacturing Systems
QA:	Qualitative Analysis
QFD:	Quality Function Deployment
QMAP:	Qualitative Modelling and Analysis of Processes
SADT:	Structured Analysis Design Technique
TRIZ:	Teoriya Resheniya Izobretatelskikh Zadatch (Russian) or the 'theory of inventive problem solving'
TRL	Technology Readiness Level
V&V	Verification and Validation

CHAPTER 1

INTRODUCTION

1.1 Scope of the Thesis

Microsystems are devices with feature sizes in the micron range. Starting in the mid-eighties, elements of mechanical systems were realised on the surface of semiconductor wafers, using the equipment that was developed for semiconductor manufacturing. These electromechanical systems brought the mechanical discipline in a new realm with various names; MEMS in the US (Micro Electro-Mechanical Systems), Micromachines in Japan, and Micro Systems Technology in Europe (Heinig et al., 2014). Microsystems connect multiple disciplines e.g.: mechanical, electrical, software, optical, fluidic, chemical, and/or biomedical. With combined elements of semiconductors, watchmaking, printed circuit board assembly, and the fact that it transcends multiple disciplines, design and assembly of microsystems has a heterogeneous character. This heterogeneous integration is submitted to the basic nature of general production principles, but unfortunately it does not profit from a high level of standardisation (Onori, 2009). The result is that development of microsystems is considerably slower and more cost intensive than products on macro scale. Due to the general level of difficulty of micro fabrication & assembly, caused by high tolerances in the design, it introduces substantial risks for development and manufacturing (Van Brussel et al., 2000; Zhang et al., 2006b; Voelkel et al., 2012). To adjust the manufacturing means as much as possible to the tolerance hungry product design and vice versa, the development of products and manufacturing means is preferably executed in a concurrent way (Dimov et al., 2012;

Sohlenius, 1992). This thesis focusses on risk adjustment during the concurrent product design of microsystems and their production means, formulated as follows:

RQ1:

How do project uncertainties, during the development process of microsystems and their production means, evolve as the project progresses?

RQ2:

How can this knowledge be applied in a protocol to guide engineers effectively and concurrently through the development process of microsystems and their production means?

1.2 Motivation for the Research

Successful launch of new and unique products will lead to competitive advantage which is at the heart of a firm's performance in competitive markets (Porter, 1985). However, prior to success, development of new products and their industrialisation means requires extensive investments and comes with undesirable risks. A failing product development process can bring a company to the brink of collapse. With such high stakes, risk mitigation is an important instrument during development of new products.

A second industrial motivation for this research are the development dynamics in modern, quickly eroding markets. Due to global competition, the purchasing power of customers is increasing and puts pressure on lead times for product design and production engineering. High production efficiency and rapid response to changing customer demand are dominant conditions for enterprises to stay successful (Koren, 2006). The market for microsystems may be considered a quickly eroding market and requires tight scheduling of system development; being 'first' leads to better market penetration, increased margins on products, and as a result, to progressively higher return on investments. If market introduction is delayed, remaining economic lifecycle is significantly reduced (Puik et al., 2002; Puik & Moergestel, 2010; Ceglarek, 2014). Figure 1.1 shows losses of late market introduction (Puik, 2009).

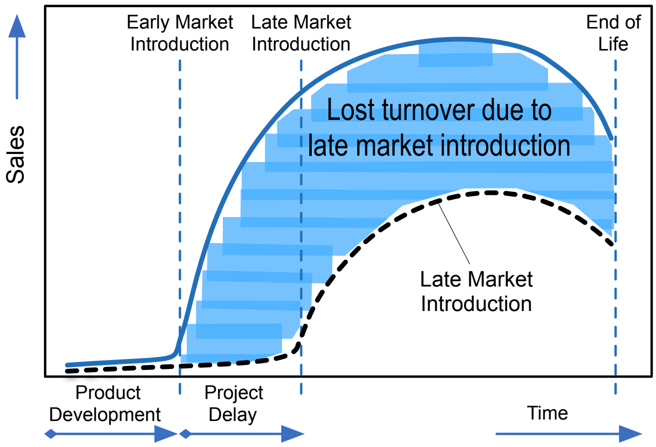


Figure 1.1 *Delayed market introduction reduces sales and duration of sales of products; total turnover will be substantially lower*

High pressure on lead times also has led to adjustments in production processes, production approach, and applied equipment. Manufacturing has become 'agile'; manufacturing equipment and production locations have become modular and subject to evolve frequently and on short notice. This is the venue of 'Reconfigurable Manufacturing Systems' (RMS) (Gunasekaran, 2001; Puik & Moergestel, 2010; Koren, 2006). RMS are a logical addition to 'Dedicated Manufacturing Systems' (DMS), and 'Flexible Manufacturing Systems' (FMS). DMS are most traditional; they are applied for a long period of manufacturing without significant changes, even up to 30 years. FMS are computer numerically controlled systems. In FMS, the application of computerised control systems enables fast adaptions to a range of variations in production. The structure of the machine, however, was determined by the 'mechanical system design' and is not able to change (Koren et al., 1999). RMS fill the gap by adding a modular architecture in both mechanical design and control system. The architecture enables agile change of machine structure by adding and removing parts of the system, and by changing the corresponding software programming (Strasser et al., 2005; Telgen et al., 2015; Moergestel et al., 2011; 2013).

A third motivation is reported in an outlook study on the future of European assembly automation for microsystems. The European EUPASS project (Evolvable Ultra-Precision Assembly Systems) makes note of a difficult relation between ‘Product Development’ and ‘Manufacturing Engineering’ (Onori, 2009). This is primarily due to the fact that processes, being catered for application in the assembly systems, are insufficiently documented and structured. A typical approach for these markets is the application of ‘Design for Assembly’ that advocates the designers to incorporate knowledge of the assembly processes, to optimise their design, and as such enable an effective transition from the product design stage to production. Onori attributes this problem to communication flaws between the designers that apply the processes, and vendors/developers of process technology. The fact that these are different people at different locations introduces a non-conformity during the process of industrialisation.

Fourthly, development of new products freezes a companies’ resources. Investments in new products and their manufacturing equipment precede potential earnings of the company. Invested resources are frozen till after market introduction when they are gradually regained; it rigidifies the company, especially in a fast-changing market. This effect opts for: (i) quicker market introduction, (ii) more efficient product development, and (iii) effective manufacturing engineering followed by pilot production and ramp-up to required production numbers (Moore, 2009).

Lastly, the areas that were selected for study were encountered as problems when the author was working for the high-tech start-up MA3Solutions that produced equipment for microsystem manufacturing. Due to lack of methods to understand and control projects, the company did not succeed in managing risks adequately.

1.3 Research Objectives and Positioning of the Thesis

This thesis focusses on the development and application of methods for ‘Systems Engineering’ (the term ‘Engineering Design’ is also applied in the US). Systems engineering is defined by Ramo as ‘the design of the whole as distinguished from the design of the parts’ (Booton & Ramo, 1984). The tools of the systems engineer are the human brain, a computer, and numerous techniques for analysis (mathematical or qualitative), to model concerned phenomena and their characteristics in detail. Every modern designer is in some extent a systems engineer and applies models to explain observed phenomena. In this thesis, systems engineering models are applied, developed, and improved with the goal to adjust development risks that may occur when developing and industrialising microsystems in a modern environment.

1.3.1 Modelling Design Progression

Problem: In an ideal situation, a systems engineering model, to determine progression of a microsystem-design during its development stages, would have the following characteristics:

- The model quantifies remaining project risks integrally, with an option to investigate the source of the risk;
- The model is able to follow the development process from the early explorative design stage to market introduction;

- It has the capability to concurrently model product development and development of production means;
- It supports an agile approach by applying short iterative cycles.

Current Situation: The impact of making decisions in the early life of product/production development is larger than the impact at the end of the development process. Figure 1.2 shows how steeply the impact declines as a product matures.

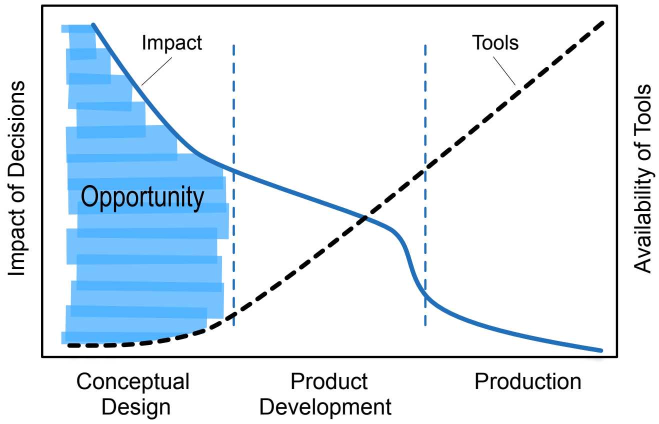


Figure 1.2 *The availability of tools is low, just as it could lead to maximum impact*

Conversely, while there are many modelling tools to help engineers make good decisions about products late in the development process, there are few available early in the process, where they are needed the most (IMTR-Project-Team, 2000; Hsu & Liu, 2000). The number of models that are applicable over the whole development span is limited. Nevertheless, some are universal in this sense, like the widely-applied Waterfall-Model and the V-Model (and derivatives). These ‘Linear’ models apply a sequence of ‘Gates’ that may be closed when all related targets for that gate are met. When properly applied, a gate that is closed will not be reopened again. For software development projects, which are typically characterised by an agile nature, project modelling tools are usually ‘Iterative’. Iterative methods execute improvement cycles that are recursively addressing problems in the design. These methods allow swift reactions in unexpected situations, but do not benefit from the rigour of a solid structure.

Table 1.1 *Positioning the investigations for linear and iterative planning methods in relation to the current research status*

Issues Considered	***Addressed By***	***Related Work***	
Methods for project planning and risk adjusted modelling of design progression	Linear planning based on Waterfall-Model and derivatives (PRINCE, PRINCE2)	(Royce, 1970), (Boehm et al., 1976), (Daft & Lengel, 1986), (Broy & Rausch, 2005), (Haskins, 2006), and (Office of Government Commerce, 2009)	
	Linear planning based on V-Model and derivatives (W-model, Butterfly-Model, Advanced-V-Model, VM-Model)	(Boehm, 1979), (Rook, 1986), (Morton, 2001), (Mooz & Forsberg, 2001), (Sheffield, 2005), (Höhn et al., 2008), (Clark, 2009), (Friedrich et al., 2009), (Bundesministerium des Innern, 2009), (Mathur & Malik, 2010), (Lau et al., 2011), (Bajaj & Narang, 2012), (Deuter, 2013), (McHugh et al., 2013) and (Sheffield et al., 2013)	
	Linear, concurrent modelling based on Axiomatic Design	(Suh et al., 1978), (Suh, 1990), (Albano & Suh, 1994), and (Suh, 2001)	
	Iterative approaches based on PDCA, Iconic Model, Innovation planning	(Shewhart, 1939), (Asimow, 1962), (Mesarovic, 1964), (Ertas & Jones, 1996), (Deming, 2000), (Kumar, 2003), (Kulak et al., 2005), and (Wiendahl et al., 2007)	
	Based on iterative, agile methods for software development (Spiral Model, Rational Unified Process, Scrum)	(Gould & Lewis, 1985; Boehm, 1988; Schwaber, 1997; Highsmith & Cockburn, 2001; Kroll & Kruchten, 2003; Pries & Quigley, 2010)	
	Combination of linear and iterative planning	For volatile requirements management	(Anitha et al., 2013)
		Iterative functional decomposition and gating	**Proposed in this thesis (Chapter 3)**
		Iterative concurrent decomposition and gating	**Proposed in this thesis (Chapter 4)**

Multi-disciplinary projects, like microsystem development projects, will benefit from the rigour of linear models, but coincidently also from the agility of iterative methods. Combinations of both models are experimentally applied in industrial environments, but such applications are still scarce in scientific literature. Table 1.1 lists the investigations performed on these topics.

Key Limitations:

- Models that support the early phase of design (conceptual design) are only sparsely available;
- Linear models could serve as a basis for microsystem development, but these methods lack an iterative approach to address the agile nature of microsystem development;
- Iterative methods apply the agile approach well, but these methods are missing the rigour of linear models for monitoring overall project progression;
- Linear and iterative models were in the past generally developed for software projects and as such they are not optimised for concurrent design.

The first research objective aims to combine the strengths of the linear and the iterative models in order to objectively determine design progression during development of microsystems. The research objective for Chapters 3 and 4 of the thesis was defined as:

First Research Objective:

Combine the overall indexing quality of existing linear project planning models with the agility of existing iterative models. The resulting model enables agile and concurrent monitoring of project uncertainties for the development of microsystems and their production means.

The objective will be addressed by the introduction of a ‘Microsystem Development Framework’, further referred to as ‘µSD Framework’, that merges linear and iterative ways of product development for microsystems. The investigations for the *µSD framework* particularly focus on two aspects of the design process: (i) ‘Functional System Decomposition’ or break-down of the project in clear cuts, and (ii) ‘Functional Gating’ that determines the degree to which a project has progressed by assessment of the functional behaviour of the system. As the project progresses, decomposition evolves and gates are successively closed; elemental project decisions are frozen when underlying project issues (at lower hierarchical levels) are specified. *Functional system decomposition* and *functional gating* are central research topics in the thesis.

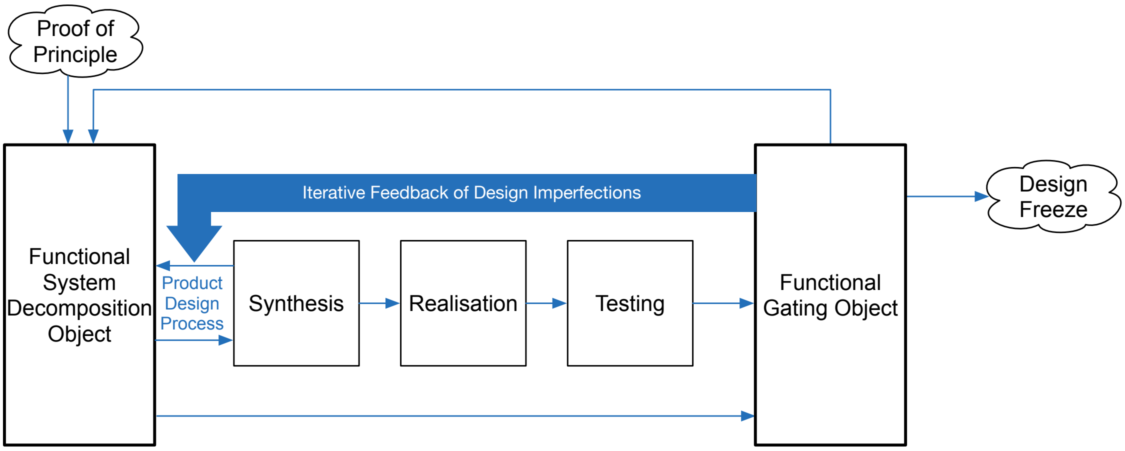


Figure 1.3 *Functional system decomposition and functional gating are elementary parts in the process of product design*

Figure 1.3 shows the global structure of the *µSD framework* that will be explained in detail in Chapter 3. Chapter 4 adds concurrent ways for these topics, called ‘Concurrent Decomposition’ and ‘Concurrent Gating’. With these concurrent ways for investigation, the *µSD framework* is upgraded to the ‘Concurrent Microsystem Development Framework’ (*CµSD framework*).

1.3.2 Understanding Complexity as Products Evolve

The solutions for *functional decomposition* and *functional gating* will be enhanced with methods of 'Axiomatic Design' (AD) in Chapter 4. AD is a systems engineering methodology that applies mathematically substantiated methods using 'Axioms', 'Domains', and 'Design Relations'. It has the unique capability to model a product design and its process technology in a concurrent way. Secondly, it supports decomposition of the product design and process technology (concurrently). As the *CμSD framework* is based on AD, its monitoring function to measure project progression was built on satisfaction of the design axioms. In principle, there are only two axioms, the 'Independence Axiom' and the 'Information Axiom'. The Independence Axiom is typically satisfied before the Information Axiom. This means that a product design can have only three conclusively defined statuses: (i) no axioms satisfied, (ii) only the Independence Axiom satisfied, or (iii) both axioms satisfied.

Problem: Ideally, project progression would be monitored with a continuous measure or at least more than a number of three stages. In Chapter 4, the resolving power of the *CμSD framework* was enhanced by merging elements of the V-Model, using the products' state of decomposition as an extra measure for project progression; this divides both axioms in a number of four stages, bringing the total resolving power of the *CμSD framework* to a number of eight stages. The approach in Chapter 5 and 6 will go further; the question arises if decomposition of design is the best measure for *functional gating* for at least two reasons. At first, it is not unusual in practice that gates need to be reopened. The need for reopening of gates is found in the general principle of decomposition. Decomposition is applied to divide a project in clear cuts that can be well-understood, as the project as a whole is too complex to understand. This does not automatically mean that functional parts that have been defined are fully understood right away. As the parts still need to be decomposed further in subparts, it is a matter of good faith that these subparts constitute no obstruction for the definite feasibility of the project. However, this cannot always be guaranteed because it is too laborious to apply detailed investigations on every subpart in advance. As a result, their nature can appear to be unwilling and persistent. To recover, it is necessary to revise the decomposition tree which requires gates to be reopened. The second reason why decomposition may not be the best measure to increase resolving power is that the principle of decomposition registers 'what was accomplished in the project so far' (successful decomposition and specification up to some level), while it is maybe more important to know 'what problems are still to come' and 'what needs to be done to address them'. This would require a completely different approach for realisation of the gating function. Such a gating function would need to look ahead to the remainder of the project instead of looking back at the project achievements so far.

Current Situation: Basically, all gating functions implemented in Waterfall- and V-Models are based on progression of decomposition of the product design. the *CμSD framework* builds further on this principle by adding true concurrency in its approach. These methods rather look back on achievements than focus on remaining work to do. Table 1.2 shows background investigations performed for these topics. AD provides another perspective that was not applied by any of the methods so far; the relatively novel 'Theory of Complexity' in AD, that is wrapped around a third 'Complexity Axiom', could be used as a continuous measure. The Complexity Axiom is not as much explored in literature as the Independence and Information Axioms. Unfortunately, the Complexity Axiom cannot be applied to address these problems without modifications. Possibilities are investigated in Chapter 5.

Table 1.2 *Positioning the investigations application of information in design as a measure for project progression in relation to the current research status*

Issues Considered	***Addressed By***	***Related Work***
Reduce the uncertainties in the design through AD complexity analysis	Information in Axiomatic Design	(Hartley, 1928), (Shannon & Weaver, 1949), (Brillouin & Gottschalk, 1962), (Suh, 1990), (El-Haik & Yang, 1999), (Suh, 2001), and (El-Haik, 2011)
	Complexity in Axiomatic Design	(Suh, 1999), (Suh, 2005b), (El-Haik, 2005), (ElMaraghy et al., 2012), and (Efthymiou et al., 2012)
	Decomposition of complexity and information in design	**Proposed in this thesis (Chapter 5)**
	Visualisation of complexity as continuous measure	**Proposed in this thesis (Chapter 6)**

Key limitations:

- Resolving power of a project progression model, based on decomposition of the design, does not provide contemplated resolving power since gates are regularly reopened;
- Gates that need reopening indicate corrective actions in the process of project management and cause disruptions. It should be prevented.

According to the Theory of Complexity, uncertainty in design is directly related to a lack of knowledge of the designer, and this uncertainty is the cause for project risks (Suh, 2005a). As a result, presence of the right knowledge with the designer would lead to prompt identification of all project risks.

The second objective of this thesis is defined as:

> ***Second Research Objective:***
>
> *Apply elements of the Theory of Complexity in Axiomatic Design to enable knowledge based gating for project control, and to increase resolving power for monitoring project progression.*

As such, the investigations of Chapters 5 and 6 build further on the need for a reliable gating function. This method will be referred to as ‘Intelligent Gating’. Decomposition of complexity is based on ‘Information in design’ according to Shannon’s ‘Information Theory’ and will be executed in Chapter 5 (Shannon, 1948). Chapter 6 presents a method for visualisation of the decomposed information measures.

1.3.3 Assessing Reconfiguration Schemes of RMS

The models of Chapters 3 & 4 may be applied to model risks in the development process of microsystems and their manufacturing means. This enables objective determination of the project status and how it evolves. The *CμSD framework*, and its related models, support decision making procedures during project execution.

Problem: However, the scope of the *CμSD framework* is limited to a single project. In practice, a factory is usually producing a portfolio of products that needs to be optimised. Strategic manufacturing decisions cannot be taken based on a single product/project; synergy with the rest of the factory influences the decision-making process. For these situations, the *CμSD framework* needs to be expanded with a method that compares options for known and new manufacturing solutions at the level of the factory floor, and as such anticipates further in the future.

Current Situation: So far, research on this topic has offered methods to compare different manufacturing principles (DMS, FMS, and RMS) (Zhang et al., 2006a) and also studies flexibility of FMS by development of optimised tooling (Hollstein et al., 2012). Also, investigations have been performed from the economic perspective (Kuzgunkaya & ElMaraghy, 2009; Amico et al., 2006). The advantages of RMS reside in the reuse of the modular building blocks or 'Process Modules' (Gutierrez, 1999; Grosser et al., 2000; Puik et al., 2002). Tailor made manufacturing systems can be largely composed of existing *process modules* that are chosen from the companies 'Library' of available modules according to a 'Reconfiguration Scheme'. However, as technology evolves, the library needs to be expanded with new *process modules* that foresee in added functionality and enable more powerful *reconfiguration schemes* to be applied.

Table 1.3 shows the positioning of assessment of manufacturing systems in relation to the state-of-the-art.

Table 1.3 *Positioning the investigations for assessment of reconfiguration schemes of this thesis in relation to the current research status*

Issues Considered	***Addressed By***	***Related Work***
Assessment of manufacturing systems	Comparison of dedicated, flexible, and reconfigurable manufacturing systems	(Amico et al., 2006), (Kuzgunkaya & ElMaraghy, 2009), (Zhang et al., 2006a), (Michaelis & Johannesson, 2012), and (Nassehi et al., 2012)
	Evaluating alternative designs for flexible manufacturing systems	(Abdel-Malek & Wolf, 1991), (Abdel-Malek & Wolf, 1994), (Lotfi, 1995), and (Yan et al., 2000), and (Hollstein et al., 2012)
	Adaptability of reconfigurable manufacturing systems	(Abdi & Labib, 2003), (Spicer et al., 2007), (Abdi & Labib, 2007), (Farid & Mcfarlane, 2006), (Farid, 2008), (Hasan et al., 2013), and (Farid, 2014)
	Evaluating alternative configurations for reconfigurable manufacturing systems	**Proposed in this thesis** **(Chapter 7)**

Key Limitations: Developing new *process modules* is labour intensive and it should be part of the companies longer term development strategy. The decision to expand the library of *process modules* should be weighted at the manufacturing department level in close cooperation with the department that executes product planning e.g.; it may be preferred to invest in a new *process module* and expand the library of modules today, because the new *process module* will be important for future products and they are members of the same product family. The decision exceeds the scope of the actual project alone:

- The process of reconfiguration of RMS influences the development of manufacturing resources (*library* of *process modules*). It has consequences for future application of RMS;
- No specific tools were found to model this problem. The problem transcends the current (single) project and should escalate to an adequate level;
- The alternatives should be compared quickly and effectively to enable comparison of many alternatives.

The research objective for this topic was defined as:

Third Research Objective:

Develop a method for assessment of different RMS reconfiguration schemes, by expanding the CμSD framework, that enables optimal reconfiguration for the present and the near future.

Assessment of *reconfiguration schemes* is investigated in Chapter 7 of the thesis.

1.3.4 Synthesis of a Generic Framework for Controlling Product Design

The final contribution of the thesis is to combine achievements of expository results of the analyses in Chapters 3 to 5. The result aims for a generic framework to model design progression, and reduce uncertainties during concurrent development of microsystems.

Problem: Such a general framework does not exist. It may be constituted by complementary models that together meet the exhaustive goal of monitoring the total product development process. The framework should inherit the method for implementation of iterative improvement cycles combined with some linear method for determination of the absolute status of the project. To achieve this, the gating function from the *CμSD framework* may be expanded with the method for *intelligent gating* and the capability to measure the knowledge of the designer as applied to the product design. The method should maintain its concurrency to address product and process design simultaneously, and have the ability to monitor the development in the explorative, conceptual, and robustness phases.

Current Situation: There is a diverse availability of systems engineering methods that serve particular goals. These methods can be collectively applied to control the design process. Only few models have the capability to integrate these methods from a higher perspective and make the aggregate of the models prevail over the joint performance of the mutual models e.g., the V-Model, and also the more traditional Waterfall-Models have been reasonable successful in this sense, but many other models focus on specific project details.

Of the available models, only a limited selection addresses the explorative phase. Some of the models that do address the explorative phase, do not address the robustness phase. Only a number of three models were found that can be applied over the full development range (three phases, exploration, conceptualisation, and robustness).

Even less models are based on testing the knowledge of the designer and if it is applied to the product design, and lastly, very few models have the capability to address product design and process engineering concurrently. No model was found that includes all these features.

Key Limitations:

- An integrated model should have the capability to track the knowledge of the designer and monitor if that knowledge is implemented well;
- It should cover the total development process, from earliest design stage to total robustness;
- It should have the capability to embed other existing models;
- Many models do offer partial solutions for the concurrent product development of microsystems. Unfortunately, no complete and guiding framework that provides all capabilities is currently available to the designer.

An inventory of methods that support modelling in the early design stage was made (Table 1.4). The framework in Chapter 8 will be based on the Theory of Complexity of AD and as such builds further on the method of *intelligent gating*. Based on the findings in this thesis, a modular model could be developed that addresses the shortcomings of existing models.

Table 1.4 *Positioning the investigations for alternative models to monitor the complete product development process in relation to the current research status*

Issues Considered	***Addressed By***	***Related Work***
Modelling the explorative development phase	Development models that address the early design process	(Deng et al., 2000), (Wang et al., 2002), (Tay & Gu, 2002), (Ulrich & Eppinger, 2004), (Haskins, 2006), (Ayag, 2007), (Komoto & Tomiyama, 2012), and (Pahl & Beitz, 2013)
	Models, based on AD, that address the early design process	(Li et al., 2010), (Tay & Gu, 2002), (Zhang & Chu, 2010), (Chen et al., 2012), and (Benkamoun et al., 2014)
	Model, based on AD's Theory of Complexity, for knowledge based monitoring the overall design process	**Proposed in this thesis** **(Chapter 8)**

The final objective of this thesis was defined as:

Fourth and Final Research Objective:

Combine the best characteristics of all models available and preserve their strengths to form a new model that: preserves a risk adjusted approach, has good flexibility, and may be applied to concurrently develop microsystems and their RMS. The model is applicable in the early project stage and equally addresses managerial and technological issues.

It is not tried in this thesis to gather all functionality into a single model. Many of the mentioned capabilities have already been included in one of the models as described in previous chapters (*µSD framework, CµSD framework*, method of *intelligent gating)*. To prevent over-complication of a heavy integrated model that features all functionality, it has been chosen to develop a framework that is wrapped around these existing models. This new framework is called the 'Constituent Roadmap' because it connects to existing models that may be applied within its context.

1.4 Limitations and Assumptions

This research has limitations in application. As investigations focus on microsystems in general, it covers a wide scope of technologies: mechanical, optical, chemical, biomedical systems etc. All these technologies have characteristics that may require a particular approach. This thesis will not provide a solution to all problems that can occur with all these technologies. Neither will it provide a solution to predict all errors during the development of all these technologies. However, to increase its coverage, the cases chosen to validate the models will be selected from different markets, e.g.: A cell phone lens-array, a geometrical measurement system for nanometre measurements, a pneumatic automotive switch, an inkjet print head, and a Nanowire hydrogen sensor for the Internet of Things. All cases serve a commercial goal.

The complexity analysis in this thesis uses the complexity definition in AD as ‘a measure of uncertainty in understanding what it is we want to know or in achieving a functional requirement’ (Suh, 2005b). This is just one of the going definitions of complexity in general, as there are many different ways to look at the concept of complexity (Gell-Mann & Lloyd, 1996; Suh, 1999; Wildemann, 2013).

Also, some assumptions apply. In these investigations, it is assumed that ‘the designer’ is a person, or a group of persons, without personal or political motives or hidden agenda. It is assumed that the designer actually has the ability (capability, means, and sufficient time) to apply his knowledge to the problems he is investigating.

Lastly, a part of this thesis deals with process technology that is applied in manufacturing systems. The process technology that is applied to produce products and systems is considered to be a part of the product itself. If manufacturing is mentioned in the thesis, it is principally about the process of configuration of a manufacturing system (as an RMS). The exception is the case where new modular *process modules* need to be developed in Chapters 4 and 7. In these cases, the development of a new process module is seen as a product on itself.

1.5 Thesis Outline

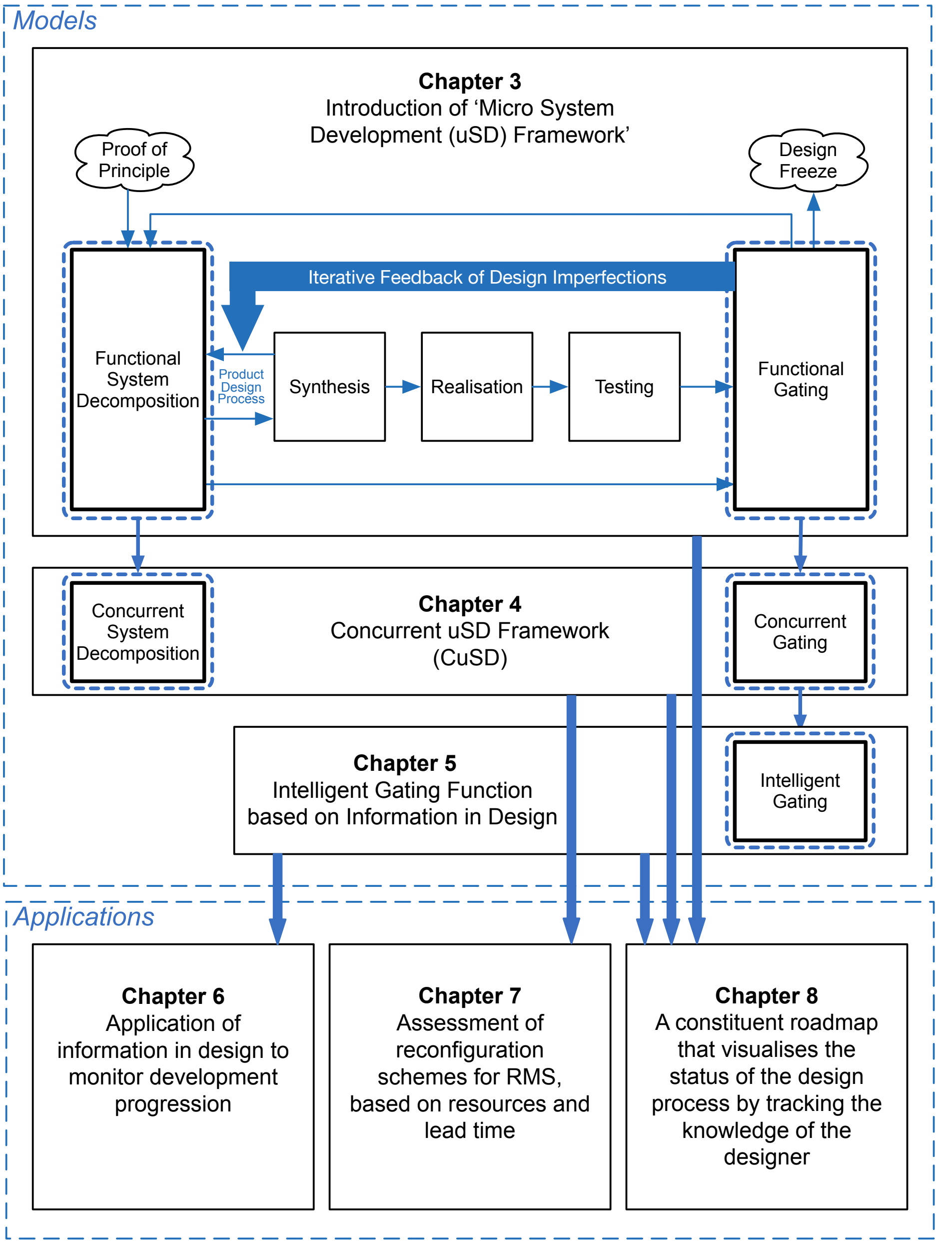


Figure 1.4 *Outline of the thesis*

This thesis consists of two parts. The first part, containing Chapters 3, 4, and 5, focuses on 'Models' that analyse microsystem product development processes. The second part, containing Chapter 6, 7, and 8, focusses on 'Applications' based on the analyses of the first part.

The outline of the thesis is shown in Figure 1.4. After Chapter 2 that focusses on background of models for microsystems and their manufacturing means, Chapter 3 introduces the *μSD framework*, and focuses on *functional system decomposition* and *functional gating*. Chapter 4 builds further on the *μSD framework* and adds the concurrent development approach in the *CμSD framework*. Chapter 5 adds *intelligent gating* based on Axiomatic Design, and that output is visualised in Chapter 6. The method for assessment in Chapter 7 builds further on Chapter 4, and finally, the *constituent roadmap* of Chapter 8 builds on the input from all analysis chapters (3, 4, and 5). The discussion in Chapter 9 reflects on the overall results of the thesis and draws conclusions.

Chapters 3 – 8 apply an equivalent structure that starts with 'Analysis & Approach' containing the problem statement, the current situation and its key limitations. This is followed by an exposition of the applied 'Methodology' of the investigations and how these address the key limitations. One or more 'Industrial Cases' are applied to demonstrate the methodology. Lastly, the 'Discussion' reflects on performance of the methodology.

CHAPTER 2
BACKGROUND

2.1 Background on Microsystems

‘Cramming More Components onto Integrated Circuits’ was the title of the influential paper of Gordon Moore in 1965 (Moore, 2005) (IEEE reprint). Moore predicted that the number of components on a semiconductor substrate would double every 24 months. Moore also stated that the future of electronics would bring about a proliferation of electronics, pushing this science into many new areas such as home computers or at least home terminals, electronic wristwatches, and communications devices. The exponential growth of semiconductor electronics indeed has caused great changes in society and even Moore’s ambitious ideas have been exceeded; electronics and forthcoming software have connected the world to the internet.

2.1.1 More than Moore, MEMS, Micromachines, and MST

In the mid-eighties, elements of mechanical systems were realised on the surface of semiconductor chips, by using the equipment that was developed for semiconductor manufacturing. The operations are called ‘Bulk Micro Machining’ or ‘Surface Micromachining’, the former oriented on etching material away from the silicon wafer while the latter focuses on deposition of layers on the surface of the silicon wafer (Madou, 2002). Despite the name suggests ‘machining’, both have little relation with traditional methods for mechanical machining of parts (Diem et al., 1995). By the end of the eighties, the micro machined electromechanical systems enabled mechanical engineering to make the next step in miniaturisation. The various movements, MEMS, Micromachines, and MST as referred to in the introduction of the thesis have their typical approach of the technology but all contribute to general ‘Microsystems’. In the US it was called ‘MEMS’ (Micro Electro Mechanical Systems) that remained close to semiconductor engineering having a strong relation with electronics, in Japan it was called ‘Micro-machines’ as it was characterised by having features in the micrometre range, and lastly ‘Micro Systems Technology’ in Europe, which typically used some electronic chip in combination with other elements giving it a more hybrid character (Heinig et al., 2014). In this thesis, all these systems are referred to as ‘Microsystems’ and their specific character is not only determined by the feature sizes within the devices, but also their characteristic to combine several disciplines e.g.: mechanical, optical, fluidic, biomedical, and software disciplines.

Where Moore’s law has led to a virtual world of electronics and software (computers, communications devices, internet, etc.), microsystems connect this virtual world to the physical world; the virtual world is adapting to humans, instead of the other way around. Figure 2.1 shows successful microsystems.

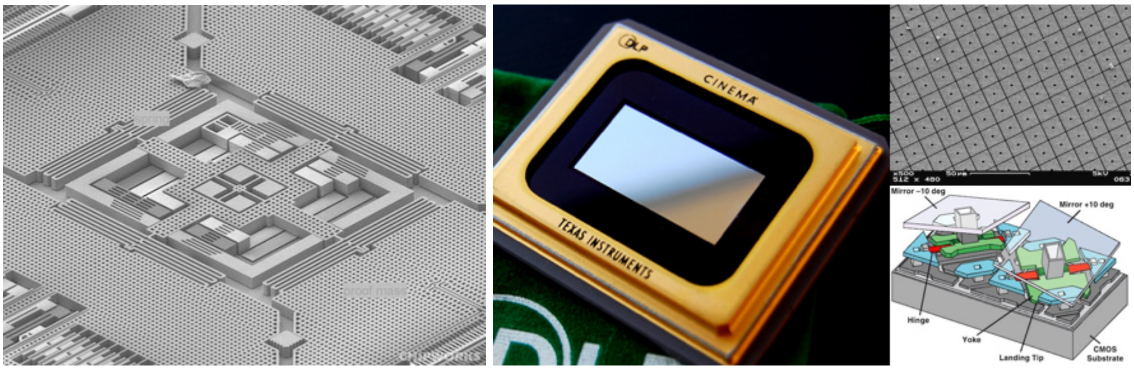


Figure 2.1 *iPhone 4S gyroscope by STMicroelectronics (left) and, Digital Mirror Device for data projectors from Texas Instruments (middle and right)*

Microsystems may be grouped in two major categories; sensors and actuators (Zhang et al., 2005); both concerning data transmittal, the former bringing data from the physical world to the virtual world (the world of electronics and software), the latter bringing data from the virtual to the physical world. Sensors measure temperature, speed, acceleration, presence, composition of gasses, etc., and transform these quantities into the field of electronics. Actuators bring electronic quantities back to our environment in the form of loudspeakers, displays, switches, light sources, etc., to basically address any quantity humans can sense. Semiconductor integration of sensors has been more successful than integration of actuators (Madou, 2011). The reason is that the small dynamics and corresponding energy levels of micro systems perfectly matches the dynamics for sampling small bits of the environment, however, actuation on macro level only makes sense if energy levels can be applied that correspond with the macro world. Figure 2.2 shows prototypes of micro actuators that were produced with state-of-the-art technology but did not meet this requirement.

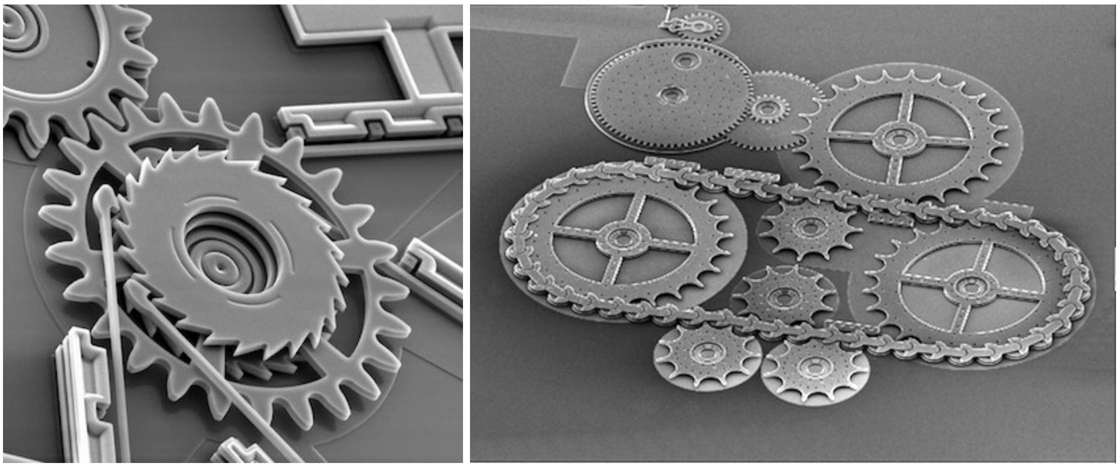


Figure 2.2 *Micro actuators as produced by Sandia National Laboratory (2002)*

Commercial successes were lacking for these systems due to their limited capability to drive anything on a sensible scale. Still, the Digital Mirror Device of Figure 2.1 was successful because the energy levels for controlling the micro mirrors is in harmony with the on-chip energy levels.

2.1.2 Monolithic and Hybrid Microsystems

In general, when the required energy levels to actuate the macro world divert too much from what is feasible on-chip, physically larger elements from neighbouring disciplines need to be integrated (e.g. mechanics, optics); this gives the micro systems a hybrid character. Hybrid systems are 'Non-Monolithic' which means they are not integrated on a single chip. Instead, they are composed of multiple elements to realise their primary function, and as a result, assembly actions are needed for integration (Kear, 1992; Tichem & Tanase, 2008; Heinig et al., 2014). Where monolithic sensors for common quantities, such as temperature or acceleration, are addressing mass markets with production numbers up and beyond 10^9 annually, hybrid systems typically address many niche markets with production numbers in the mid volume area, in between 10^4-10^7 products annually (Lang, 1999; Vigna, 2005). This thesis focusses on effective design and manufacturing of hybrid microsystems; hybrid microsystems consist of a monolithic part, typically a micro machined chip, usually called 'Die', and a non-monolithic part of some other physical discipline, that are brought together with assembly actions. The Dies in hybrid microsystems are based on semiconductor technology and may be expected to follow the relatively aggressive Moore's law. However, the hybrid parts of these systems that are realised with more conventional production methods such as milling, grinding, moulding etc., will follow traditional dynamics of these technologies. Also the assembly of hybrid microsystems is not exempted from the basic nature of general production principles (Onori 2009). It does not profit from the high level of standardisation from semiconductor manufacturing and it makes development considerably slower and more cost intensive. Due to the general level of difficulty of micro fabrication & assembly, it introduces substantial risks for development and manufacturing of hybrid systems (Van

Brussel et al., 2000; Zhang et al., 2006b; 2005; Dimov et al., 2012; Voelkel et al., 2012; Onori, 2009).

2.1.3 Concurrent Design of Microsystems

An approach to modularise microsystems has been the most important driver to increase flexibility and deal with change in the highly dynamic markets accordingly. Gutierrez (Gutierrez, 1999) introduced a modular design framework based on standardised ‘Micro-Bricks’ to compose microsystems from standard building blocks. This design framework was expanded with a modular production framework that provides standardised process technology for the interfaces of the Micro-Bricks to enable integration on RMS (Puik et al., 2002; Puik & Moergestel, 2010). Another attempt was the German ‘Projekt 2000’ which endeavoured to merge production of microsystems closely with semiconductor equipment (Grosser et al., 2000). Standardised interfaces for manual as well as automated manufacturing stations enabled a volume upgrade scenario. A second German attempt for standardisation of microsystems was MatchX (Stock & Schünemann, 2002) wherein standard building blocks were stacked using a standardised bus for interconnection. All three attempts have two common factors:

- All methods applied modular structures for product design and manufacturing to enable flexibility;
- The design and the methods for integration were addressed concurrently.

Concurrent approach is a central theme in development of microsystems. As manufacturing means are addressed at or beyond their maximum capability to produce with small tolerances, functionality of products and production yield of equipment may fail to deliver adequately. Therefore, it may be necessary to modify a product design to be more tolerant for manufacturing accuracies. When a product is redesigned for manufacturing, its performance may be reduced, e.g. a sensor system may have limited accuracy or reduced efficiency, but as a trade-off its production process will become more tolerant, have a better manufacturing yield, and therefore it will be more cost effective.

2.1.4 Manufacturing of Microsystems; Application of RMS

There is no clear public roadmap such as Moore's law available for micro assembly (Gutierrez, 1999; Brecker, 2005; Heeren et al., 2004). The variety of processes, mainly for hybrid assembly, leads to a diverse need of process technology. Some processes introduce little manufacturing risks because they were successfully applied (many times) before, but others will need specific improvements or total development. The potential of reusing existing manufacturing technology gives RMS an advantage in application. The reuse of standard building blocks draws little attention of the engineers so their focus may be shifted to the development of new building blocks. The initial advantage of RMS is not their reconfigurability, but the advantage that the system may be flexibly configured in the first place; every RMS starts as a configurable manufacturing system. The advantages of reconfiguration will prove their benefits further in time. Though many successful applications exist (Abdi & Labib, 2003; Koren et al., 1999; Garbie, 2014), reconfiguration is not always applied, or not applied at the location of the customer. The main reason for this is that the complexity of RMS may be a barrier for actual reconfiguration (Puik & Moergestel, 2010; Semere et al., 2013). The documentation of RMS in industry leaves much to be desired (Onori, 2009). This means that the knowledge about its configuration and the applied process technology are not, or not fully, secured in documents and may fade after time. Especially when the original designers are no longer available, reconfiguration of these systems is risky; the new engineers have no solid understanding of the equipment yet, and the operator, though a specialist on operating the equipment, has no knowledge of reconfiguration. There are substantial risks of malfunction, reduced yield, and non-operating equipment for a longer than forecasted time if changes are carried through to these moderately documented systems. The more complex the system has become, the larger the knowledge-gap will be, and the less probable that reconfiguration will take place in the field. In problematic cases, the RMS may be shipped back to the location where it was originally configured, usually the site of an equipment supplier. This reduces the problem of complex reconfigurations because more specialised resources and expertise are available at that location (even if it is an internal department). The drawback of this scenario is that production ceases during reconfiguration and the duration will be extended due to transportation and planning issues. Generally, RMS are of more value to a company if their complexity is kept low (Puik & Moergestel, 2010).

2.2 Methods for Linear Project Planning

2.2.1 Early Work on Design Science

Early work on design science is mainly from German schools with Hansen dating back to the fifties as reported in the book ‘Konstruktionssystematik’ (Hansen, 1966) and the widely spread book of Pahl & Beitz (Pahl & Beitz, 2013) in the late sixties. This last work was well maintained with rewrites up to 2013. The ‘Theorie der Maschinensysteme’ is a design science framework presented in 1973 by Hubka (Hubka, 1973). This work introduces design assumptions (like axioms) and vector representation of domains. From here, the established theories were developed inter alia by Suh with the Axiomatic Design methodology (Suh et al., 1978), but also e.g. Andreasen with the ‘Domain theory’ (Andreasen et al., 2014).

2.2.2 The Waterfall Model

In the sixties, the notion to monitor the course of the development process gained in support. With focus on the integral process more than the iterations, it made project control a linear process. Widely applied models in industry are the ‘Waterfall-Model’ and the ‘V-Model’. Royce, who was the first to report the Waterfall-Model (Royce, 1970) criticised the model in the same article blaming the lack of (old school) process iterations and testing. The Waterfall-Model also forms the basis for the process-model of the ‘PRINCE’ method that was introduced in 1989 (PRojects IN Controlled Environments). PRINCE’2’, was a continued development to enable broader application than PRINCE that was mainly intended for ICT developments. PRINCE2 was developed in 1996 and is still maintained to date by the British semi-governmental organisation ‘Office of Government Commerce’ (Office of Government Commerce, 2009). The current implementation addresses project management in a broader context than a process-model by further adding ‘Management Principles’ and ‘Management Themes’.

2.2.3 The V-Model

The V-Model, also based on the Waterfall-Model, was originally introduced by Boehm (Boehm, 1979) and simultaneously developed further in Germany and the US in the second half of the eighties (Rook, 1986; Friedrich et al., 2009). In the 1991 proceedings for the National Council on Systems Engineering (NCOSE); now INCOSE as of 1995, the V-Model was adopted in the US for modelling of mainly software systems. Though introduced as a life-cycle model, it is in fact a planning tool, comparable to the

PRINCE2-process-model. A specific feature is that it adds solid testing functionality at different hierarchically decomposed levels. There are three basic parts: ‘Specification & Decomposition’, ‘Realisation’, and ‘Integration & Testing’. The first and last parts are usually divided in three sequential hierarchical levels (Figure 2.3) but this can be expanded as required.

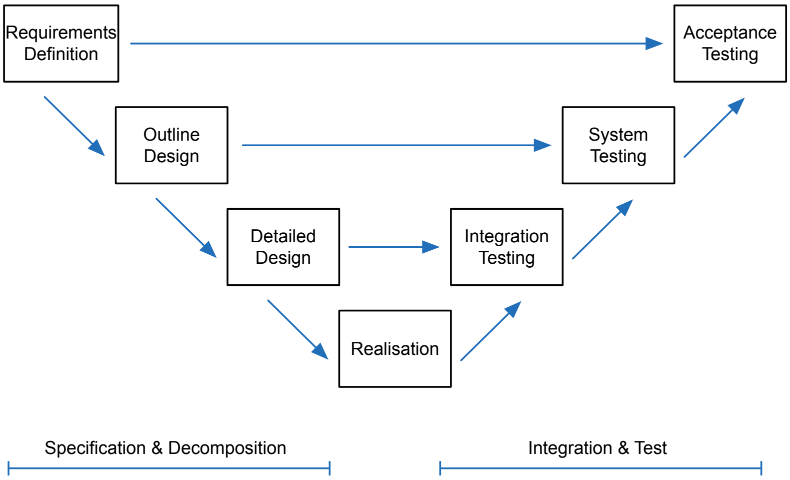


Figure 2.3 *The V-Model in its initial appearance*

Like all basic Waterfall-Models and PRINCE2, the V-Model suffers from the problem of ‘missing iterations’ (Christie, 2008). This is not as much a problem to accountants and project managers as it is for developers and testers. The most damaging aspect might be the effect that the V-Model effectively discourages user involvement in evaluating the design before arriving at the formal testing stages. By then it is too late to make significant changes to the design. It must be mentioned that the need for sufficient iterations was emphasised when Rook introduced the V-Model, but since the model does not specifically visualise it, unilateral application of the model has become the standard for most industrial applicants. Nevertheless, the V-Model, and in somewhat lesser extent the Waterfall-Model, today are popular systems engineering methods in industry since they meet needs for management.

2.2.4 The V-Modell XT

Boehm was the first to introduce a V-Model for systems design (Boehm, 1979), though the name V-model was first introduced by Rook (Rook, 1986). In parallel, the German ministry of defence developed their V-Modell (German spelling). This was released for civilian use in 1991 and upgraded a number of times (V-Modell 92, V-Model 97). Since it was poorly maintained after '97, a new initiative was started in 2004 to bring it back to state-of-the-art level as the V-Modell XT (Extreme Tailoring), making the earlier versions obsolete (Broy & Rausch, 2005). A number of improvements were implemented to enhance adaptability, scalability, changeability, growth potential, and to meet novel standards, but the most characteristic changes were broader applicability than just ICT projects and the extension of the V-Model to the entire system life cycle of a product. The V-Modell XT is till to date maintained by the German 'Industrieanlagen-Betriebsgesellschaft mbH'.

The V-Modell XT has generally four hierarchical levels that may be adapted for specific situations. Analogue to the V-Model as introduced by Rook, the highest level is the level of project acquisition and the lowest level is the level of parts. The remarkable difference is the strict separation between the left- and the right-hand leg of the model. This is shown in Figure 2.4. The left-hand leg addresses 'Specification and Decomposition' and the right-hand leg 'Integration and Test'. This is slightly different from Rook's version of the V-Model where the lowest hierarchical level is integrated into a single stage for 'Code and Unit Test'.

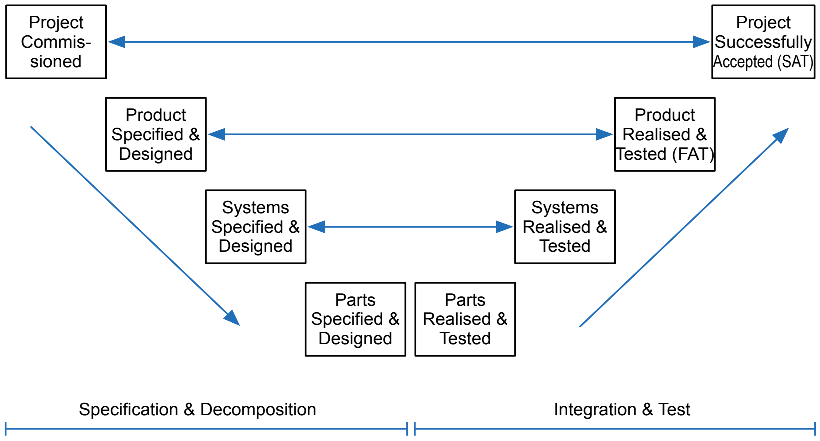


Figure 2.4 *Structure of the V-Modell XT*

2.2.5 Drawbacks and Developments of the V-Model and its Derivatives

One feature that is regularly questioned in the V-Model and V-Modell XT is the fact that these models discourage exploration and testing in the early stages of design (Christie, 2008; Liversidge, 2005). Early stages in this context means the left-hand side of the V. This problem partially arises from the lack of understanding of the model. Rook advises to regularly take the following two actions: (i) update the specifications, and (ii) revise decomposition (Rook, 1986). An issue is that the graphical presentation of the V-Model does not indicate anything of these iterations, and testing indeed is delayed till the right-hand stages. Newer variants of the V-Model have addressed these issues like the W-Model (Herzlich, 1993) and the butterfly model (Morton, 2001). These models implement a kind of recursive operative. Unfortunately, the implementation is incomplete and the models did not get much traction (iterations are limited to a single level of decomposition, concurrency is not supported). Though the V-model was presented over 30 years ago, discussion is still active and many variations of the model are still being developed (Anitha et al., 2013; McHugh et al., 2013; Höhn et al., 2008; Mathur & Malik, 2010). In literature, the iterative approach of investigations, research, and developments appears significantly older than the linear or sequential project control methods that were developed in the early seventies and eighties. In the nineties, an overshoot of the linear approach seems to be compensated with a renewed focus of iterations in design to address weaknesses of the linear methods. This will be addressed in Section 2.3.

2.2.6 Axiomatic Design

Axiomatic Design (AD) declares 'Axioms' that cannot be proven nor derived from physical phenomena. A number of seven conceptual axioms were defined in 1978 when the first paper about AD was presented (Suh et al., 1978). Two of those seven axioms stood the test of time and form the foundation of AD today, now known as the 'Independence Axiom' and the 'Information Axiom'. The Independence Axiom advises to 'Maintain the independence of the functional requirements', the Information Axiom recommends to 'Minimise the information content of the design'. A product design will be a 'Good Design' if both axioms are satisfied.

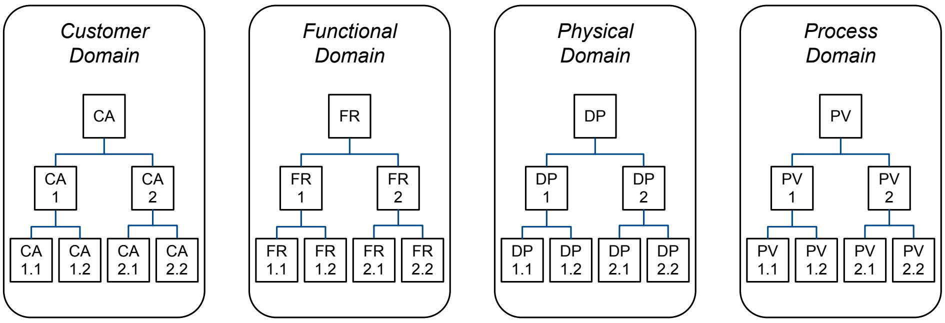


Figure 2.5 *Axiomatic Domains and their hierarchical organisation*

AD demands clear formulation of design objectives through the establishment of 'Domains' called: (i) 'Customer Attributes', (ii) 'Functional Requirements', (iii) 'Design Parameters', and (iv) 'Process Variables' (Figure 2.5). These domains are hierarchically organised and mathematically related by a process called 'Mapping'. Mapping applies

‘Design Matrices’ that follow decomposition. AD is explained more in detail in Appendix B.

2.2.7 Methods to Support Decomposition and Synthesis

An iterative development cycle will require some form of analysis and synthesis that supports decomposition of the system. A number of methods have been applied in literature with some regularity.

‘Quality Function Deployment’ (QFD) is a value-engineering tool that is usually applied for mapping customers’ wishes in relation to a product design. The method transforms user demands into design quality, in order to deploy the functions forming quality. Secondly, it deploys methods for achieving the design quality into subsystems and component parts to ultimately specify elements of the manufacturing process (Akao, 2004).

‘Structured Analysis Design Technique’ (SADT) was originally developed for software development but appeared to have a much broader application area (Ross, 1977). It was adopted by the U.S. Airforce as IDEF0, and for manufacturing purposes, SADT has been refined to focus on errors that tend to inherit through subsequent process steps. This method is called ‘Qualitative Modelling and Analysis of Processes’ (QMAP) (Bullema et al., 1999; Brands & Weert, 2000). Structured analysis methods, either SADT or QMAP, further referred to as SADT, can be applied when no hardware has been realised yet. This makes the method particularly suitable for the early stage of development.

TRIZ, or the ‘theory of inventive problem solving’ (Teoriya Resheniya Izobretatelskikh Zadatch) is a Russian analysis and forecasting tool derived from the study on patterns of invention in the global patent literature (Hua et al., 2006). The theory was developed on a foundation covering thousands of inventions across many technological fields. TRIZ defines generalisable patterns in the nature of inventive solutions and the distinguishing characteristics of the problems that these inventions have overcome. These solutions have been combined in an algorithmic approach to the invention of new systems and the refinement of existing ones.

Decision Matrix Analysis, by application of the 'Morphological Matrix' or the 'Pugh Matrix', are useful techniques for making informed decisions (Kroll, 2012; Hubka, 1973). A condition is that there are multiple options from which can be chosen, and many different factors to take into account. Decision Matrix Analysis works by drawing up (usually) a two-dimensional table with the factors that need considering as columns and the options to satisfy them as rows. Then, the options are scored for each factor taken into account. Eventually, the scores are weighted by the relative importance, and finally the scores are added up to give an overall score for each option. The method may also be reversed to weigh risks in the design. These methods are known as risk-analysis-techniques such as the widely-applied Failure Modes and Effect Analysis (FMEA). Decision Matrix Analyses are old methodologies and nowadays universally applied by most designers and in many ways.

QFD and Morphological Analysis are mainly suitable for the decomposition during product development, while SADT is more suitable for decomposition of manufacturing systems since it addresses a sequence of occurrences. TRIZ could be applied for the syntheses of solutions. Since this research focusses on the close relation of product development and its RMS (Design for Manufacturing), a decomposition-method is selected that supports assembly and manufacturing. The most suitable tool is SADT due to its sequential nature and the capability to analyse the hierarchy of the system.

2.2.8 Combinations of Systems Engineering Models

Combinations of different systems engineering models have been made in literature as well. As such, the V-Model and AD have been combined before by Do & Suh (Do & Suh, 1999; 2000) where it was applied to object oriented programming. Comparable is the implementation of Fiege and Stelzer (Fiege & Stelzer, 2007) for service oriented ICT architectures. In both cases the development of software hierarchy is executed in the left-hand leg of the V-Model and the realisation of the object-oriented model in the right-hand leg. Intermediate stages are: definition of FRs, mapping to DPs, and Decomposition. In the right-hand leg this is respectively: the identification of classes, establishment of interfaces, and coding with the system architecture. A completed design matrix separates the left- and the right-hand leg and as such indicates the tip of the V.

SADT and QFD were also applied before in relation to the V-Model and AD for modelling of manufacturing systems. Triki applies SADT and AD for the optimisation of an equipment occupation ratio (Triki et al., 2011). Kim combines AD and SADT for development of control software of manufacturing systems (Kim & Suh, 1991), and Buseif applies SADT for the design of FMS (Buseif & Elfeituri, 2006). Gonçalves-Coelho et al. apply QFD and AD to analyse the voice of the customer in concurrent design (Gonçalves-Coelho et al., 2005). Puik et al. combined the V-Model with AD to define a method for indexing the development of Reconfigurable Manufacturing Systems (Puik et al., 2013a). Though applied for RMS, the method is generic and may also be applied to index the product design process. The application of AD is straightforward; the

Independence Axiom is applied for the left-hand leg of the V-Model and the Information Axiom is applied the right-hand leg. The tip of the V is reached when the Independence Axiom is Satisfied. For product designs, QFD can be applied for decomposition as QFD transforms user demands into functional requirements to deploy the functions forming quality (Akao, 2004). For decomposition of process technology QFD is replaced by SADT.

2.3 Iterative Models for Project Control

2.3.1 Early Methods for Iterative Development

Shewhart described in 1939 the ‘PDCA cycle of continuous improvement’ based on the principles of empiricism as induced by Bacon in his 17th century work ‘Novum Organum’ (Shewhart, 1939; Bacon, 1620). The initial Plan-Do-Check-Act was advertised more broadly by Deming who replaced the stage ‘Check’ by ‘Study’ to emphasise that the analysis in this stage was to prevail over inspection (Deming, 2000). The method was optimised in the sixties by respectively Asimow and Mesarovic as the ‘Iconic model of the Design Process’ (Asimow, 1962; Mesarovic, 1964). The Iconic model introduces the cycle of Analysis, Synthesis, Evaluation, and Communication. This foundation forms the basis for modern iterative models for iterative project control up to date.

2.3.2 Agile Methods for Project Management: Renewed Focus on Iterative Development

After the development of the Waterfall-Model and its derivatives, methods for iterative development were regaining in popularity towards the end of the Millennium. This was driven by the need for a combination of structure and dynamics in the ICT world. Two schools of thought (Mooz & Forsberg, 2001) may be recognised: on one hand the more conventional form as characterised by the application of the classic waterfall scheme (PRINCE2 and V-Model), and on the other hand, the more agile school named 'Agile Development' that tries to overcome some of the drawbacks of the process oriented waterfall-based methods. Agile software development methodologies focus upon incremental design and hence a cyclic approach. The aim with these methods is to: (i) make the development process more responsive in changing environments, (ii) pursue functioning software over extensive documentation, (iii) centre individuals and their interactions rather than tools and processes, and (iv) value customer collaboration over customer contract negotiation.

Of great influence are the 'Spiral Model of Software Development' by Boehm (Boehm, 1988), the 'Engineering Design Process' by Ertas & Jones (Ertas & Jones, 1996), HP's 'Product Development Process', the 'Scrum development method' (Schwaber, 1997), IBM's 'Rational Unified Process iteration cycle' (Kroll & Kruchten, 2003), and Kumar's 'Innovation Planning' (Kumar, 2003). Except for the last method, which is for optimisation of general services and design, all these methods were initially developed to streamline software developments but later on found their ways for broader application.

Scrum may be considered the most valued form within the family of agile development methodologies. Scrum uses incremental development procedures with an objective to get working software into the hands of the stakeholders as quickly as possible. This way of working puts business value functions into stakeholder possession early on in the software development life cycle. The more traditional process oriented development methods cannot provide this agile capability; stakeholders typically would not have access to any software produced until far later in the process. This agile performance is provided in a straightforward procedure that enhances focus and communication in an iterative process. Scrum starts with the business case just as one would do with process-oriented development. From this point, it diverges from linear development methods. The customer requirements are inventoried and refined in close cooperation with stakeholders and the project group (comparable to Mesarovic's Iconic *Analysis*). The remaining requirements or 'User Stories' are kept in a list known as the 'Backlog'. Cycles or 'Sprints' are initiated from the backlog to address the customer requirements with the objective to produce operating solutions (comparable to Iconic *Synthesis*). The solutions should be fully functional, tested, and documented with the ability to be shipped as a finished product, though with limited functionality (Iconic *Evaluation*). Sprints may last from one week to a month and their progression is kept in a 'Burn Down Chart' to feed its status back to the team (Iconic *Communication*). A structure of usually brief meetings takes care of extra information exchange within the project team and leads to joint decisions that are supported by the customer as he regularly participates meetings.

Scrum and related agile methods also suffer from drawbacks compared to the traditional methods. It may fail at the following aspects: (i) a drawback according to Highsmith & Cockburn (Highsmith & Cockburn, 2001) is the fact that an external client has to be actively involved in the project. The client has to be able and available to test the typical monthly releases and to suggest new or modified functionalities, (ii) by applying Scrum, the vision of the client highly influences development. Highsmith & Cockburn show that if the client does not have a clear sense of the product's direction, the members of the development team will tend to behave in the same way, and the final product can be significantly different to what is expected. This makes the main strength of Scrum also one of the main weaknesses: client involvement in the development process, and (iii) another potential weakness is the relatively low visibility over the project outside sprints. This makes it difficult to estimate how long a project will take or how much it will cost. In projects with external clients, where bidding is used to determine the contractor for projects, this can be a major drawback.

2.3.3 Social Approach of Design

Over the years, designers have moved closer to the users that apply their products (Sanders & Stappers, 2008). Parallel to the methods that are strongly relying on a technological basis, like the V-Model and AD, methods were developed to focus stronger on the social and interactive side of design. An approach gaining in interest is that of ‘Participatory Design’ or ‘Co-Design’ that finds its way in the sixties (Sanders & Stappers, 2008). The goal is to join forces with all stakeholders of a future design to define a product that will receive broader acceptation. It changes the role of the designer from ‘translator’ of the user’s wishes to ‘facilitator’ to stimulate creativity. This makes participative design suitable for the early design of new products. The early and explorative phase in participative design is also called the ‘Fuzzy Front End’ (Figure 2.6) due to its chaotic and unpredictable nature.

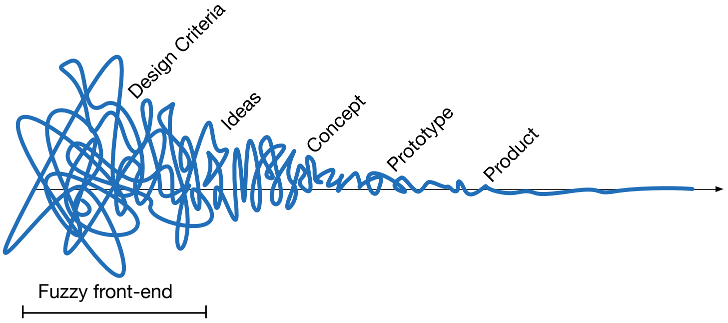


Figure 2.6 *The fuzzy front-end of product design*

These methods acknowledge the elusiveness of the early development stages and study the complex processes, which contrasts with the more technological oriented approach in industry that aims to understand the complexity of the early development stages. Of the tension thus created both streams can benefit.

Another more socially engaged method to approach design is 'Design Thinking' originally based on Simon's reference work 'The Sciences of the Artificial' from 1969 (Simon, 1996) (reprinted). Though initially applied for architecture and later as a process for problem solving in design, Design Thinking consists of a linear sequence of gated actions consisting of: (i) definition, (ii) creation of many options, (iii) refine promising results (eventually repeat step ii and iii), and (iv) select a winner and execute. Design Thinking is especially suitable in the early stage of the fuzzy front-end. It implements iterations and is initially highly diverging.

2.4 Phases in the Design Process

2.4.1 Models for Explorative & Conceptual Design Combined with Axiomatic Design

Though there are not many generally applicable models for the explorative and conceptual design phases, a lot has been reported in literature. Though dated, a good overview of work till 2002 is given by Wang & Hu (Wang et al., 2002). More recent is the work of Ayag (Ayag, 2007) and the work of Li et al. (Li et al., 2010). After closer investigation, quite some conceptual design methods appear to have been presented over the last thirty years. These models can be classified into three categories based on focal points and tools used: (i) design models according to the design criterion of products, (ii) design models based on the design strategies of products, and (iii) design models adopting artificial intelligence. As this investigation focusses on the first category, where the designer has the traditional role of being in charge of the design process, category (ii) and (iii) are not further investigated.

More recent models of the first category include the work of Li et al. (Li et al., 2010) where a method is presented based on AD with alternative domains. The conceptual design process is defined as an integrated system with five stages and four mappings and mathematical descriptions are applied as input for an expert system. A similar approach is applied by Tay & Gu (Tay & Gu, 2002). AD is applied to derive the hierarchical topology of the design from the functional and physical domains. The thus obtained primitives are inputted into a relational data model. The work of Chen et al. (Chen et al., 2012) expands this method with a production framework. The method stays in the conceptual phase. Deng et al. (Deng et al., 2000) also have a similar approach as Tay & Gu. However, this work does not use the AD methodology but instead of this, a self-defined framework called 'Functional Design Model' is applied. The architecture framework for manufacturing system design of Benkamoun et al. (Benkamoun et al., 2014) also uses the axiomatic domains and the hierarchical structure. The framework applies IDEF0 to define relations between the domains. Knowledge about process and configuration is stored in the framework and can be reapplied when the system needs to be reconfigured. Zhang and Chu have developed an interesting approach for the design of product and maintenance by combining AD, QFD and FMEA (Zhang & Chu, 2010). Knowledge and applied-knowledge are combined in a single model that gives a complete overview of their relations to indicate if parts are missing. Unfortunately, the model is only applied during the conceptual design phase and would need to be expanded for product development and production. Ulrich & Eppinger have broken down the process of concept development in seven stages (Ulrich & Eppinger, 2004). These stages are each again broken down in 4-7 steps, which provides an extensive amount of fairly simple steps to follow. However, this apparent simplification does not guarantee that this solves the complexity of the conceptual design stage. As reference, to check that the designer does not forget important issues, it can be useful. Komoto & Tomiyama (Komoto & Tomiyama, 2012) describe a product modelling framework called System Architecting CAD. SA-CAD tracks system decomposition, it models parameter relations, and performs consistency management of the parameters. An interesting aspect is that SA-CAD could eventually store design knowledge used in system architecting independently from specific engineering disciplines such as physical contacts to constrain the topology of a set of entities. The work of Benkamoun, Ulrich & Eppinger, and Komoto is particularly valuable for this research since they all add the capability of actively securing the knowledge content in the model itself or in the periphery of the model. Zhang's model does the same but additionally links this knowledge to the applied knowledge; the current appearance of the design itself, though it should be converted from the realm of maintenance to that of product design.

2.4.2 Phases of the Design Process

Many models divide the total product design process in two basic stages. The initial stage is the conceptual phase that ends with a proof of concept and the second stage is a product development stage that deals with realisation and test. The V-Model visualises this with its two legs; the left-hand leg handles conceptual design and the right-hand leg handles integration and testing. Other conceptual design methods that follow the standard V-Model, e.g. Komoto & Tomiyama (Komoto & Tomiyama, 2012) do the same. In AD, the Independence Axiom focusses on conceptual design and the Information

Axiom focusses on robustness of the design. The standard work of Pahl & Beitz (Pahl & Beitz, 2013) divides the design process in four stages: 'Definition', 'Conceptual', 'Embodiment', and 'Carryout'. Note that the conceptual phase ends with a number of alternative options and the embodiment stage ends with proof of principle of the design, so basically the conceptual phase is split into two stages.

Banathy describes in his theory 'Dynamics of Divergence and Convergence', shown in Figure 2.7, an iterative approach of diverging and converging cycles, respectively focusing on the 'Image of the future system' and the 'Model of the future system' (Banathy, 1996).

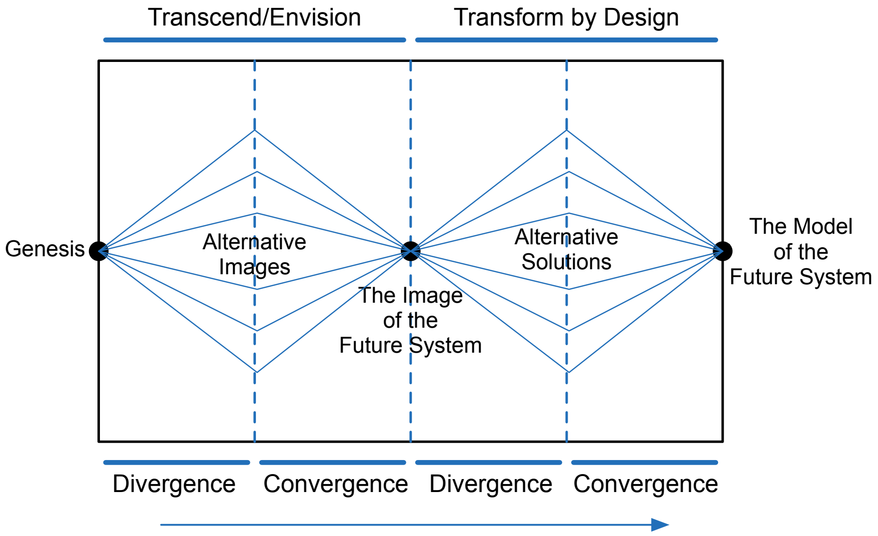


Figure 2.7 *The 'Theory Dynamics of Divergence & Convergence' splits the conceptual phase in two stages*

This is comparable to the approach of Pahl & Beitz, dividing the conceptual phase in two parts. Wang et al. (Wang et al., 2002) also split the conceptual phase in two stages by defining a mapping stage for fuzzy customer requirements to functional specifications, and a development stage for multiple design solutions. This is not conforming the AD methodology where this could be done simultaneously by the application of concurrent design. As explained in Subsection 2.4.1, the approach of Ulrich & Eppinger is quite detailed. However, a more profound look learns that this approach also sets target specifications, analogue to Banathy's image of the future system, and subsequently it reduces the number of alternatives to a final concept. This makes the approach on a par with Pahl & Beitz and Banathy.

PART 1: MODELLING

CHAPTER 3

CONCURRENT MICROSYSTEM DEVELOPMENT: FRAMEWORK AND RATIONALE[*]

3.1 Introduction

The current requirement of reducing lead-times, necessary for effective development of micro products and their manufacturing systems, is a challenge which requires a novel and innovative approach. The capabilities to take into consideration are engineering changes and testing ('Verification and Validation', V&V). Therefore, a central theme of the proposed methodology is to introduce and implement iterative feedback loops within a new product development framework. The proposed *Micro System Development Framework* is shown in Figure 3.1.

---

[*] *Parts of this chapter were published in:*

Puik, E. C. N., Telgen, D., Ceglarek, D., & Moergestel, L., van. (2013). Structured Analysis of Reconfigurable Manufacturing Systems. In A. Azevedo (Ed.). Presented at the 23th International Conference Flexible Automation and Intelligent Manufacturing FAIM2013, Porto, (Puik et al., 2013c).

Puik, E. C. N., Telgen, D., Ceglarek, D., & Moergestel, L., van. (2013). Qualitative product/process modelling for reconfigurable manufacturing systems. 2013 IEEE International Symposium on Assembly and Manufacturing (ISAM) (pp. 214–218). IEEE., (Puik et al., 2013b).

Puik, E. C. N., Telgen, D., & Moergestel, L., van. (2013). Structured Analysis of Reconfigurable Manufacturing Systems. 23rd International Conference on Flexible Automation and Intelligent Manufacturing (FAIM), (Puik et al., 2013c).

Puik, E. C. N., Gielen, P., Telgen, D., Moergestel, L., van, & Ceglarek, D. (2014). A Generic Systems Engineering Method for Concurrent Development of Products and Manufacturing Equipment (Vol. 435, pp. 139–146). Berlin, Heidelberg: Springer Berlin Heidelberg, (Puik et al., 2014a).

The Proposed *Micro System Development Framework* (further referred to as *μSD framework)* adopts systems engineering models, initially created for large software development projects, and enhances them by creating a framework with capabilities to: (i) embed flexible and interchangeable functional 'Objects' (represented by blocks in Figure 3.1), and (ii) integrate them by using a novel procedure for assessment of quality using feedback control loops.

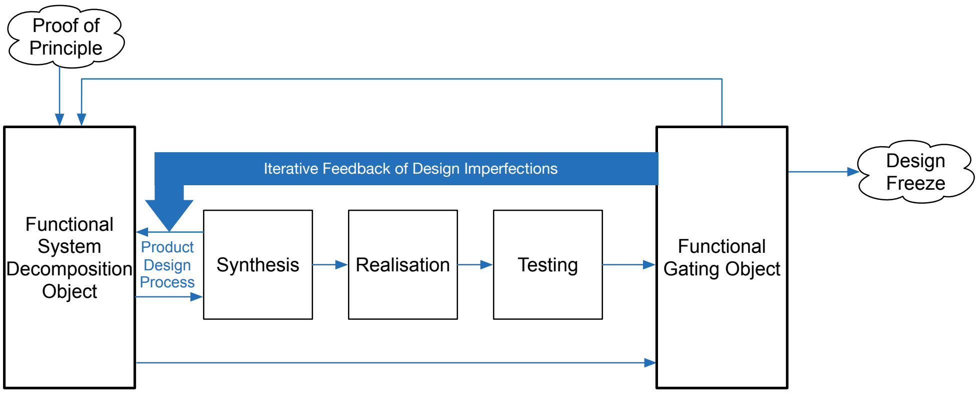


Figure 3.1 *The proposed Micro System Development Framework*

Chapter 3 presents the overall *μSD framework* with: (i) focus on implementation of the feedback control procedure, (ii) a modular structure in which the Objects *functional system decomposition* and *functional gating* that may be adapted to required capabilities of the method, and (iii) examples how state-of-the-art system engineering methods can be applied to provide solutions for the key Objects *functional system decomposition* and *functional gating*. Topic (i) will be explained by comparing the *μSD framework* with the traditional product development procedure when the V-Model is applied. Topic (ii) and (iii) will be explained by initial application of state-of-the art systems engineering approaches. To illustrate the modular capability of the Object *functional system decomposition*, 'Structured Analysis Design Technique' (SADT), and 'Failure Modes and Effect Analysis' (FMEA) are applied. To illustrate the modular capability of the

Object *functional gating*, 'Qualitative Analysis' (QA) is applied in two ways; initially with a unidirectional coding scheme followed by an enhanced approach using a bi-directional coding scheme called 'Maturity Grid-based Qualitative Analysis'. An overview of the approaches used and developed as part of the SD framework is shown in Table 3.1.

Table 3.1 *The concurrent microsystem development μSD framework – presentation outline*

Object / Chapters	*Functional system decomposition*	*Functional gating*
Chapter 3	Structured Analysis Design Technique (SADT) in combination with Failure Modes and Effect Analysis (FMEA)	Three different methods: (i) Remaining uncertainties determined with FMEA (ii-a) Qualitative Analysis (QA) with unidirectional coding (ii-b) Qualitative Analysis using a Maturity Grid (MG) with a bi-directional coding scheme
Chapter 4	*Concurrent system decomposition* based on Axiomatic Design	*Concurrent gating* based on the completion of decomposed hierarchical levels (or tested levels)
Chapter 5	Same as in Chapter 4	*Intelligent gating* based on Information in Design

The presented *μSD framework*, with its capabilities for iterative and modularly implemented feedback loops, provides two necessary enablers (*functional system decomposition* and *functional gating*) that serve as a model for further enhancement into a concurrent microsystem development approach. Chapters 3, 4 and 5 present solutions for these enablers.

This chapter is organised as follows: Section 3.2 analyses the current situation and defines the key limitations. Section 3.3 explains the methodology of the proposed *μSD framework* and shows the ways to embed the framework into project execution by application of the underlined *μSD rationale* (3.3.3 and 3.3.4). Section 3.4 illustrates the methodology using three industrial case studies in which the *μSD rationale* is applied and tested. Section 3.5, discusses the findings and lessons learned. Finally, Section 3.6 draws conclusions.

3.2 Analysis and Approach of Microsystem Development

This section analyses the current industrial practice: (i) the problem definition is elaborated in Subsection 3.2.1, (ii) current industrial practices are inventoried in Subsection 3.2.2, and (iii) the key limitations are discussed in Subsection 3.2.3.

3.2.1 Problem Definition

The process of microsystem development currently encounters tremendous challenges due to rapid development of emerging new pieces of technology, which need to be integrated into a final microsystem product within shorter and shorter lead times. These challenges draw substantial resources during the process of microsystem development. It creates high pressure to keep up with the emerging new pieces of technology, leaving limited resources for the integration and V&V phases of the project. Therefore, microsystem development projects are organised to focus on having a mix of novel and reused pieces of technology, to carefully dose the amount of novel technology to match the development capability and ambition of the project. This scenario leads to integration of various product and process technologies being at various maturity levels (Technology Readiness Levels; TRL). This varying maturity of individual pieces of technology brings varying uncertainties, which need to be addressed during project execution. On one hand, development of novel technology needs to be accelerated in order to catch up their development lag compared to reused and mature technology. On the other hand, application of mature technology needs to be monitored for correct reuse.

This is one of the underlined causes that many microsystem development projects are research-intensive with strong need for having: (i) flexible and concurrent *system decomposition* to support plug-and-play of various technology objects, (ii) flexible and reliable *functional gating* of various system configurations at a broad spectrum of TRLs, (iii) short response times by iterative feedback (improvement) cycles between these objects, and (iv) a simple but clear information flow in the design process. The *μSD framework* needs to demonstrate capability and efficiency under these requirements.

3.2.2 Current Industrial Practice

Currently, a number of system engineering methods and process design models are used to improve the operational structure of projects, i.e.: to prevent relevant issues of being overlooked, and to assure that these issues are satisfactorily addressed. Examples of systems engineering methods that are generally used in industry include the Waterfall-Model and the V-Model (Höhn et al., 2008; Deuter, 2013).

Figure 3.2 shows the conceptual stages of the Waterfall-Model, these stages are basically the same as the stages in the left-hand leg of the V-Model. The bottom of Figure 3.2 shows the schematic process flow through the conceptual phase; from *proof of principle* it implements an iterative process till *proof of concept* is reached. Not shown in the V-Model nor Waterfall-Model is that product design consists from a highly dynamic interaction of functional system decomposition, basically analysis in the functional domain, and the process of synthesis, shown in the lower half of Figure 3.2.

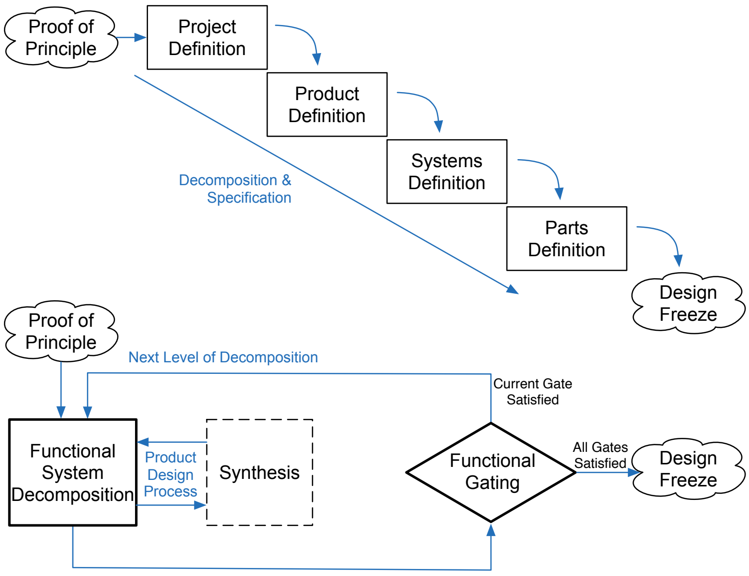


Figure 3.2 *Top: functional system decomposition and gating in the Waterfall-Model and V-Model. Bottom: iterative representation of the development process for these models*

Where decomposition is a diverging process that breaks the project down in small overseeable pieces, synthesis is a converging process that merges functional bits to a coherent whole (Roozenburg & Eekels, 1995). When functional performance goals have been coherently defined, the gating process enables the next hierarchical level. In the process, the whole product design may as such be decomposed and coherently specified till *proof of concept* is reached. The cycle, that is much alike Mesarovic's iconic model of the design process (as was explained in Chapter 2), has an iterative nature due to: (i) the highly dynamic interaction of decomposition and synthesis, and (ii) decomposition and gating functions are intermittently addressed. However, the iterations are either very dynamic, as is the case for option (i), or the iterations are very slow as is the case for option (ii) since the conceptual stage typically applies (only) a number of 4 gates.

Waterfall- and V-Models provide the capability to include three important elements of the microsystem development process: (i) planning, (ii) control, and (iii) structure. Planning and control are required capabilities for general project management (Burke, 2013). Structure is especially important in case of high-tech development projects that require additional time to develop core competences. Therefore, the focus during microsystems development is divided consciously between novel and reused parts of the design (Scott, 2001).

3.2.3 Key Limitations

The main shortcoming of the Waterfall-Model and the V-Model is their limitation in emphasising on functional verification during the development process; instead, these methods provide option for evaluation at the final and formal testing stages (further down in the Waterfall-Model, or right-hand leg of the V-Model). By then it is too late to make any significant changes to the design without substantial rework (Royce, 1970; Christie, 2008). The problem strongly affects developers and testers during the microsystem development process. It has also been observed that initially this problem might not be known or noticed by project managers and/or finance officers; as it simplifies the execution of the microsystem development project to a linear process, omitting frequent feedback loops. This shortcoming of the Waterfall-Model and the V-Model in lacking to provide iterative and intermediate feedback during the development process, and instead postponing it to the final and formal testing stages, leads to the following drawbacks:

- Lack of capabilities for intermediate assessment and testing of microsystem performance during the recursive process of functional decomposition and during the optimisation of the decomposed systems. Postponed testing leads to a time-lag between the occurrence of design errors and their detection. This frequently leads to subsystems that are not well-defined and as such the errors escalate as the design evolves, incurring unnecessary cost exponentially. The formal testing stages may be expected to reveal the error in a later stage, but even then, a relatively large part of the design needs to be redone;

- Results of separate design cycles remain in isolation, as they do not provide options to link them to the overall microsystem performance. Thus, intermediate design results cannot be explicitly used in making consecutive development decisions.

Another limitation is the fact that the Waterfall-Model and the V-Model, though they advocate decomposition and specification in the early stages, only provide rough guidelines on how decomposition and specification should be executed in the early stages. Without exhaustive decomposition, testing cannot be exhaustive either, as the decomposition tree of the design will act as a basis for testing during V&V. As a result:

- Poor specification leads to an incomplete system decomposition tree and, as a result, to poor functional specifications that lead to incomplete testing schemes. Incomplete testing can lead to errors remaining undiscovered during the testing stages. Problems are expected to reveal themselves at a later moment in time when the product may be released;
- As the Waterfall-Model and the V-Model were initially intended for software systems, they are not optimised for concurrent development of (physical) products and related manufacturing systems. Their capabilities are not scalable for the concurrent design of microsystems. This missing capability is developed and presented in Chapter 4.

3.3 Methodology for Implementation of the µSD Framework

This section describes the proposed *µSD framework* with iterative feedback loop. The section is organised as follows: (i) description of the functional elements and the information flow through the framework in Subsection 3.3.1, (ii) how the *µSD framework* addresses the key limitations in Subsection 3.3.2, and (iii) how state-of-the-art methods will be embedded for *functional system decomposition* and *functional gating* in Subsections 3.3.3 and 3.3.4 (as was explained in Section 3.1, respectively applying FMEA and SADT for *functional system decomposition* and mono and bi-directional QA for *functional gating*).

3.3.1 Explanation of the Objects in the µSD Framework

The *µSD framework* was yet only briefly proposed in the introduction of this Chapter, though it was explained that the Objects *functional system decomposition* and *functional gating* are central themes of this thesis.

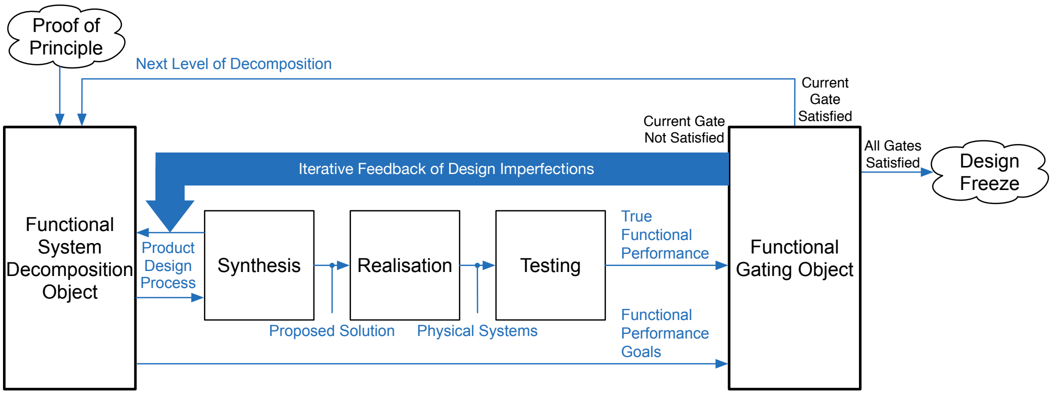


Figure 3.3 *Information flow in the µSD framework*

The *µSD framework* starts with a proof of principle, shown in Figure 3.3, which will develop into the targeted microsystem. Proof of principle forms the basis for the design process that is composed from two Objects: *functional system decomposition* and 'Synthesis'. These objects are defined as follows:

- *Functional system decomposition* is an activity of analysis that breaks a complex problem into smaller parts that can be overseen by the designer. As it concerns 'Functional' decomposition, this analysis takes place in the functional domain, i.e., the performance of the design is considered for its functional behaviour (what should the system do) (Suh, 1990). As such, complete and matching functional requirements are needed to complete successive levels of decomposition; *functional system decomposition* cannot be performed without *functional specification*. *Functional system decomposition* will be illustrated with state-of-the-art approaches in Subsection 3.3.3 and a new concurrent approach is demonstrated in Chapter 4;
- *Synthesis* has the opposite characteristic of decomposition; instead of breaking down the complexity of the design problem, it integrates parts to form solutions for the design problems. Roozenburg and Eekels describe synthesis as 'thinking up a provisional design' (Roozenburg & Eekels, 1995). This implies that the integration does not take place in the functional domain (what does the system do) but instead in the physical domain (what does the system look like). *Synthesis*, as applied in this model, therefore causes a domain change, i.e., it realises functional goals and proposes a solution for realisation. *Synthesis* is not discussed in details as it is not the main focus and contribution of this thesis; however, the *synthesis* is extensively used in the presented industrial case studies.

The product design process consists of rapid interactions of *decomposition* and *synthesis* causing diverging and converging activities, simultaneously weighing the various design options for successful design. Successful design can be described as reaching a set of solutions which satisfy the pre-determined functional performance goals from the decomposed function-tree (Suh, 2001). As the design progresses, small uncertainties in the design will accumulate, as many successive design decisions, that are not absolutely certain, increase the probability that something will not exactly work out as expected. It is a role of the designer to determine the acceptable level of uncertainty in the design, and at that point the developments need to be stopped to mitigate these design performance uncertainties. At this point, the designer will want to test one or more intermediate solutions to eliminate or reduce the cumulative uncertainties as related to the functional performance goals (Figure 3.3). The proposed (intermediate) design solution is realised in the Object called 'Realisation' and then tested as part of the Object 'Test':

- *Realisation* is the function of transforming a proposed intermediate or final design solution to a physical system, often called a prototype, that can be tested;
- *Test* is a set of physical or computer experiments to determine the *true functional performance* of an intermediate or final product solution as predetermined by the *functional performance goals* (Figure 3.3). As such, *test* executes a reverse domain change as compared to design *synthesis*. It expresses physical action in terms of functional behaviour without imposing a value judgement. An overview on methods to perform testing was described by Maropoulos and Ceglarek (Maropoulos & Ceglarek, 2010).

The process of evaluation of test results is conducted as part of the Object *functional gating*:

- *Functional gating* provides a necessary link by closing the feedback loops between the Objects *realisation & test* and the Objects *decomposition & synthesis*,

> which are executed iteratively in the development process. Feedback provided by *functional gating* is based on: (i) evaluation of test results or *true functional performance* by comparing them to the predetermined *functional performance goals* (Figure 3.3), followed by (ii) a supporting decision-making process to determine necessary steps for the next development cycle. This is done by assessing functional behaviour of the system based on two criteria: (ii-a) satisfaction of specified limits for targeted functional requirements at the current level of decomposition, and (ii-b) satisfaction of the decomposition level of the design (the increasing granularity level of hierarchical decomposition, successively decomposing final product, major and minor subassemblies, up to the individual parts). If criterion (ii-a) is satisfied, a gate is closed and the design advances to the next more granular level of decomposition (and start next development cycle). If criterion (ii-b) is satisfied, all levels of decomposition are completed, all functional performance goals are met, iterations stop, and the development process is completed (Design Freeze as shown in Figure 3.3). In all other situations, the development cycles continue with successive iterations.

The comparison of true and intended functional performance that takes place at *functional gating* is essential in providing correct feedback (Figure 3.3). The expected *functional performance goals* are an output of *functional system decomposition*, represented as a functional model which will explain what the system is expected to do. The true functional performance that is an output of tested prototypes, provides understanding what the system actually does.

3.3.2 How the µSD Framework Addresses the Key Limitations

As discussed in Section 3.2.3, the main shortcoming of the state-of-the-art approaches such as Waterfall-Model and V-Model is that they do not provide iterative and intermediate feedback about the status of the development process before the final and formal testing stage. This shortcoming is addressed in the proposed *µSD framework* as the framework allows the iterative approach which starts from the beginning of the project and it compares at every cycle the *functional performance goals* of the microsystem design with its *true functional performance*. This comparison provides early and iterative feedback on the following issues:

- Where outcomes differ, actual understanding of the product design may be insufficient and can be increased by additional investigations. As such, understanding of the product design evolves;
- Feedback is available on short notice and results of design cycles can be directly linked to overall microsystem performance. Development builds further on intermediate design results and the thus obtained knowledge can be applied for making consecutive development decisions;
- When applying the *µSD framework*, the designer is challenged to apply *functional system decomposition* and derive the products functional requirements at various hierarchical levels. Comparison of *true functional performance* and *functional performance goals*, as is done by the Object *functional gating*, gives feedback to the designer to consciously elaborate on the way the microsystems are decomposed and the way the functional requirements are organised. As a result, the quality of the functional requirements, and their decomposition can be improved;

- The designer has the freedom to choose various methods to implement functionality for *functional system decomposition* and *functional gating* as long as these methods foresee in the basic functionalities for these two objects as described earlier. This makes the *μSD framework* generally modular and flexible for various approaches and implementations.

3.3.3 μSD Rationale for Implementation of Functional System Decomposition Based on SADT and FMEA

This subsection presents application and modular capability of *functional system decomposition* in the *μSD framework* by using state-of-the-art approaches of Structured Analysis Design Technique (SADT) and Failure Modes and Effect Analysis (FMEA). Similarly, Subsection 3.3.4 will present application and modular capability of *functional gating* in the *μSD framework* by using: (i) proposed in this thesis, Qualitative Analysis (QA) based on a unidirectional coding scheme (Saldaña, 2012), and (ii) proposed in this thesis, Maturity Grid (MG) based on a bidirectional coding scheme. Successively, the aforementioned *μSD framework* rationales will be illustrated by conducting 3 industrial case studies in Section 3.4.

Figure 3.4. illustrates modular capability of the Object *functional system decomposition* by using two state-of-the-art approaches within the *μSD Framework*: (i) SADT as part of the ‘System Decomposition Kernel’; and, (ii) FMEA as part of the ‘Problem Identification and Prioritisation’.

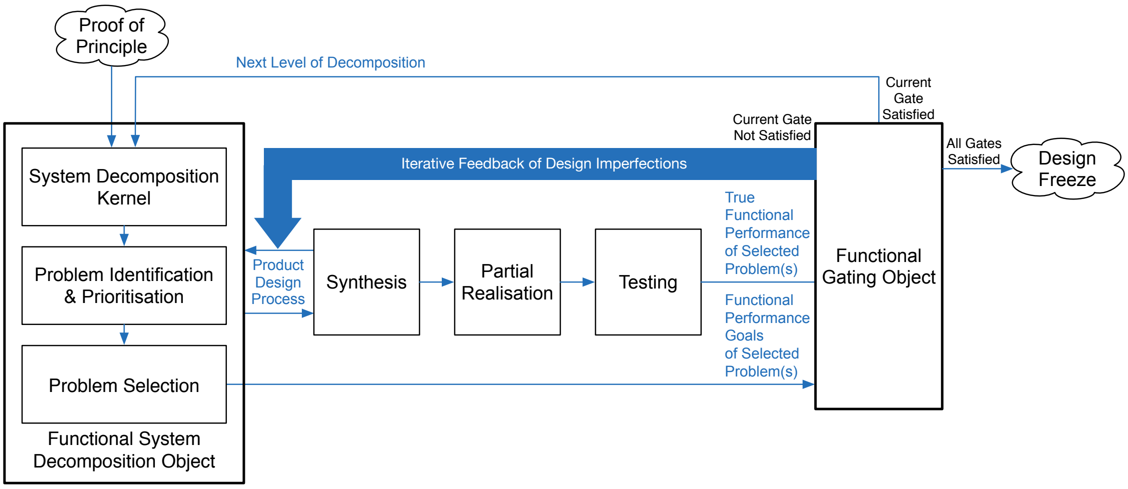


Figure 3.4 *Modular application of the μSD framework. Illustration of two state-of-the-art methods SADT & FMEA as part of functional system decomposition object*

The proposed *μSD Framework* can be applied with both systems engineering methods in the following way:

- *System decomposition kernel*: The proposed *μSD Framework* can be used with various state-of-the-art system decomposition kernels, such as SADT as illustrated here. Typically, the start of a new *μSD* project comes with a large amount of uncertainties that need to be taken into consideration. As a result, a *μSD* project cannot simultaneously be overseen by the designer as a whole. Therefore, requirements and development work need to be conducted sequentially. To enable the sequential approach, the microsystem is broken down into smaller parts that can be overseen by the designer. As the product is decomposed hierarchically into sub-systems, it is of importance that the critical interactions are clearly understood. As such, the microsystem is decomposed level by level until there are preferably no interactions between the sub-systems of a given level. In general, 'no interactions' means that subsystems can be specified and developed concurrently in separate tasks and separate contexts. All functional requirements are defined as system decomposition advances. After a while, a functional skeleton of the future product emerges.

It is possible to start with more than a single alternative when decomposing the functionality of the envisioned product; it provides the opportunity to select the better option later on.

Usually, four levels of decomposition are suitable to functionally describe the product as shown in Figure 3.5.

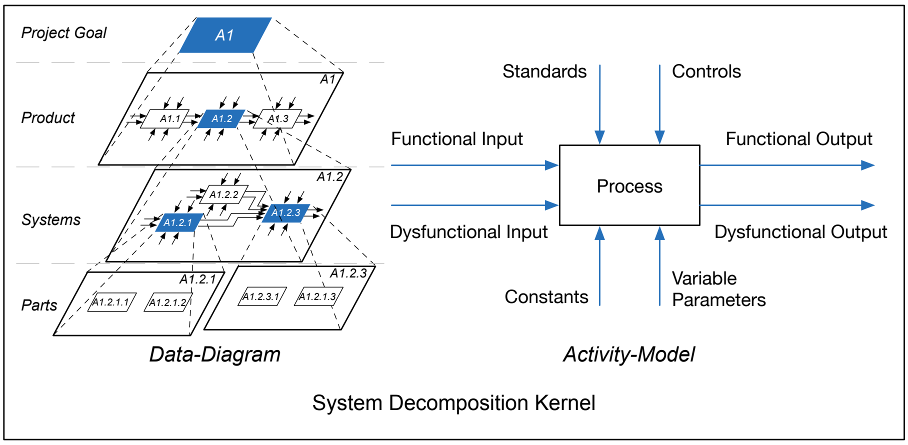


Figure 3.5 *System Decomposition Kernel Using SADT*

When applying SADT, a functional product flow or manufacturing process can be described in a layered structure. A top down decomposition in ‘Data-Diagrams’ is derived layer by layer (left-hand diagram of Figure 3.5). Basic functionalities are described using an ‘Activity-Model’ as shown in the right-hand diagram of Figure 3.5. The activity-model uses parameters to describe functionality of the particular function. Input parameters, can be ‘Functional’ or binding characteristics of a good product at start, or ‘Dysfunctional’ representing potential hazards or errors of the product before the particular process has even started. Conditional input parameters, like ‘Norms and Controls’ reflect boundary conditions or 'the demands' of the process. Parameters that are related to the transformation mechanism, comprising of 'Constants and Variables', are representing process or equipment characteristics. All input parameters serve as determinants for the output parameters, again *functional* or *dysfunctional*. Appendix A describes the SADT in more detail;

- *Problem identification and prioritisation*: The second step focuses on the approach for identification of most relevant problems. In this step, the method for FMEA is used for categorisation of the remaining uncertainties in the microsystem development project. At this point in the project, uncertainties arise as decomposition and requirement specification advance during the product development process (as was explained in Subsection 3.3.1). The uncertainties hinder the decomposition process, and at some point, they pile up which makes it difficult for the developer to have clear understanding of the project. It is no longer possible for the designer to understand the magnitude of uncertainties and their impact on the critical key characteristics (parameters) of the microsystem which is being developed. The uncertainties may lead to errors in the design and it is desired to address them. However, the consequences caused by the errors are depending on the initial uncertainty of: (i) the error happening, and (ii) the effect sorted by the particular error. Basic risk mitigation techniques may be applied to prevent potential consequences from happening, e.g.: (i) avoidance of risks by selecting a different design solution, (ii) Transfer of risks to another (external) party, more skilled in solving that particular problem, or (iii) graphical charting like Risk plotting, Pugh matrix, or Voting Methods (Garvey & Landsdowne, 1998). In this thesis is chosen to apply the FMEA for prioritisation of project uncertainties and their effects in order to plan one or more most suitable tests to address the uncertainties and their consequence, this is shown in Figure 3.6.

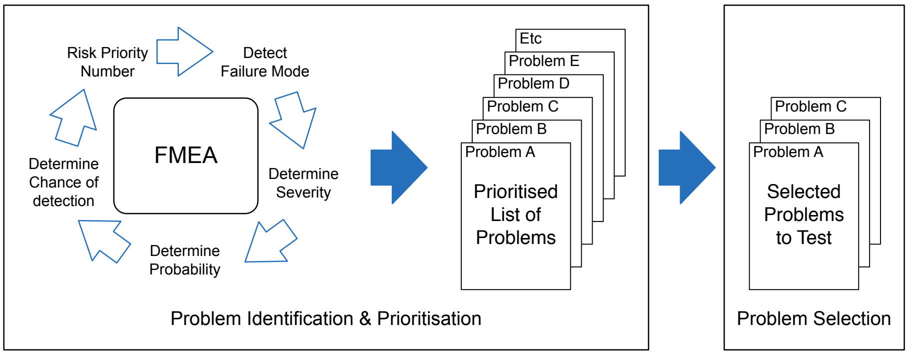


Figure 3.6 *Problem Identification & Prioritisation using FMEA followed by the Problem Selection process*

Application of the *problem identification & prioritisation* process is straightforward: (i) FMEA is applied to make a list of the uncertainties and respected effects, and (ii) that list is prioritised in order of project impact. The result is a prioritised list of project risks;

- *Problem Selection*: The third step is to make a selection of the problems from the prioritised list that was produced by the FMEA and then resolve them as part of the *iterative feedback loop* within next development cycle (right-hand part of Figure 3.6). The method is flexible to select single or multiple options to address in the next improvement cycle, but interpretation in the Object *functional gating* should be corresponding. For example, if a single problem is selected for optimisation in the next improvement loop, a single solution will be assessed in the Object *functional gating* which provides opportunities to focus the main resources of the project on in-depth resolving of a single issue. However, with much smaller breadth, i.e., not addressing the other potential problems and their interactions with the selected problem which also need to be resolved. On the other hand, in case of selecting multiple problems to be addressed concurrently, there is a good handle of breadth of the issues and their interactions; however, with given constant project resources strategy of addressing multiple problems might lead to shallow solution(s) for each individual problem. Typically, based on the author's experience, the spread or breadth of uncertainties to be addressed seems as a good guideline to start determining how many problems needs to be simultaneously investigated before addressing in-depth the most important ones. The breadth in selection of the problems allows selecting the most important one; and the in-depth strategy of the most significant ones provides the best resolution option for the selected problem(s). The in-depth analysis of the problem is conducted via (i) *system decomposition kernel* interaction with the *synthesis* object; and (ii) control via *functional gating* during iterative feedback loop development cycles.

It is important to notice that the source of uncertainties during the microsystem development is often caused by varying TRL levels of the addressed problems and their solutions. Therefore, the TRL level can also be applied as an index not only to prioritise the multiple problems to be addressed, but also the need for in-depth analysis.

3.3.4 μSD Rationale for Implementation of Functional Gating Based on Qualitative Analysis

This subsection focuses on illustrating role and functioning of the Object *functional gating* within the *μSD framework*. Overall, *functional gating* as proposed here works as a sequential evaluator of the problem-solution, i.e., evaluates the following aspects of the problem-solution: (i) causes of the problem(s) selected within the Object *functional system decomposition*, (ii) maturity of the suggested solution by the Object *synthesis* (e.g., the TRL level of the proposed technology to be used in the solution), and (iii) performance of the solution obtained within the *Realisation* and *Testing*.

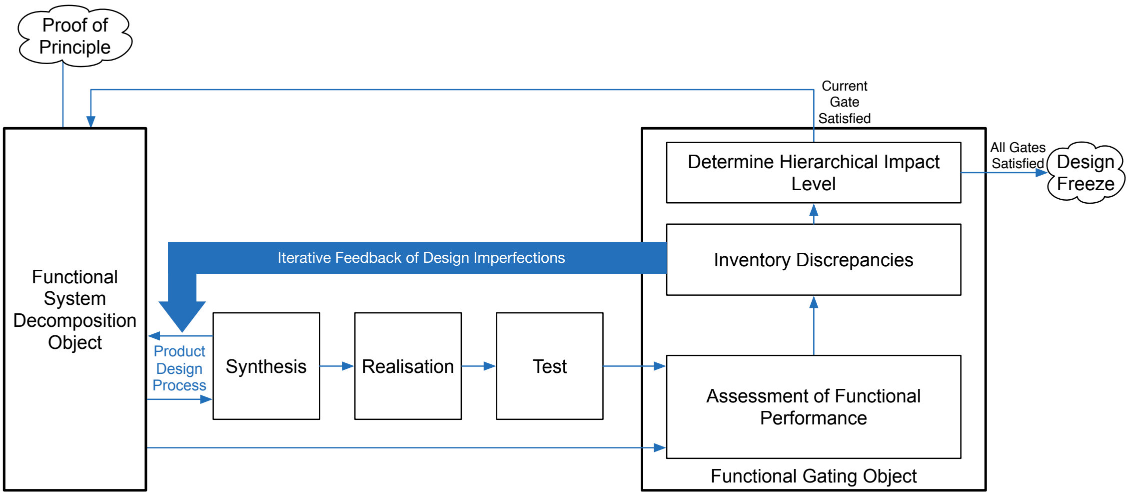


Figure 3.7 *Implementation of functional gating.*

Functional gating is explained by applying two systems engineering methods that can be implemented as a gating function: (i) state-of-the art approach to Qualitative Analysis (QA), and (ii) proposed in this dissertation Maturity Grid (MG). The outline and rationale for *functional gating* is shown in Figure 3.7 and detailed explanation is provided below in Figure 3.8. The initial step 'Assessment of Functional Performance' compares the *functional performance goals* and the *true functional performance*. The second step 'Inventory Discrepancies' drafts a list of the remaining problems in the product design and the impact of these problems. Depending of this outcome, the final step 'Determine Hierarchical Impact Level' is applied to decide if the current hierarchical design level may be advanced to the next more granular level of decomposition.

- *Assessment of Functional Performance*: The first step is based on the process of QA. This first example describes QA with a unidirectional coding scheme. The principle of 'coding' in the principle of system engineering is not to be confounded with software coding, but refers to arrangement of development tasks and problems in a systematic order for classification (Saldaña, 2012). Coding is applied to differentiate qualitatively between a set of attributes and/or parameters of the identified problems or solutions. A code can be a word or short phrase that symbolically assigns summative attributes for a situation to be assessed. The coding scheme applied for this case is: Cause known, Solution Known, Solution Tested, and Solution Successful. Figure 3.8 shows the three steps of the rationale for *functional gating* in more detail, the bottom rectangle representing *Assessment of functional Performance*.

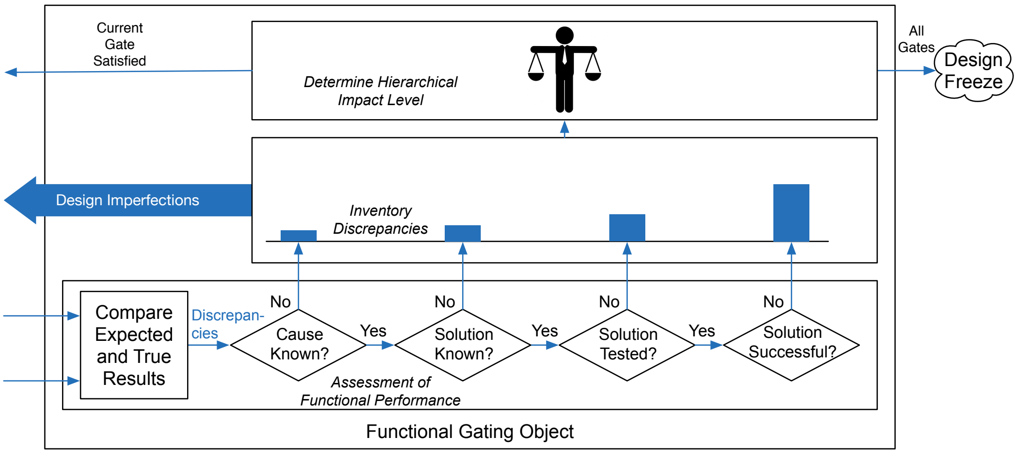
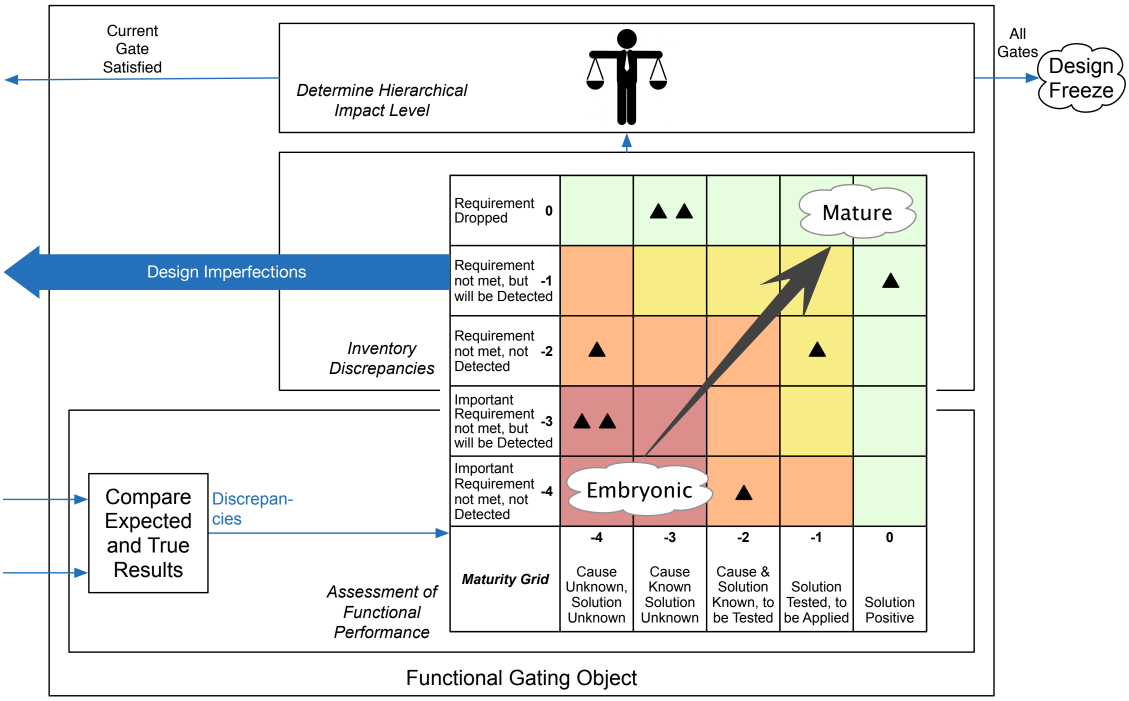


Figure 3.8 *Implementation of functional gating within the Object functional gating*

The procedure starts with a comparison of the *expected* and *true* outcomes of the *test object*, which provides the discrepancies between the actual performance of the tested system and what was expected by the designer. The discrepancies are weighted against the coding scheme. If the first summative attribute is not met, indicating that that imperfection is of no importance, the discrepancy is weighted against the second summative attribute that has a lower gravity, followed by the third and fourth attribute. In this situation, if the *cause* of a discrepancy is *not* *known*, the weighting process for that discrepancy stops and continues with the next discrepancy. If the *cause* indeed is *known,* the discrepancy is tested for a *known solution*, an imperfection that may be considered to have a lower gravity. This process continues till the actual gravity of the discrepancy is determined;

- *Inventory Discrepancies*: The second step is to inventory the discrepancies as determined by the process of QA, this is shown in the middle rectangle of Figure 3.8. In this principle, where a unidirectional coding scheme is applied, a basic histogram is prepared from the output of the QA. The histogram gathers the numbers of cases that coded attributes were rejected, providing a distribution of gravity of the observed discrepancies revealed by the *test object*. The acquired knowledge of the product design is brought back in the feedback loop to improve the process of *functional system decomposition* and *requirement specification*;
- *Determine Hierarchical Impact Level*: The third and final step is to determine the actual decomposition level of the project to: (i) indicate up to what level the *functional system decomposition* has been completed, and (ii) index the absolute status of project to determine project progression. The current hierarchical level of decomposition can be closed when there are no discrepancies and the histogram inventoried by the *inventory discrepancies object* remains empty.

In addition to QA with unidirectional coding, a second implementation shows analysis with bi-directional coding. Bi-directional coding is in literature also known as 'Risk Plotting' (Bullema et al., 1999). In a bi-directional coding scheme, two coding schemes are applied, independently from each other, to determine the risk of discrepancies between *expected* and *true* outcomes of the *test object*: (i) the primary coding scheme is applied to the level of understanding of the solution, and in this example, it is done by applying the same scheme as for unidirectional coding above, and (ii) a secondary coding scheme is applied to estimate the impact if a problem is not solved. Together, these coding schemes provide a measure for risks, not only taking into account *assessment of functional performance*, but also *inventory discrepancies*. Figure 3.9 shows the implementation of bi-directional coding, applying a bi-directional coded diagram that will further be referred to as 'Maturity Grid' (MG). The primary coding scheme is plotted on the horizontal axis of the MG, and the secondary coding scheme is plotted on the vertical axis.



Maturity Grid	-4 Cause Unknown, Solution Unknown	-3 Cause Known Solution Unknown	-2 Cause & Solution Known, to be Tested	-1 Solution Tested, to be Applied	0 Solution Positive
Requirement Dropped 0		▲ ▲		Mature	
Requirement not met, but will be Detected -1					▲
Requirement not met, not Detected -2	▲			▲	
Important Requirement not met, but will be Detected -3	▲ ▲				
Important Requirement not met, not Detected -4	Embryonic		▲		

Figure 3.9 Implementation of the Maturity Grid in the concept of functional gating

The MG has the capability to concurrently visualise the understanding and the severity of the identified problems in design (represented by triangular marks in Figure 3.9). When concurrent problems are visualised, the MG provides information about: (i) position of the identified problems on the Maturity Grid, and (ii) the scattering of the identified problems on the Maturity Grid. The position of the identified problems in the maturity Grid is used to enable the feedback-loop and trigger the action by the *functional gating*. On the other hand, the scattering of the identified problems on the Maturity Grid, provides feedback on design imperfections to be solved, and how to prioritise them. The feedback and prioritisation improves the developing strategy on how to accelerate the process of maturing the design. The MG is updated during each development cycle and thus it can provide an overview of the product maturity at any given time and provide almost continuous feedback on the product development process.

The colours (or grey tones) in the grid show the interpretation of the severity of the problem (S) based on the product of the scores (-4, -3, -2, -1, 0) on the two axes. The colours are defined as follows: 16 =< S =< 9 were marked as red (dark grey); 8 =< S =< 4 were marked as orange; 3 =< S =< 1 were marked as yellow; and (S = 0) were marked as green (light grey). When testing the use of the MG with designers, it was concluded that the MG can present clearer results for designers by using colours (or grey tones) to mark the severity of the identified problems.

3.4 Three Industrial Case Studies to Illustrate and Test the μSD Framework with Functional System Decomposition and Functional Gating

This section includes three industrial case studies, which are described in Table 3.2. The selected three cases share the same rationale for functional system decomposition. However, each case study implements a different approach for functional gating, in order of illustration its effectiveness and capacity to handle increasing complexity. All case studies use a multidisciplinary approach in which: (i) the product, (ii) the assembly process, and (iii) the assembly equipment are developed concurrently. The cases focus on the innovative aspects of development.

Table 3.2 *Applications of the μSD framework for the cases in successive sections*

Object Section	*Functional system decomposition*	*Functional gating*
Subsection 3.4.1; Assembly of Cell Phone Lenses	Structured Analysis Design Technique (SADT) in combination with Failure Modes and Effect Analysis (FMEA) as explained in Subsection 3.3.3	Project flow is based on remaining uncertainties as selected by the FMEA
Subsection 3.4.2; Assembly of Nanometre Measuring Probe	Same method as above	Qualitative Analysis (QA) based on a unidirectional coding scheme explained in Subsection 3.3.4
Subsection 3.4.3; Assembly of an Automotive Actuator	Same method as above	Maturity Grid (MG) based on a bi-directional coding scheme, also explained in Subsection 3.3.4

3.4.1 Case Study 1: Cell Phone Lens Assembly Development Using the μSD Framework with Functional System Decomposition Based on SADT and FMEA Approaches

Product: The lens-assembly aims to image environmental light to an optical sensor to enable the user of taking pictures. The lens parts consist of transparent and dark opaque parts to control the light rays through the lens-assembly. The diameter of the parts is 5-6 mm and expected to decrease to 2 mm in future designs. It is required that a reduction of the parts' diameters should be taken into consideration. The lens-parts need the capability to withstand temperature and humidity changes and shock within the operational specification of the cell phone device.

Figure 3.10 *Two types of cell phone lens assemblies*

Process: The case focuses on the assembly and joining of the parts of the lens stack. To manufacture the small cell phone lens assemblies, 4 to 6 cylindrical plastic parts (lenses and diaphragms) need to be stacked on top of each other to form a 'Lens-stack' (Figure 3.10). This case study introduces a new procedure to align the cylindrical parts. Instead of the state-of-the-art design that uses a small cylindrical housing called 'Lens Barrel' for alignment of the optical parts, the new design aligns optical parts by pressing parts into a 'V-groove', and the lens barrel is no longer needed. This new alignment process is expected to have better accuracy and capability to be used for a considerable smaller diameter of the lens assembly. However, this process needs design modifications and new manufacturing solutions. The required modification has a disruptive character from the current manufacturing perspective for: (i) the new V-groove alignment principle, and (ii) the process of adhesive bonding of the parts. At first, the V-groove alignment is executed by constraining the stack of lenses by two bodies that apply a light clamping force along the optical axes of the lenses. Initially, the lenses are roughly pre-aligned, and as a next step they need to be aligned to their final position as can be seen in Figure 3.11.

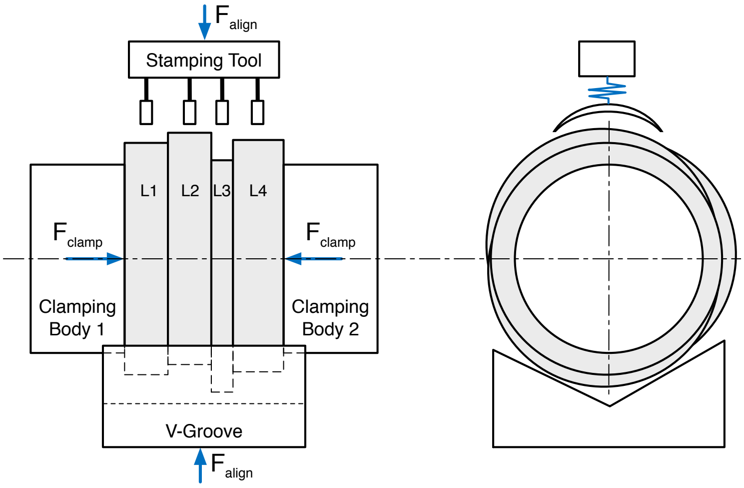


Figure 3.11 *Process P4 aligns the lens parts L1-L4 using two combinations of forces*

Secondly, the process of adhesive bonding is done by dispensing a low viscosity adhesive to the side of the lens-stack that adheres the microscopic gaps between the parts. Micrometre accuracies are achieved when the process is performed well, depending on geometry, surface condition, dust particles, and triboelectric charge of the parts.

The assembly of lens-stacks was planned to be executed using a Reconfigurable Manufacturing System. The company uses a framework of reconfigurable, reusable, and modular process equipment for micro assembly, e.g., robotic manipulators, feeders, grippers, glue dispensers, and curing solutions. The reconfigurable manufacturing framework may be applied to compose a manufacturing system with a new layout from existing *process modules*. If needed, new *process modules* can be developed or improved.

Functional system decomposition – system decomposition kernel: To structure the product development, functionality of: (i) product design, (ii) manufacturing processes, and (iii) assembly system were decomposed, (i) and (ii) using a standard decomposition tree as show in Figure 3.12, and (iii) since assembly is a sequential process, using SADT as shown in Figure 3.13.

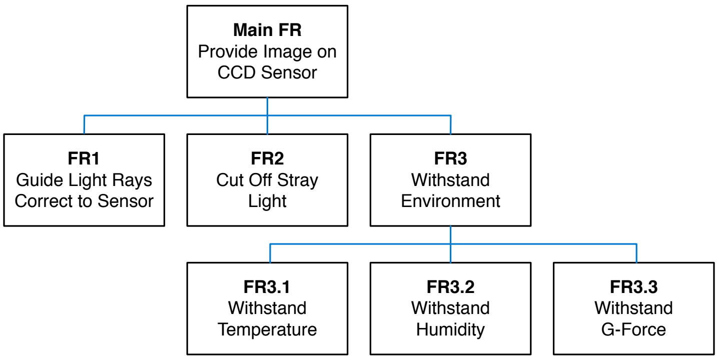


Figure 3.12 *Functional System Decomposition tree of product design (left) and manufacturing process (right)*

The functional decomposition trees are realised in interaction with synthesis of the product. Parallel to the decomposition process, the product is designed. As the design process evolves, the adhesive bonding process was developed concurrently. Sequential development of product design and process technology, for this system, is not an option as the parts of the lens-stack need optimisation for the adhesive bonding process. Changes in the design of the lens-stacks were carried through to enable the bonding process, e.g., sharp corners in the lens design were added to prevent the low viscosity glue from reaching and polluting the lens surface.

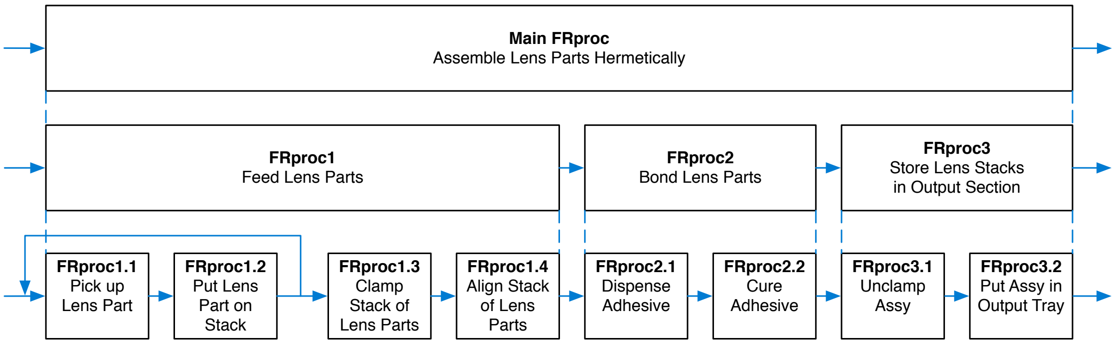
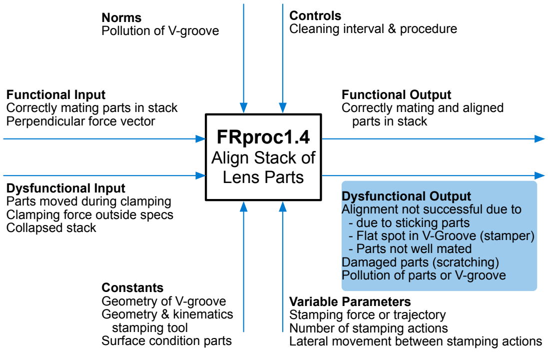


Figure 3.13 *Functional System Decomposition tree of manufacturing process using SADT*

Concurrently with the decomposition of the product design and adhesive bonding process, the layout for the RMS was determined and the assembly process was decomposed. Figure 3.13 shows the process flow (SADT Data Diagram). All processes were analysed in detail. In this chapter, the most critical process ‘FRproc1.4’: ’Align Stack of Lens Parts’ is emphasised. Process FRproc1.4 includes the new alignment procedure by application of the V-groove, and it was recognised as the process with the highest risk at the start of the development process.

Functional System Decomposition

System Decomposition Kernel



Problem Identification & Prioritisation
- FRproc1.4-1 — Alignment not successful due to sticking parts
- FRproc1.4-2 — Flat spot in V-groove or stamping tool
- FRproc1.4-3 — Parts not well mated
- FRproc1.4-4 — Scratched optical parts
- FRproc1.4-5 — Pollution optical parts
- FRproc1.4-6 — Pollution V-Groove

FMEA of Identified Problems

Problem	Chance	impact	Result	Priority
FRproc1.4-1	25%	10	2,5	1
FRproc1.4-2	10%	6	0,6	4
FRproc1.4-3	20%	8	1,6	2
FRproc1.4-4	10%	8	0,8	3
FRproc1.4-5	10%	6	0,6	4
FRproc1.4-6	10%	6	0,6	4

FMEA Prioritised Problems

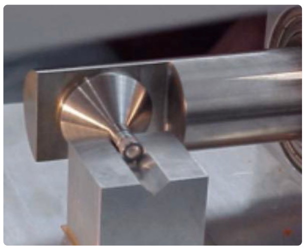
Problem	Chance	impact	Result	Priority
FRproc1.4-1	25%	10	2,5	1
FRproc1.4-3	20%	8	1,6	2
FRproc1.4-4	10%	8	0,8	3
FRproc1.4-2	10%	6	0,6	4
FRproc1.4-5	10%	6	0,6	4
FRproc1.4-6	10%	6	0,6	4

Problem Selection (to be addressed in following cycle)
- FRproc1.4-1 — Alignment not successful due to sticking parts
- FRproc1.4-3 — Parts not well mated

Realisation & Test

Realisation
- FRproc1.4-1 — Lens-Parts were produced with preliminary production tools
- FRproc1.4-3 — Laboratory test setup was made for manual assembly and alignment

Test
- FRproc1.4-1 — Geometrical test were performed to quantify mechanical alignment performance of the lens-stack
- FRproc1.4-3 — Optical tests were determined to quantify optical performance of lens-stack
- FRproc1.4-3 — Shock, Temperature & Humidity test were performed by customer



Functional Gating

Assessment of Quality
- FRproc1.4-1— Low friction, low stick slip. Smooth alignment of lens-stack — minimal discrepancy
- FRproc1.4-3 — Alignment successful. Lens stack assemblies perform significantly better than current standards — minimal discrepancy

Design Imperfections were coupled back to the designers

Inventory Discrepancies
- *No significant discrepancies*

Determine Hierarchical Impact Level
- Though the selected problems were addressed, not all problems were addressed yet. Therefore, the decomposition level was not considered to be completed yet.

Figure 3.14 *Application of the µSD framework for lens assembly development*

The input of FRproc1.4 is formed by a stack of lenses that have been placed in the V-Groove with limited accuracy of the robotic manipulator. The output should be an aligned stack. Figure 3.14 shows the consecutive stages of *functional system decomposition*. The system decomposition kernel adds the SADT *activity model* (as explained in Figure 3.5). The steps of the *μSD rationale* were executed as follows:

- *Functional system decomposition – problem identification & prioritisation:* The output of the SADT *activity model* delivers ‘Functional Output’; ‘what should go right’, and ‘Dysfunctional Output’; ‘what can go wrong’ (more details on the SADT activity model in Appendix A). The dysfunctional output forms the basis for the *identification process* of problems in the design, shown with problem definition FRproc1.4-1 to FRproc1.4-6 in Figure 3.14. The FMEA is applied to weigh the risks as defined by the SADT and then the list is prioritised;
- *Functional system decomposition – problem selection:* finally, a selection is made by the engineers of what seems feasible to investigate in a single iteration. Two problems were selected for the first iteration cycle (FRproc1.4-1 and FRproc1.4-3). The selection process concludes *functional system decomposition* for the initial cycle.

The results of the Objects *realisation* and *test* are shown in the second column of Figure 3.14. Lens parts were produced using preliminary production tools and assembled by a laboratory test tool in which parts were manually inserted but the alignment principle was automated using a robotic manipulator. The parts were then tested for geometrical accuracy, optical accuracy, and permanence for variations in temperature and humidity. Next, the steps for *functional gating* were applied as follows:

- *Functional Gating – Assessment:* Both problems FRproc1.4-1 and FRproc1.4-3 were successfully addressed and provided a fully functional production process. Little discrepancies compared to the expectations were noticeable. The test was considered a success, *design imperfections* and lessons learned were coupled back to the designers;
- *Functional Gating – Inventory Discrepancies:* Since no substantial discrepancies are noticeable, the histogram for this case is empty;
- *Functional Gating – Determine Hierarchical Impact Level:* Though the selected problems were addressed well, not all initial problems were investigated. Therefore, the decomposition level was not considered to be completed yet.

Two more iterative cycles were executed to complete the product and its process. These cycles, the second and third cycles, are shown in Figure 3.15. In the second improvement cycle, risk FRproc1.4.2 was addressed together with two next risks of the prioritised list; FRproc1.4.4 and FRproc1.4.5. The tests were still executed with a test setup and all three problems could be reduced and verified for functional performance (better gripper, cleaner environment, improved geometry of the stamping tool). In the third and final cycle, the RMS was actually configured. It was largely composed from standard *process modules*. The newly developed process module for alignment, FRproc1.4, was engineered but the mechanical core of the test setup was maintained. Due to the synergy, test setup and final system were functionally comparable and no perceptible functionality was lost.

Functional System Decomposition

Problem Identification & Prioritisation

- FRproc1.4-1 — Alignment not successful due to sticking parts
- FRproc1.4-2 — Flat spot in V-groove or stamping tool
- FRproc1.4-3 — Parts not well mated
- FRproc1.4-4 — Scratched optical parts
- FRproc1.4-5 — Pollution optical parts
- FRproc1.4-6 — Pollution V-Groove

FMEA of Identified Problems

Problem	Chance	impact	Result	Priority
FRproc1.4-1	5%	10	0,5	3
FRproc1.4-2	10%	6	0,6	2
FRproc1.4-3	5%	8	0,4	4
FRproc1.4-4	10%	8	0,8	1
FRproc1.4-5	10%	6	0,6	2
FRproc1.4-6	10%	6	0,6	2

FMEA Prioritised Problems

Problem	Chance	impact	Result	Priority
FRproc1.4-4	10%	8	0,8	1
FRproc1.4-2	10%	6	0,6	2
FRproc1.4-5	10%	6	0,6	2
FRproc1.4-6	10%	6	0,6	2
FRproc1.4-1	5%	10	0,5	3
FRproc1.4-3	5%	8	0,4	4

Problem Selection (to be addressed in following cycle)

- FRproc1.4-4 — Scratched optical parts
- FRproc1.4-2 — Flat spot in V-groove or stamping tool
- FRproc1.4-5 — Pollution optical parts

Realisation & Test

Realisation

- FRproc1.4-4 — Apply optimised (hollow) gripper to increase pre-alignment of parts. This prevents gripper from touching optical surface of lenses
- FRproc1.4-2 — Circular shape of stamping tool
- FRproc1.4-5 — Test in cleanroom environment

Test

- FRproc1.4-4 & 1.4.5 — Produced lens-stacks were investigated for particles under microscope
- FRproc1.4-2 — Parts were rotated to investigate tolerant angle for successful positioning

Functional Gating

Assessment of Quality

- FRproc1.4-4 & 1.4.5 — No further damage & pollution detected — minimal discrepancy
- FRproc1.4-2; Large tolerance on position V-groove without consequences positioning accuracy lens parts — minimal discrepancy

Design Imperfections were coupled back to the designers

Inventory Discrepancies

- No significant discrepancies

Determine Hierarchical Impact Level

- Though the selected problems were addressed, not all problems were addressed yet. Therefore, the decomposition level was not considered to be completed yet.

Functional System Decomposition

Problem Identification & Prioritisation

- FRproc1.4-1 — Alignment not successful due to sticking parts
- FRproc1.4-2 — Flat spot in V-groove or stamping tool
- FRproc1.4-3 — Parts not well mated
- FRproc1.4-4 — Scratched optical parts
- FRproc1.4-5 — Pollution optical parts
- FRproc1.4-6 — Pollution V-Groove

FMEA of Identified Problems

Problem	Chance	impact	Result	Priority
FRproc1.4-1	5%	10	0,5	2
FRproc1.4-2	3%	6	0,18	5
FRproc1.4-3	5%	8	0,4	3
FRproc1.4-4	1%	8	0,08	6
FRproc1.4-5	5%	6	0,3	4
FRproc1.4-6	10%	6	0,6	1

FMEA Prioritised Problems

Problem	Chance	impact	Result	Priority
FRproc1.4-6	10%	6	0,6	1
FRproc1.4-1	5%	10	0,5	2
FRproc1.4-3	5%	8	0,4	3
FRproc1.4-5	5%	6	0,3	4
FRproc1.4-2	3%	6	0,18	5
FRproc1.4-4	1%	8	0,08	6

Problem Selection (to be addressed in following cycle)

- FRproc1.4-6 — Pollution V-Groove

Realisation & Test

Realisation

- FRproc1.4-6 — Extra suction of clean air to bottom of V-groove. Robotic manipulator for fully automated handling of parts and alignment of stack

Test

- FRproc1.4-6 — Produced lens-stacks were investigated for particles under microscope
- V-groove stays clean after test run of 500 pcs
- Improved pre-aligment, no scratching surfaces
- Good alignment. Parts mating well
- No problems with flat spot
- No particles found

Functional Gating

Assessment of Quality

- FRproc1.4.6 — No particles detected — minimal discrepancy

Inventory Discrepancies

- No significant discrepancies

Determine Hierarchical Impact Level

- All problems were addressed. The decomposition level was considered to be completed. Since FRproc1.4 was completely operational, proof of concept for this process step was satisfied.

Figure 3.15 *Results of the second (column left) and third development cycle (column right)*

3.4.2 Case Study 2: 3D Measuring Probe Assembly Development using the µSD Framework with Functional Gating Based on Unidirectional Qualitative Analysis

3D Measuring Probes are made for geometrical measuring of high-tech products with high accuracy and small tolerances. The probes are applied in 'Coordinate Measuring Machines' to enable measuring in the Nanometre range. The heart of the probe consists of a micro-machined 'Silicon Die' that contains sensitive strain gauges to convert mechanical nanometre-displacements to electrical signals. The black, rectangular Si Die on the aluminium substrate (Figure 3.16)-left) is interfaced with the outside world using a flexible PCB connected by wire-bonding (Figure 3.16-right).

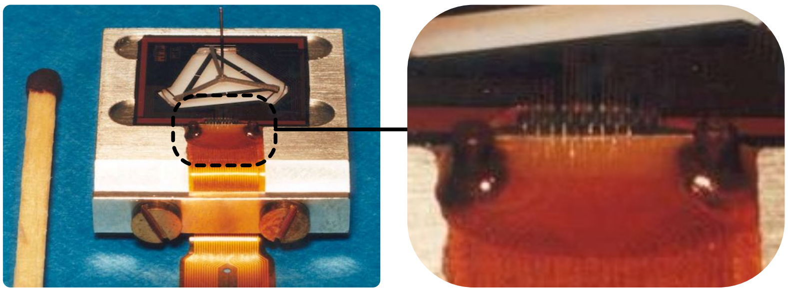


Figure 3.16 *3D probe for high-accuracy geometrical measurements*

When the product is designed and the project is at the 'systems'-level of *functional system decomposition*, the engineers decide to produce a number of engineering prototypes to validate manufacturability of the device. The functional Process Design Requirements of the 3D measuring probe are decomposed using SADT which gives an overview of the assembly process of the device (Figure 3.17). The total manufacturing process consists of a number of seventeen stages that may be partially executed in parallel for the 'Base' of the probe and the 'Stylus' that are assembled in the final assembly stages. The last stage of the base-assembly process is monitored in this case. It executes the wire-bonding process. The *µSD framework* is applied for the development of the 3D measuring probe:

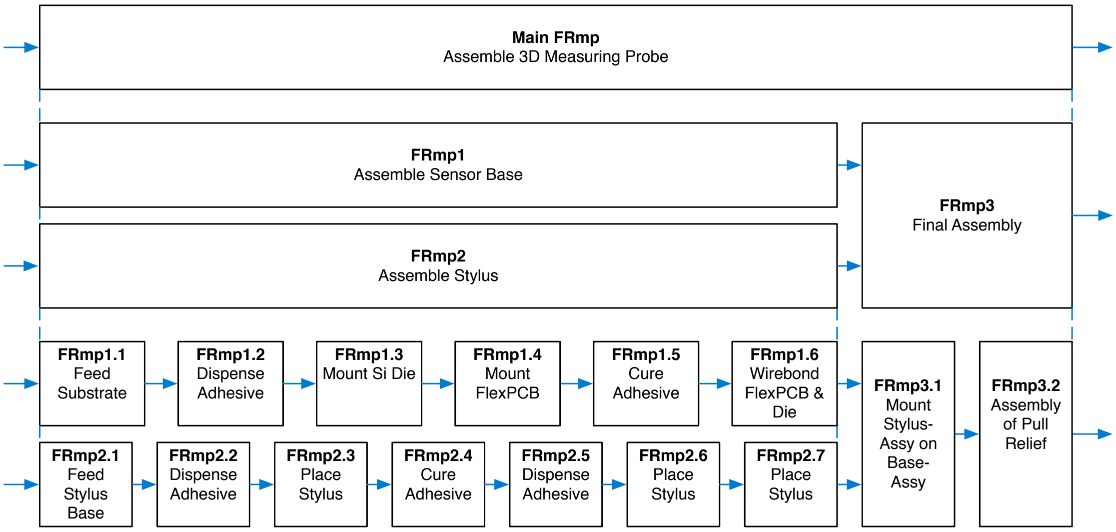
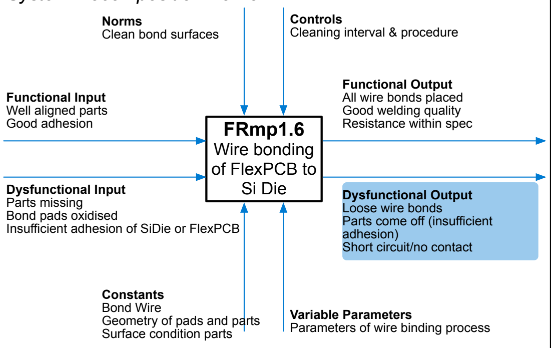


Figure 3.17 *Functional System Decomposition tree of manufacturing process of measuring probe using SADT*

- *Functional system decomposition:* The *μSD process* starts with functional decomposition of Figure 3.17. The SADT analysis at data-level is expanded to the activity level in Figure 3.18;
- *Realisation & Test:* A number of ten measuring probes were produced manually. The manufacturing process was closely monitored. The reliability of the wire bonding process appeared to have a low manufacturing yield and many wire-bonds were failing to bond well;
- *Functional Gating:* In the first cycle of the *μSD process*, the cause of malfunction (failing wire-bonds) was not understood (left column of Figure 3.18). A second problem with the pull-relief of the flex PCB was understood tough a solution was not yet present. The *discrepancy histogram* shows the two problems with *cause not known* and *no solution known*. The histogram is not empty and no advance in decomposition level was made.

Functional System Decomposition

System Decomposition Kernel



Norms
Clean bond surfaces

Controls
Cleaning interval & procedure

Functional Input
Well aligned parts
Good adhesion

Functional Output
All wire bonds placed
Good welding quality
Resistance within spec

FRmp1.6
Wire bonding of FlexPCB to Si Die

Dysfunctional Input
Parts missing
Bond pads oxidised
Insufficient adhesion of SiDie or FlexPCB

Dysfunctional Output
Loose wire bonds
Parts come off (insufficient adhesion)
Short circuit/no contact

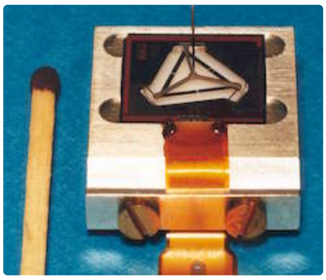
Constants
Bond Wire
Geometry of pads and parts
Surface condition parts

Variable Parameters
Parameters of wire binding process

Problem Identification & Prioritisation
- FRmp1.6-1 — Loose Wire Bonds
- FRmp1.6-2 — Parts come off due to insufficient adhesion
- FRmp1.6-3 — Short circuit/no contact

Due to the fact that the list of three items is oversee able, no prioritisation was applied; complete system was realise and full functionality of the wirebond process was tested

Realisation & Test



Realisation
A number of 10 prototypes were manually assembled. All parts were fully conforming specification. A manual wire-bonder was applied.

Test
- FRmp1.6-1 — Wire bonds appear to detach regularly. Welding of the bonds seems inadequate. No feasible explanation was found so far
- FRmp1.6-2 — No parts came off, adhesion was as expected
- FRmp1.6-3 — No short circuits. Resistance is within specification. The wire bonds that remain intact show no signs of short circuit

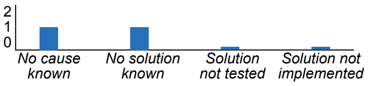
Functional Gating

Assessment of Quality
- FRmp1.6-1— **NOK**, FRmp1.6-2, and FRmp1.6-3 — OK

A new problem that was not expected shows up; when the pull relief for the flexPCB is attached, the flex tends to delaminate completely. Investigation shows that strain is attached to the flexPCB when the pull relief is attached

All *design Imperfections* were coupled back to the designers

Inventory Discrepancies
- The discrepancies were gathered in the discrepancy histogram
- FRmp1.6-1 — No cause known
- FRmp3.2 (pull relief) — No solution known



Determine Hierarchical Impact Level
- Discrepancies in histogram, therefore no change of decomposition level

Functional System Decomposition

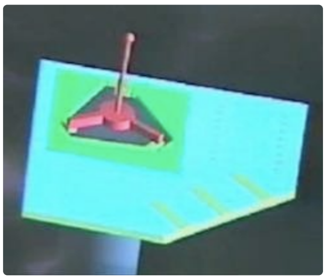
System Decomposition Kernel

Problem Identification & Prioritisation
- FRmp1.6-1 — Loose Wire Bonds

A twofold strategy was applied:
1/ the product design was evaluated:
- The bond process of FlexPCB was specified more tightly
- Pull relief was redesigned to apply a 'loose fold'

2/ An alternative design was elaborated in parallel:
- Eliminating the FlexPCB
- Substituting it by a thick film ceramic part that contains conducting tracks for the interconnection. This solution was applied before.

Realisation & Test



Realisation
Again a number of 10 prototypes were assembled manually.

Dummy substrates and Si-Dies can be applied for a test with the new interconnection

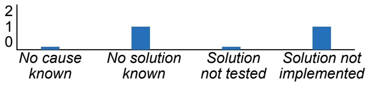
Test
1/ FRmp1.6-1 — Wire bonds **still** appear to detach regularly. The loose fold brings positive outcome; no FlexPCBs were detached further
2/ The alternative design provides full functionality. The FlexPCB is completely eliminated and the external connection is made with a connector on the ceramic substrate (picture)

Functional Gating

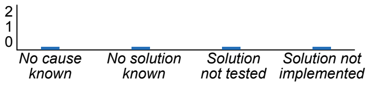
Assessment of Quality
- FRmp1.6-1— Still not working but the cause is found; the adhesive layer underneath the FlexPCB appears to absorb energy during the wire bond process. Therefore the bonds refrain from welding correctly.

All *design Imperfections* were again coupled back to the designers

*Inventory Discrepancies of **option 1***
- FRmp1.6-1 — No solution known
- FRmp3.2 (pull relief) — Solution not implemented



*Inventory Discrepancies of **option 2***
- No discrepancies; solution works as expected



Determine Hierarchical Impact Level
Option 1: Discrepancies in histogram, therefore no change of decomposition level

Option 2: No discrepancies in histogram, with this option the project may advance to the parts level

Figure 3.18 *Application of the µSD framework for the 3D measurement prob (source MA3Solutions BV)*

The second iteration leads to the detection of the cause of the wire-bond problem; a problem with a flexible subsurface under the flexible PCB that absorbs too much bonding energy during the wire-bond process. Figure 3.18 shows the success of the alternative design option; the discrepancy histogram is empty indicating that the process is understood well. After stakeholder discussion, the decision was made for the second option. *Functional gating* was concluded by advancing the level of decomposition from 'system'-level to 'parts'-level. From here the project was brought to proof of concept.

Figure 3.19 *Measuring probe after finalisation of the alternative design, detail of the wire-bonds, the wire-bond tool, and the reconfigured RMS (source MA3Solutions BV)*

The modified design of the measuring probe, and the completion of the RMS, are shown in Figure 3.19.

3.4.3 Case Study 3: Automotive Piezo Actuator Assembly Development using the µSD Framework with Functional Gating based on the Proposed Maturity Grid

The third case study follows the design optimisation of a piezo actuator for an automotive comfort system. The developments of the product itself, as well as the production means, were traced from earliest conception till completion of ramp-up. The case is described in more detail in Appendix A; 'Supplementary Data; Design &

Implementation of Reconfigurable Production Equipment for an Automotive Piezo Actuator'.

Definition of the Product: The actuator is applied for a multi-fold pneumatic switch. It consists of a piezo element that has to be electrically connected to a printed circuit board. The goal is to realise the connection with three small metal 'Contact Springs' as shown in Figure 3.20 (one contact spring is not visible because it is at the back of the product).

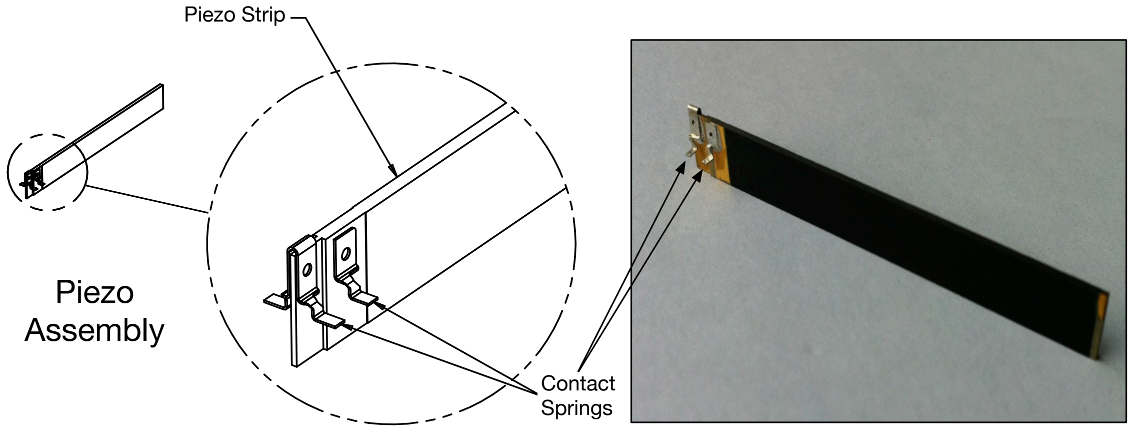


Figure 3.20 *Three contact springs are to be mounted to a piezo element*

The piezo elements have been functionally tested and will be supplied in a product specific tray. The metal contacts will be produced with a stamping process from a coated strip of material (lead-frame). During placement of the contact springs, an ultra-violet curing adhesive is applied over the three contact springs and the piezo to fixate the contacts in their position. After this, the products will be cut from the lead-frame. The estimated cycle-time of the whole assembly process is set to 10 seconds. The carbon paste will be cured in a thermal oven at the end of the assembly sequence.

Functional system decomposition: The decomposed production flow is shown in Figure 3.21 and it consists of a number of nine processes of which the first 2 x 2 processes are executed parallel.

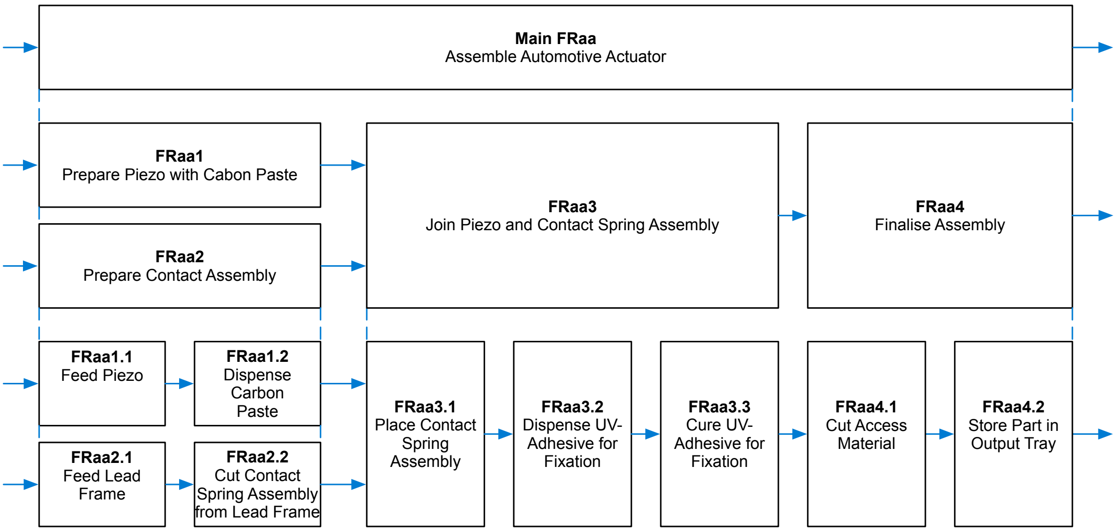


Figure 3.21 *Description on the SADT data-level*

The FMEA is applied to inventory and prioritise the most difficult processes, this is shown in Figure 3.22. Three processes that are completely new, were identified to have the largest risk to potentially delay the project, these processes are (in order of priority):

- FRaa1.2 Dispense carbon paste. This is the process of attaching the electrically conductive adhesive that connects the contact springs with the metallic electrodes of the piezo;
- FRaa4.1 Cut access material. The process of cutting the excess material of the lead frame from the piezo assembly;
- FRaa3.1 Place contact spring assembly. Placement of the assembly with three contact springs, separated from the lead frame but still attached to each other, on the piezo actuator.

A Reconfigurable Manufacturing System (RMS) was configured with three process modules that could perform the prioritised processes. An initial test run was made with 100 products.

Functional System Decomposition

System Decomposition Kernel
- The SADT Activity models of processes FRaa1.2, FRaa3.1, & and FRaa4.1 are shown in appendix A

Problem Identification & Prioritisation
- FRaa1.2 Feed Piezo
- FRaa1.2 Dispense carbon paste
- FRaa2.1 Feed lead frame
- FRaa2.2 Cut contact spring assembly from Lead frame
- FRaa3.1 Place Contact Spring Assembly
- FRaa3.2 Dispense UV-adhesive for fixation
- FRaa3.3 Cure UV-adhesive for fixation
- FRaa4.1 Cut access material
- FRaa4.2 Store part in output tray

FMEA of Identified Processes

Problem	Chance	impact	Result	Priority
FRaa1.1	1%	3	0,03	5
FRaa1.2	25%	10	2,5	1
FRaa2.1	1%	3	0,03	5
FRaa2.2	5%	5	0,25	4
FRaa3.1	20%	5	1	3
FRaa3.2	5%	5	0,25	4
FRaa3.3	1%	3	0,03	5
FRaa4.1	20%	10	2	2
FRaa4.2	1%	3	0,03	5

FMEA of Prioritised Processes

Problem	Chance	impact	Result	Priority
FRaa1.2	25%	10	2,5	1
FRaa4.1	20%	10	2	2
FRaa3.1	20%	5	1	3
FRaa2.2	5%	5	0,25	4
FRaa3.2	5%	5	0,25	4
FRaa1.1	1%	3	0,03	5
FRaa2.1	1%	3	0,03	5
FRaa3.3	1%	3	0,03	5
FRaa4.2	1%	3	0,03	5

Most Problematic Processes (to be addressed in following cycle)
- FRaa1.2 Dispense carbon paste
- FRaa4.1 Cut access material
- FRaa3.1 Place Contact Spring Assembly

Realisation & Test

Realisation
An RMS was configured with processes FRaa1.2, FRaa4.1, & FRaa3.1. An test series of 100 products were produced

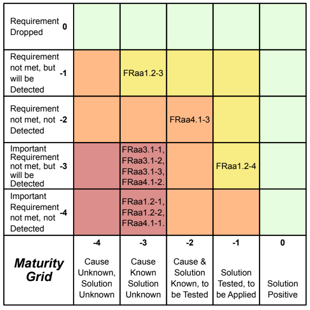
Test
Parts were inspected after manufacturing

Functional Gating

Assessment of Quality
Following problems were found:
- FRaa1.2-1 Wrong amount of carbon paste
- FRaa1.2-2 Short circuit due to adhesive threads
- FRaa1.2-3 Polluted Piezo due to misplace carbon paste
- FRaa1.2-4 Inaccurate position of the carbon paste dots
- FRaa3.1-1 Position Piezo vs contact spring assembly NOK
- FRaa3.1-2 Placement accuracy of the contact spring changes while handling
- FRaa3.1-3 Contact spring assembly bent due to handling
- FRaa4.1-1 De-bonding during cutting action
- FRaa4.1-2 Bending contacts due to cutting action
- FRaa4.1-3 Polluted contacts due to lubricant of the cutting tool

All *design Imperfections* were coupled back to the designers

Inventory Discrepancies
The discrepancies were gathered in the Maturity Grid



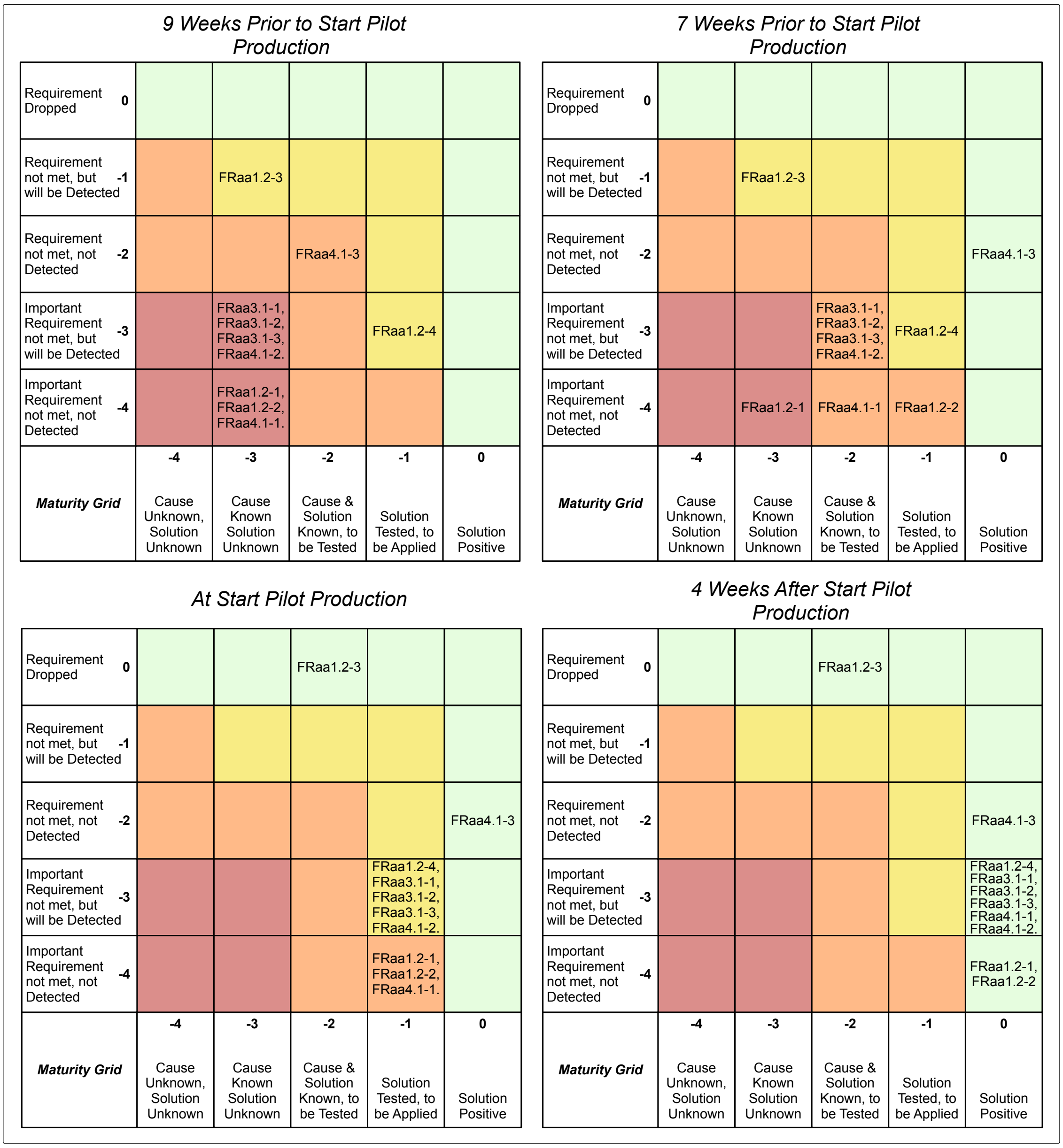
Requirement Dropped **0**					
Requirement not met, but will be Detected **-1**		FRaa1.2-3			
Requirement not met, not Detected **-2**			FRaa4.1-3		
Important Requirement not met, but will be Detected **-3**		FRaa3.1-1, FRaa3.1-2, FRaa3.1-3, FRaa4.1-2.		FRaa1.2-4	
Important Requirement not met, not Detected **-4**		FRaa1.2-1, FRaa1.2-2, FRaa4.1-1.			
Maturity Grid	**-4** Cause Unknown, Solution Unknown	**-3** Cause Known Solution Unknown	**-2** Cause & Solution Known, to be Tested	**-1** Solution Tested, to be Applied	**0** Solution Positive

Determine Hierarchical Impact Level
- Discrepancies in histogram, therefore no change of decomposition level

Figure 3.22 *Application of the µSD Framework for an automotive actuator*

Functional Gating: After assessment, a number of ten problems were defined. The processes are listed in Figure 3.22. All feedback so far was coupled back to the engineers (product and process designers). Next, the problems were gathered in the Maturity Grid Given the number of problems to address, the project could not advance to the next stage.

Iterative improvement cycles were performed on a daily basis. Every 2 to 7 weeks, the risk inventory was completely updated and the status was plotted in the Maturity Grid. Figure 3.23 shows the progression of development during the course of the project.



9 Weeks Prior to Start Pilot Production

Requirement Dropped 0					
Requirement not met, but will be Detected -1		FRaa1.2-3			
Requirement not met, not Detected -2			FRaa4.1-3		
Important Requirement not met, but will be Detected -3		FRaa3.1-1, FRaa3.1-2, FRaa3.1-3, FRaa4.1-2.		FRaa1.2-4	
Important Requirement not met, not Detected -4		FRaa1.2-1, FRaa1.2-2, FRaa4.1-1.			
Maturity Grid	-4 Cause Unknown, Solution Unknown	-3 Cause Known Solution Unknown	-2 Cause & Solution Known, to be Tested	-1 Solution Tested, to be Applied	0 Solution Positive

7 Weeks Prior to Start Pilot Production

Requirement Dropped 0					
Requirement not met, but will be Detected -1		FRaa1.2-3			
Requirement not met, not Detected -2					FRaa4.1-3
Important Requirement not met, but will be Detected -3			FRaa3.1-1, FRaa3.1-2, FRaa3.1-3, FRaa4.1-2.	FRaa1.2-4	
Important Requirement not met, not Detected -4		FRaa1.2-1	FRaa4.1-1	FRaa1.2-2	
Maturity Grid	-4 Cause Unknown, Solution Unknown	-3 Cause Known Solution Unknown	-2 Cause & Solution Known, to be Tested	-1 Solution Tested, to be Applied	0 Solution Positive

At Start Pilot Production

Requirement Dropped 0			FRaa1.2-3		
Requirement not met, but will be Detected -1					
Requirement not met, not Detected -2					FRaa4.1-3
Important Requirement not met, but will be Detected -3				FRaa1.2-4, FRaa3.1-1, FRaa3.1-2, FRaa3.1-3, FRaa4.1-2.	
Important Requirement not met, not Detected -4				FRaa1.2-1, FRaa1.2-2, FRaa4.1-1.	
Maturity Grid	-4 Cause Unknown, Solution Unknown	-3 Cause Known Solution Unknown	-2 Cause & Solution Known, to be Tested	-1 Solution Tested, to be Applied	0 Solution Positive

4 Weeks After Start Pilot Production

Requirement Dropped 0			FRaa1.2-3		
Requirement not met, but will be Detected -1					
Requirement not met, not Detected -2					FRaa4.1-3
Important Requirement not met, but will be Detected -3					FRaa1.2-4, FRaa3.1-1, FRaa3.1-2, FRaa3.1-3, FRaa4.1-1, FRaa4.1-2.
Important Requirement not met, not Detected -4					FRaa1.2-1, FRaa1.2-2
Maturity Grid	-4 Cause Unknown, Solution Unknown	-3 Cause Known Solution Unknown	-2 Cause & Solution Known, to be Tested	-1 Solution Tested, to be Applied	0 Solution Positive

Figure 3.23 *Plotting process risks in the Maturity Grid shows progression of maturity of the relation between product & production*

Pilot Production & Ramp-up: From the start of Pilot Production, the overall process yield was monitored (yield being defined as ‘successfully produced parts’ divided by ’produced parts'). The first SADT analysis took place twelve weeks prior to start of pilot production. The process yield, showed with the black curve in Figure 3.24, took off quickly to over 80% in eight weeks. The green (light-grey) area indicates the daily production of the actuator.

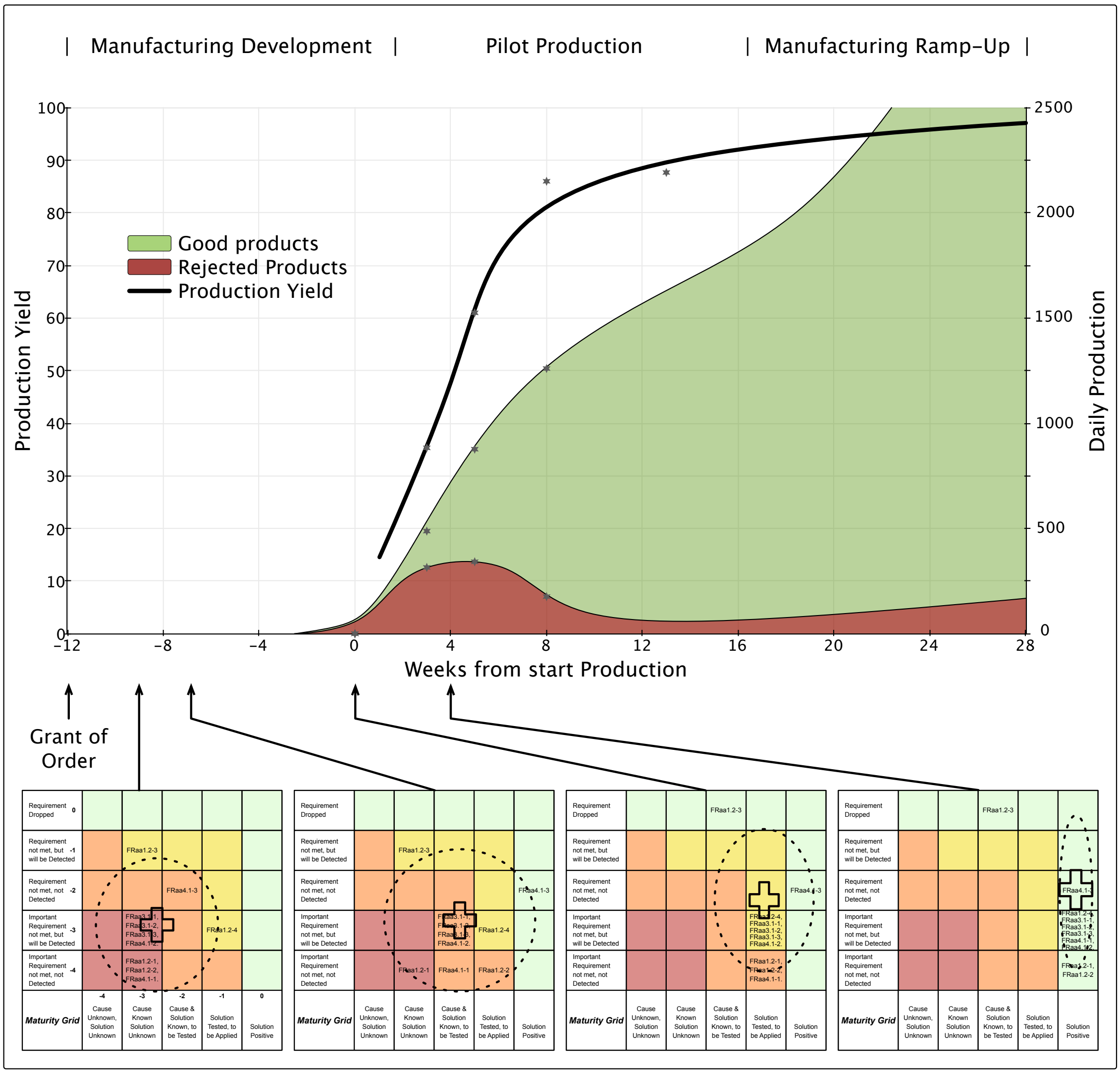


Figure 3.24 Development of production yield as function of time

The red (dark-grey) area represents the development of rejected parts. The four matrices represent the Maturity Grids from Figure 3.23, and the arrows point to the week number they were revised. Rejected parts showed, after a recognisable starting-gust, a decline due to increasing yield. After this, absolute numbers of rejected parts were again increasing but the relative number decreases; during pilot production, the yield was steadily improving, mainly from week 4 on, when the production equipment reached full maturity according to the Maturity Grid. The bottom of Figure 3.24 plots the Maturity Grids with a cross at the gravitational centre to indicate the project progression and a circle to show the scattering of problems.

3.5 Discussion

The chapter proposes an iterative *μSD framework* that applies solutions for *functional system decomposition* and *functional gating* in a flexible framework. It is shown how it can be applied in three different ways to structure and monitor the product development process. In all three cases, the risks are gradually reduced till an acceptable level of satisfaction is reached. As executed in the example cases, the application of QA within the *μSD rationale* indicates progression of the absolute status of the design process, due to the process of plotting multiple results in a histogram or MG. As such it provides optimal feedback during *functional gating*. The shift of the gravitational centre of Figure 3.24 is a measure for project progression. What stands out is that production was started too early since the manufacturing yield in the first 4 weeks of production stays under 50%; many rejected parts are produced. In the next 4 weeks, the yield passes the value of 80% and in another 16 weeks it exceeds 95%. This slow increase is due to the ascending robustness of the manufacturing process, realised by statistical optimisation of the many partial processes. As a general result, this way of monitoring project progression is considered to be successful.

3.5.1 Strengths of the μSD Framework

The iterative *μSD framework* gives feedback on the product- or manufacturing-process at an early stage of development. As such, it enables early optimisation of the product design, changes in parts flow, and equipment layout. At this stage, the moves of the organisation are still flexible due to the fact that no or little investments in production hardware yet have been made. Typically, as investments increase, the 'compliance for change' of the organisation decreases. The reason for this is twofold; on one hand the 'cost of change' increases rapidly as the project evolves (Chokshi et al., 2008; Puik & Moergestel, 2010). On the other hand, reopening gates during the process of *functional gating* causes a psychological barrier leading to organisational resistance. Errors have to be explained in the organisation and time and efforts need to be spent to solve the problems; earlier investments apparently have not addressed the development risks adequately. For this reason, the *μSD framework* is preferably started as early as the initial conception of a system. Since estimation of uncertainties and coding can be applied on conceptual data before quantitative data becomes available, the methodology also works when knowledge about the final system is far from complete. As implemented, *functional gating* supports structured reasoning based on fragmented production concepts. This makes the *μSD rationale* useful in the early stages of development.

The use of a risk analysis tool to visualise severity and nature of the remaining risks in the development process will give an overview of the relevant project risks and reveals the objective project status. This will spin-off with a broader scope than just the engineering level. Though the engineers profit by a complete description of the available options and the effects when actions are omitted, the management level will also be capable of estimating the cause and effect of project control options. This reduces the differences in perception of managers and engineers in the estimates of needed effort to complete the work; less mutual explanation is required, better understanding between departments is obtained.

If SADT is applied, as is done for all three cases, it supports good decomposition of behavioural issues of the applied processes. As a spin-off, it actively documents the production system as whole; standards, operator instructions, and parameters for setup and tuning will all be defined and documented from within the methodology, saving time in a later phase of the project.

3.5.2 Weaknesses of the µSD Framework

Though the industrialisation process of the three test cases, as described in this chapter, was considered successful, the question arises if this also would have been the case if another design logic had been applied. Processes of industrialisation for hybrid microsystems are diverse and involve large investments. This makes objective reference expensive and heterogeneous.

Following *functional system decomposition* with the method of SADT, by describing and mapping the risks as shown, enforces decomposition to proceed in a methodological and systematic way. It leads to the revelation of many process artefacts, which enhances early discovery of problems in: the design, the production method, and the relation between product and processes. Unfortunately, this also has a downside; uncertainties and lack of knowledge about the exact process details could cause an overprotected attitude with (some of) the engineers. This over-consciousness could lead to over-engineering, making the process more complicated than strictly necessary for its application and thus increasing costs without meaning.

Application of risk analysis tools also has drawbacks. The output of the FMEA tends to be uncertain when the magnitude of risks may show erratic ups and downs in the early project stadium, and as a result prioritisation changes accordingly (Werdich, 2011). QA has less sensitivity for these variations because the interpretation of well-chosen coding schemes converges. At the same time, the output result of QA will be dependent on the quality of its coding schemes, and the method of coding tends to increase psychological distance of researchers and their data; it could lead to oversimplification when drawing conclusions (Saldaña, 2012). As a result, *functional gating* loses some of its validity.

3.5.3 Limitations of the μSD Framework

The iterative *μSD framework* deals with the technological uncertainties of the project. It does not necessarily deal with project interactions from a managerial perspective. The managerial interactions have a more social kind of character. They may be less structured and therefore complex to control or to improve. The method would need changes to deal with the complexity of the early project acquisition process. For now, the method is likely to fall short.

Though coding schemes can be adjusted for particular situations, interpretation of the same situation with a different coding scheme could lead to different conclusions. Therefore, changing a coding scheme does not increase confidence with the procedure and compromises objective reference based on experience. In this chapter, the coding schemes for the second case and the first dimension of the third case were maintained to enable straight comparison. The bottom line is that interpretation of the results of QA requires confidence with the applied coding scheme and therefore the method may be less flexible than it initially appears. The same situation applies for the Maturity Grid, since it basically is a two-dimensional application of the same principle.

The *μSD framework*, as described in this chapter, was tested on three cases that concerned high-tech microsystems with a high pressure on technological development and the lead-time available to get the work done. The method appears generic, but may need recalibration if applied to other markets that have other dynamics or different technological demand.

All experiments thus far were carried out applying the SADT analysis. For product development, SADT can be replaced by the Morphological Matrix, Pugh Matrix, or by QFD. When products and processes are developed concurrently, this means that these methods should be applied simultaneously.

3.5.4 Other Considerations

In the *μSD framework*, three feedback processes work together, each with their own dynamics. This can be seen in Figure 3.1: (i) the process of design consisting of rapid interactions between *functional system decomposition* and *synthesis*, (ii) the iterative feedback that realises and compares true and targeted functionality, and (iii) the linear completion schedule of *functional gating* that progresses every time a gate closes. The design interactions may take place with cycles between minutes and weeks (depending on the hierarchical level), physical testing could take place on a scale of days to months, and the gating process typically advances on a monthly or quarterly basis. *Functional gating* has only influence on the latter which also is the steadiest process, but *functional system decomposition* is part of all three processes.

3.5.5 Opportunities for Further Improvement

The requirements analysis that forms an important part of *functional system decomposition* was not yet optimised for the *µSD framework*. Expectations are that good requirements definition which: (i) follows decomposition, and (ii) addresses product design as well as manufacturing, will improve the capabilities for concurrent design of the *µSD framework*. One of the envisioned methods is the methodology of Axiomatic Design as developed and optimised at the Massachusetts Institute of Technology from the late seventies on (Suh, 1990). Axiomatic Design supports the structured decomposition of requirements in the functional, physical, and process domains. As such it can enable the *µSD framework* for concurrent development.

3.6 Conclusions

The *µSD framework* as proposed in this chapter may be considered successful. The solution forces *functional system decomposition* of the product design and the chosen manufacturing solution. Results from tested prototypes are available at an early stage. These results are compared with the expected results from the model of the product. And they arrive when the moves of the organisation are still flexible due to the fact that only little investments in hardware have been made. It will lead to improved flexibility to adapt to changes and it supports design for assembly. *Functional gating* was applied in three ways. FMEA will deliver a numerical update of the design progression every time the loop is completed in a way that is familiar to managers and engineers. Qualitative Analysis presents the quality of the design in a coded scheme. If the Maturity Grid is applied, it will visualise absolute status of the project and thus project progression. Researchers, engineers and managers may embrace a graphically oriented way of presenting information. Being an intuitive way to communicate, barriers seem to fade, and better understanding in the organisation is attained.

CHAPTER 4

‘CONCURRENT MICROSYSTEM DEVELOPMENT’ (CμSD) FRAMEWORK; ENHANCING THE μSD FRAMEWORK WITH RATIONALES FOR CONCURRENT SYSTEM DECOMPOSITION AND CONCURRENT GATING[*]

4.1 Introduction

The *μSD framework* that was introduced in Chapter 3 has proven to work for microsystems. A feature that really contributes to microsystem development could be further improved; the quality to concurrently address the product design and the required manufacturing technology. The concurrent approach will affect both *functional system decomposition* and *functional gating*. The former because decomposition would need integration of the manufacturing domain and the latter would need adjustment to enable concurrent assessment. Therefore, the Objects *functional system decomposition* and *functional gating* will be developed further as shown in Table 4.1. The Object ‘Concurrent System Decomposition’, proposed in this chapter, enhances system

---

[*] *Parts of this chapter were published in:*

Puik, E. C. N., Smulders, E., Gerritsen, J., Huijgevoort, B. V., & Ceglarek, D. (2013). A Method for Indexing Axiomatic Independence Applied to Reconfigurable Manufacturing Systems. In M. K. Thompson (Ed.), (1st ed., Vol. 7, pp. 186–194). Presented at the 7th International Conference on Axiomatic Design, Worcester.

Puik, E. C. N., Gielen, P., Telgen, D., Moergestel, L., van, & Ceglarek, D. (2014). A Generic Systems Engineering Method for Concurrent Development of Products and Manufacturing Equipment (Vol. 435, pp. 139–146). Berlin, Heidelberg: Springer Berlin Heidelberg. http://doi.org/10.1007/978-3-662-45586-9_18 decomposition to a concurrent rationale. The same applies for the Object *functional gating*; it will be converted to a new Object 'Concurrent Gating' that supports the concurrent capability of *concurrent system decomposition*. The thus obtained method is called 'Concurrent μSD' (CμSD) framework and it is applied to a case for microsystem development.

Table 4.1 *Chapter 4 improves concurrency of the CμSD framework*

Object Chapters	*Functional system decomposition*	*Functional gating*
Chapter 3	Structured Analysis Design Technique (SADT) in combination with Failure Modes and Effect Analysis (FMEA)	Three different methods: (1) Remaining uncertainties based on the FMEA (2) Qualitative Analysis (QA) with unidirectional coding (3) Qualitative Analysis using a Maturity Grid (MG) with a bi-directional coding scheme
Chapter 4	*Concurrent system decomposition* based on Axiomatic Design	*Concurrent gating* based on the completion of decomposed hierarchical levels (or tested levels)
Chapter 5	Same as in Chapter 4	*Intelligent gating* based on Information in Design

The proposed *CμSD framework* also provides a rationale and guidelines on ways to embed it into project execution, which will be discussed as *CμSD Rationale*. Chapter 5 of the thesis will contribute further to the *CμSD framework* with a more advanced solution for the gating function. In that chapter, *concurrent system decomposition*, is maintained as is.

The chapter is organised as follows: Section 4.2 analyses the current situation and defines the key limitations. Section 4.3 explains the methodology of the *CμSD framework* and discusses the *CμSD rationale*. Section 4.4 illustrates the methodology using an industrial case study in which the *CμSD rationale* is applied and tested. Section 4.5, discusses the findings and lessons learned. Finally, Section 4.6 draws conclusions.

4.2 Analysis of the μSD Framework and Approach for improvement

The way the feedback loop is implemented in the *μSD framework*, will remain unchanged in the *CμSD framework*. The problem definition that builds further on the problem statement of Chapter 3 will be analysed in Subsection 4.2.1. The current situation will be inventoried in Subsection 4.2.2, and key limitations will be defined in Subsection 4.2.3.

4.2.1 Problem Definition

It is important for product designers to have knowledge of manufacturing technologies to oversee if their designs can or cannot be manufactured. The cost of manufacturing is largely determined by how solutions are implemented in the design by the designer e.g.: material use, manufacturing processes, and tolerances. Characteristic for microsystems are the small geometrical features that require higher accuracies for comparable shape-variation-tolerances. Small features have a dominant impact on manufacturing cost for microsystems because: (i) accurate equipment is more cost intensive than standard equipment, (ii) the number of rejected parts statistically grows when the limits of the manufacturing process are achieved, and (iii) small parts only require small amounts of material which reduces material costs, relatively emphasising the cost for manufacturing. The high impact of manufacturing actions on total product costs pleads for a concurrent approach in development. Concurrent development means that product and process technology (and eventual production means) are not developed sequentially but simultaneously. The parallel approach of products and processes enables the product design to be adjusted to process technology and vice versa. In combination with an iterative design process, product features, that are difficult to manufacture, can be recognised and addressed on a short notice. Also, new opportunities for optimisation of the design may come available if process technology offers features that were not known to the designer.

The systems engineering models that are widely applied for microsystem development in industry, like the Waterfall-Model and the V-Model do not address concurrent behaviour. Microsystem development will benefit from a method that combines the *μSD framework* with an intrinsically concurrent approach.

A related problem is found with the process of *functional gating*. The concurrent way of working needs a concurrent gating function that assesses current state of design from the perspective of overall project progression; the measured progression should be based on advances of the product design as well as the manufacturing processes, because both evolve concurrently during project execution.

4.2.2 Current Situation

The V-model does not support or implement concurrent design but on the other hand, the concurrent way of working is not excluded in the model. A closer look reveals that the way that the V-Model approaches the product design is actually not consistent during the successive stages; at the top of the left-hand leg, the V-Model starts with decomposition and specification of FRs that, as explained in Subsection 2.2.6, both are executed in the functional domain. However, as the project moves towards the right-hand leg, focus is on ‘realisation and testing’ that relates to the physical domain. During the testing stages, the focus shifts back to the functional domain because ‘Acceptance Tests’ are performed on functional requirements and as such take place in the functional domain.

Processes used for manufacturing are basically not applied in the V-Model. The reason is that the V-Model, finding its origin in software engineering, was not developed with manufacturing in mind. Due to the digital nature of software, reproduction of software is a digital and thus tolerant process that needs little specific attention.

Subsection 3.3.4 explained the idea behind the gating function; decisions at the hierarchically higher decomposition levels need proper substantiation before moving to the next level. Once a gate is closed, all decisions are frozen and the project is assumed to never move back. Unfortunately, the process of gate-closing gives an indication of certainty that cannot be substantiated. Figure 4.1 shows the gates in the left-hand leg of the V-Modell XT (the complete overview of the V-Modell XT was shown in Figure 2.4).

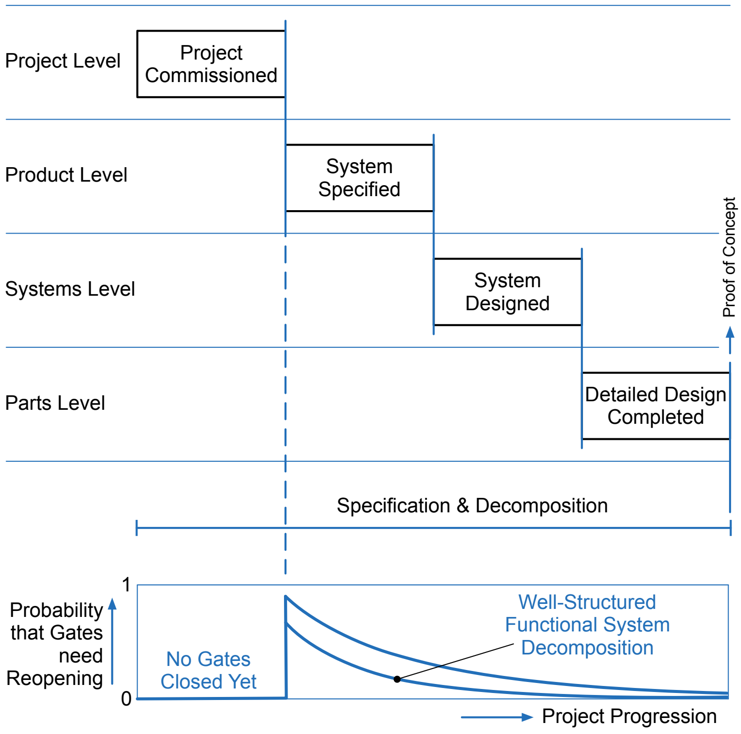


__Figure 4.1__ The gates of the left-hand leg in the V-Modell XT

During development of the project, and thus descending in the V-Modell XT, designers apply their capabilities to foresee future obstacles as much as possible, but this does not guarantee that the project will evolve as expected. For instance, if a project completes *functional system decomposition* at Product Level as shown in Figure 4.1, the Systems- and Parts Levels are not completely crystallised yet. This means that there is still uncertainty left in the remaining development process. As a result, there is no guarantee that these Systems and Parts Levels can be completed without obstacles when the gate at the Product Level is closed. The general idea behind the V-Modell XT is that there are many options to provide solutions for problems at lower levels, but there is no guarantee that all problems indeed can be solved. As a result, a chance remains that the project needs intervention at the (hierarchically higher) product level; closed gates need reopening, and previous choices are re-considered. This risk continues till the tip of the V is reached and 'Proof of Concept' is verified by exhaustive understanding of the design (most right hand object in Figure 4.1).

The bottom chart of Figure 4.1 plots the evolving chance that gates need reopening. Though the shape of the curve is cannot be completely substantiated without further investigation, it may be stated that: (i) the curve starts at a value lower than 1, (ii) it generally descends as knowledge of the designer increases, and (iii) it asymptotically goes to zero since theoretically there is always some remaining project risk left (as will be proven in Chapter 5). It shows that the project risk is not equal to zero and at least a small chance exists that gates need to be reopened.

4.2.3 Key Limitations

A critical limitation of the Waterfall- and V-Models is their shortcoming in addressing product development concurrently to the, functional, physical, and process domains. Instead, focus shifts between functional and physical domains, but in a successive manner, and not concurrently. This leads to the following drawbacks:

- The domains are not structurally addressed to: (i) guard functionality, (ii) guard the physical design, and (iii) guard process technology. Because of this, these domains will not be adjusted to each other;
- It is difficult for unexperienced users to understand the way the domains change because it is not implemented transparently.

Another limitation is that the Waterfall- and V-Model do not forcefully require analytic substantiation of *functional system decomposition* and *functional gating*. If decomposition is incomplete, or branches of the decomposition tree are not decoupled, there is no signalling function that monitors the quality of decomposition. Analogue, if cross domain adjustments of the design and required process technology fail, there is no guarantee that testing brings these shortcomings to the surface, simply because the process domain is not included in the model. It leads to the following drawbacks:

- If decomposition is incomplete, testing will not take place for that particular functionality;
- Even if decomposition is complete, testing may miss certain functionality because rigorous and exhaustive testing is laborious and costly. As a result, certain functionality is not evaluated and could fail in a later stage;
- Neither Waterfall- nor V-Model address the process domain; all potential optimisations between the physical domain and the process domain remain unused.

4.3 Methodology for Implementation of the CμSD Framework

The *CμSD framework* maintains the iterative structure that was exposed in Chapter 3, however, the way *functional system decomposition* and *functional gating* are implemented is upgraded to improve the capability to approach the design concurrently. Figure 4.2 shows the full layout of the *CμSD framework* as investigated in this chapter.

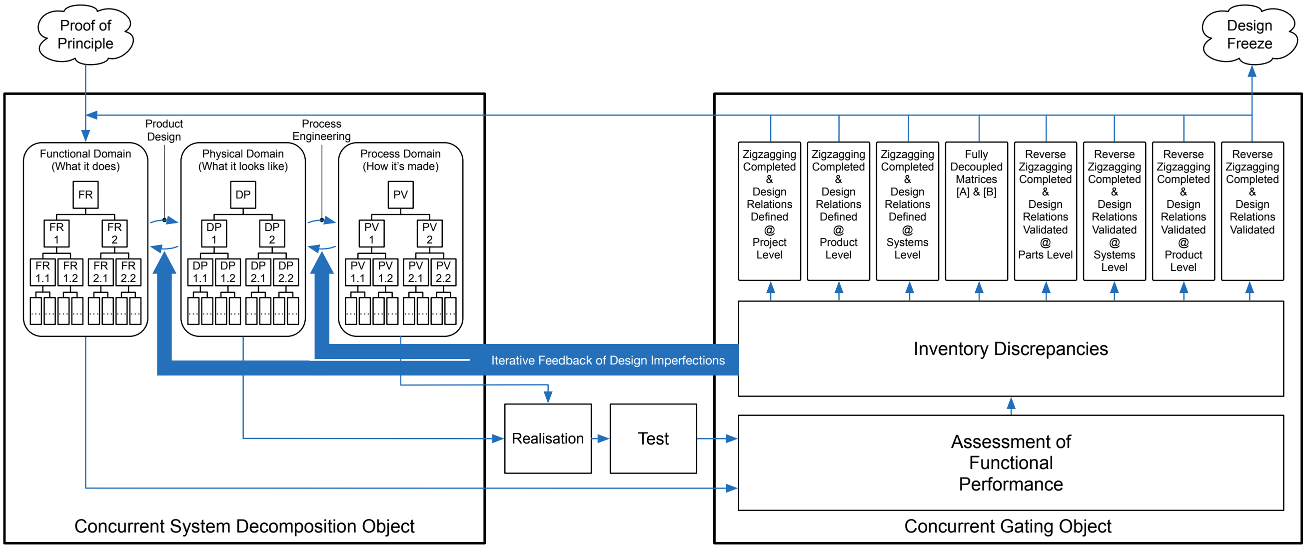


Figure 4.2 ***CμSD framework***

The improvement is realised by the application of the methodology of Axiomatic Design (AD). Knowledge of AD is assumed; Appendix B provides necessary background information for reading this chapter if necessary.

The next two subsections explain *concurrent system decomposition* and its rationale, the two subsections following thereafter explain *concurrent gating* and its rationale.

4.3.1 Concurrent System Decomposition

As the different domains: Customer Attributes (CAs), Functional Requirements (FRs), Design Parameters (DPs), and Process Variables (PVs), are elementary starting points for AD, it intrinsically addresses product, systems, and even organisational design concurrently. A process of zig-zagging (Appendix B) makes sure that the domains stay lined up with each other. Figure 4.3 shows how product planning, product design, and process design are positioned between the domains.

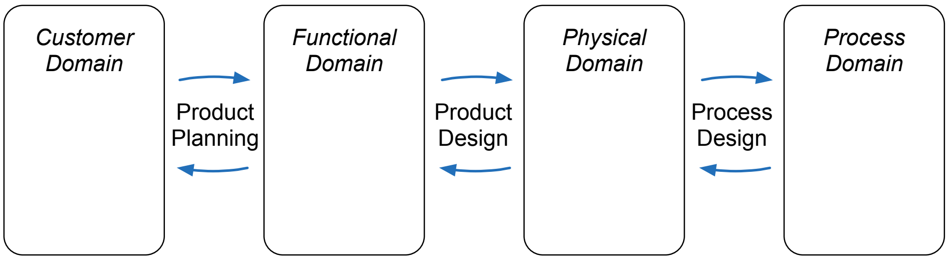


Figure 4.3 *Axiomatic domains and their relations*

Ideally, the CAs would be the leading factor in the design process as they accurately describe the customer demand, but since CAs are strongly context dependent, and the context is that of the customer, it is hard for the designer to develop an accurate understanding of what the customer needs to be satisfied. Typically, a translation takes place and the wishes of the customer are expressed in the functional domain. Functional properties of the product are caught in: FRs, and the value of those FRs. These two form the basis for Project Initiation Documents (PID) and contracts. From this point on, the functional domain is typically leading for the development process. The list of FRs and their values usually starts as a shortlist for project and product definition. During the project, completion of a set of decomposed FRs continues till a list emerges that is Collectively Exhaustive and Mutually Exclusive (CEME) (Brown, 2005).

Domains are basically static containers of the current status of the design (Puik & Ceglarek, 2015b). The product designer has the capability to adjust and align the status of the functional and the physical domains. Simultaneously, process engineers have the capability to adjust and align the physical and the process domains. The process of design, as well for products and processes, is the act of adjusting the content of the domains in a structured manner. There are two options: (i) the traditional way in which CAs are translated to FRs. The list of FRs is then evaluated with the customer who agrees to use the list as a general starting point of the project, or (ii) the customer joins the project and the FRs are developed as the project continues. The first option would be the option for the more traditionally managed projects like the Waterfall-Model and the V-Model, the second option is typically applied for the agile project management methods e.g., Scrum. For now, the traditional method is applied, later in the thesis the agile way will be integrated as well.

To substantiate decomposition, the design relations of AD are applied to structure the completed part of the decomposition tree. Figure 4.4 shows how the design matrices [A] and [B] respectively represent product design and process design.

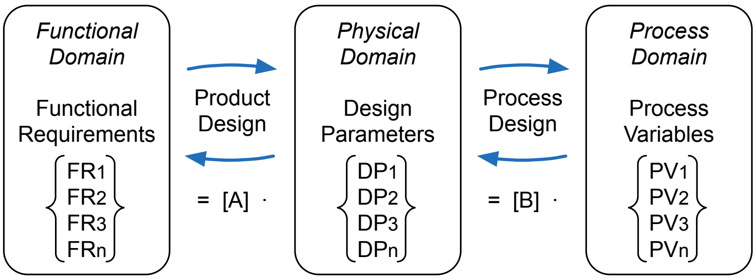


Figure 4.4 *Axiomatic domains and the design matrices*

A complete decoupled or uncoupled design matrix, per level of decomposition. The design relations, when applied correctly, provide the capability to make sure that DPs are satisfied by PVs, and thereafter, FRs are satisfied by DPs. The design relations may be decomposed conforming the *functional system decomposition* of the product and by stretching this over all three domains a true way of *concurrent system decomposition* is acquired.

4.3.2 CμSD Rationale for Implementation of Concurrent System Decomposition

The decomposition kernel of the *CμSD framework* is shown more detailed in Figure 4.5. The proposed rationale applies the following steps:

- The project starts with project brief, or any list of project-FRs. The FRs are decomposed in a 'Zigzagging' motion over the different domains, as is the typical procedure in AD (blue line in Figure 4.5, explained in detail in Appendix B.2). While doing this, the design relations are defined and the product and process design matrices are developed. As long as definition of the design relations is successful, this process is continued. However, when the procedure falters, because the definition of design relations is uncertain and this cannot be successfully analysed further, the decomposition process stops;

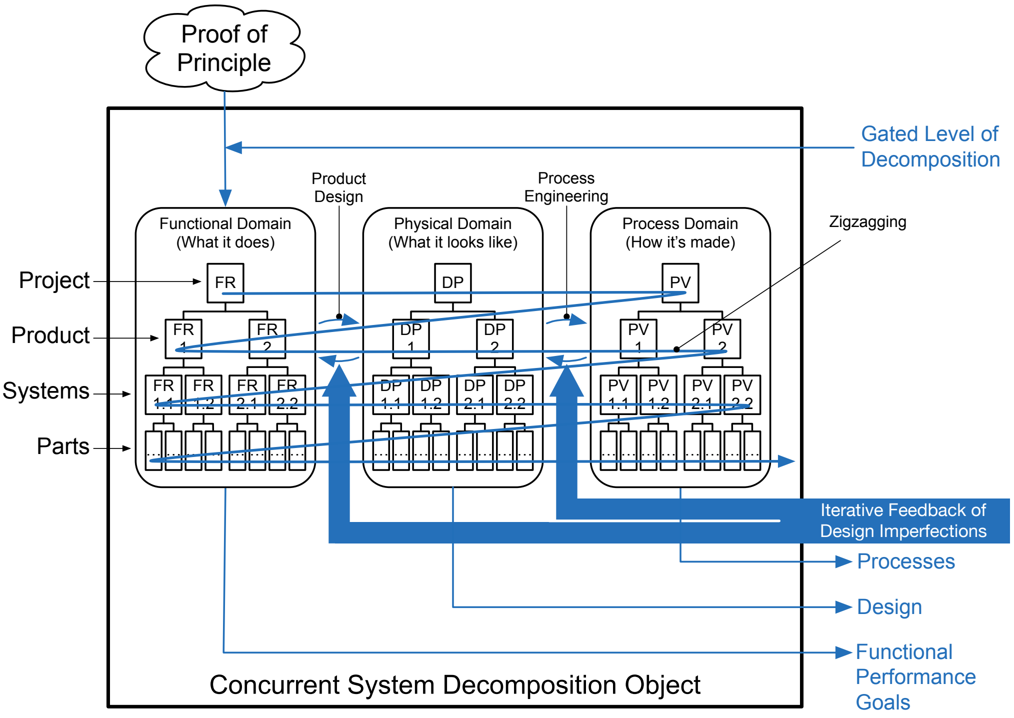


Figure 4.5 *The functional system decomposition object and the process of zigzagging across the domains of Axiomatic Design*

- The second step is to plan an investigation that sorts out the particular uncertainty (or a number of uncertainties). The investigation should be safe-fail; whether the test produces a positive or a negative outcome, the outcome should be worth the investment;
- Finally, the investigation will be executed by the Objects 'Realisation' and 'Test'. The Object *synthesis* that was applied in the *µSD framework* is not explicitly named here, but it is still intrinsically embedded between the functional and physical domains of AD. The design output is represented by the DPs that are realised by the PVs. Therefore, the PVs are applied as enablers for realisation of the DPs. Via the Object *test*, the results are forwarded to the Object *concurrent gating*.

4.3.3 Concurrent Gating

Concurrent gating in the *CµSD framework* implemented as an enhanced version of gating in the V-Modell XT (Figure 4.6). It also applies AD as a basis whereby decomposition and gating work together. AD is applied to substantiate the gating stages as implemented in the V-Modell XT; It applies the Independence and Information Axioms and the hierarchical layers of the decomposition tree (Figure 4.6). It serves as a guideline to the designer to agree on preconditions that are clear and measurable.

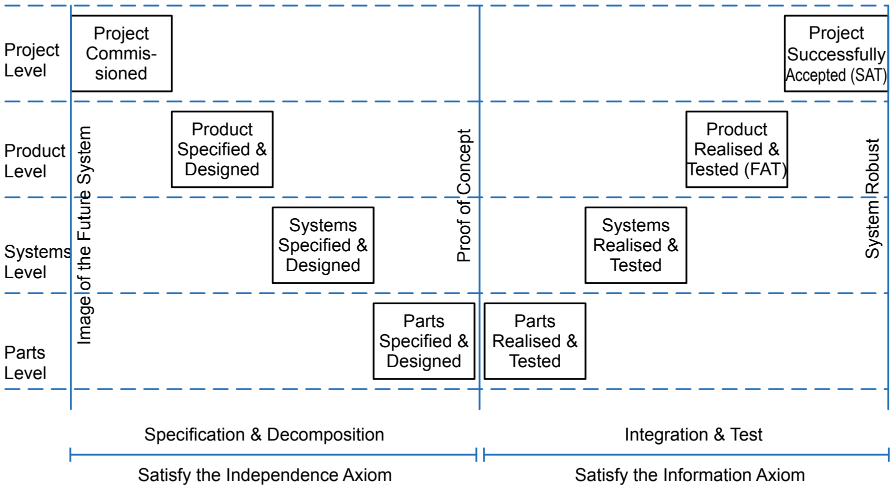


Figure 4.6 *Gating function of the V-Modell XT*

The Axioms are embedded in the gating function of the V-Modell XT as follows. The left-hand leg addresses decomposition and specification and it ends with: (i) a fully defined decomposition tree, and (ii) all specifications known in detail. If the last gate of the left-hand leg is closed, the structure of the design should be completely known and understood. In terms of AD this means that the design relations are known, the design is uncoupled or decoupled, and the Independence Axiom is satisfied. The right-hand leg of the V-Modell XT assesses the quality of the design to see if the structure is robust. If testing is successfully completed, by passing the ‘Factory and Site Acceptance Tests’, the design may be considered fully robust. In terms of AD this means that both the Independence and the Information Axioms are satisfied. Not only the design relations are known, but they are also intrinsically and statistically robust.

Both the V-Modell XT and AD only address the conceptual and robustness stages of the product design but not the explorative phase. This can be shown by comparing it to Banathy’s model of Dynamics of Divergence and Convergence (as was shown in Figure 2.7). Table 4.2 compares the project phases.

Table 4.2 *Overview of design phases of V-Modell XT, AD, and Dynamics of Divergence and Convergence*

Phase / Models	Exploration	Conceptual	Robustness
V-Modell XT (Höhn et al., 2008)	Project acquisition and definition	Left-hand leg Decomposition and Specification	Right-hand leg Realisation and Test Verification & Validation
Axiomatic Design (Suh, 1990)	Complexity Axiom may be used	Satisfy Independence Axiom (Focus on Structure)	Satisfy Information Axiom (Focus on Robustness)
The Dynamics of Divergence and Convergence (Banathy, 1996)	The Image of the Future System	The Model of the Future System	Not Applicable

Banathy's model shows the explorative phase and the conceptual phase. The V-Modell XT and AD shown the relative position of the conceptual and robustness phases within the whole development process. The explorative phase is not so well defined in the V-Modell XT nor AD. Therefore, only the last two phases, conceptual and robustness, will be implemented for *concurrent gating*.

4.3.4 CµSD Rationale for Implementation of Concurrent Gating

Figure 4.7 shows a close up of the way gating is executed and how AD is implemented. The upper row shows the implementation of gating in the V-Modell XT and the lower row shows the way this is expanded to AD.

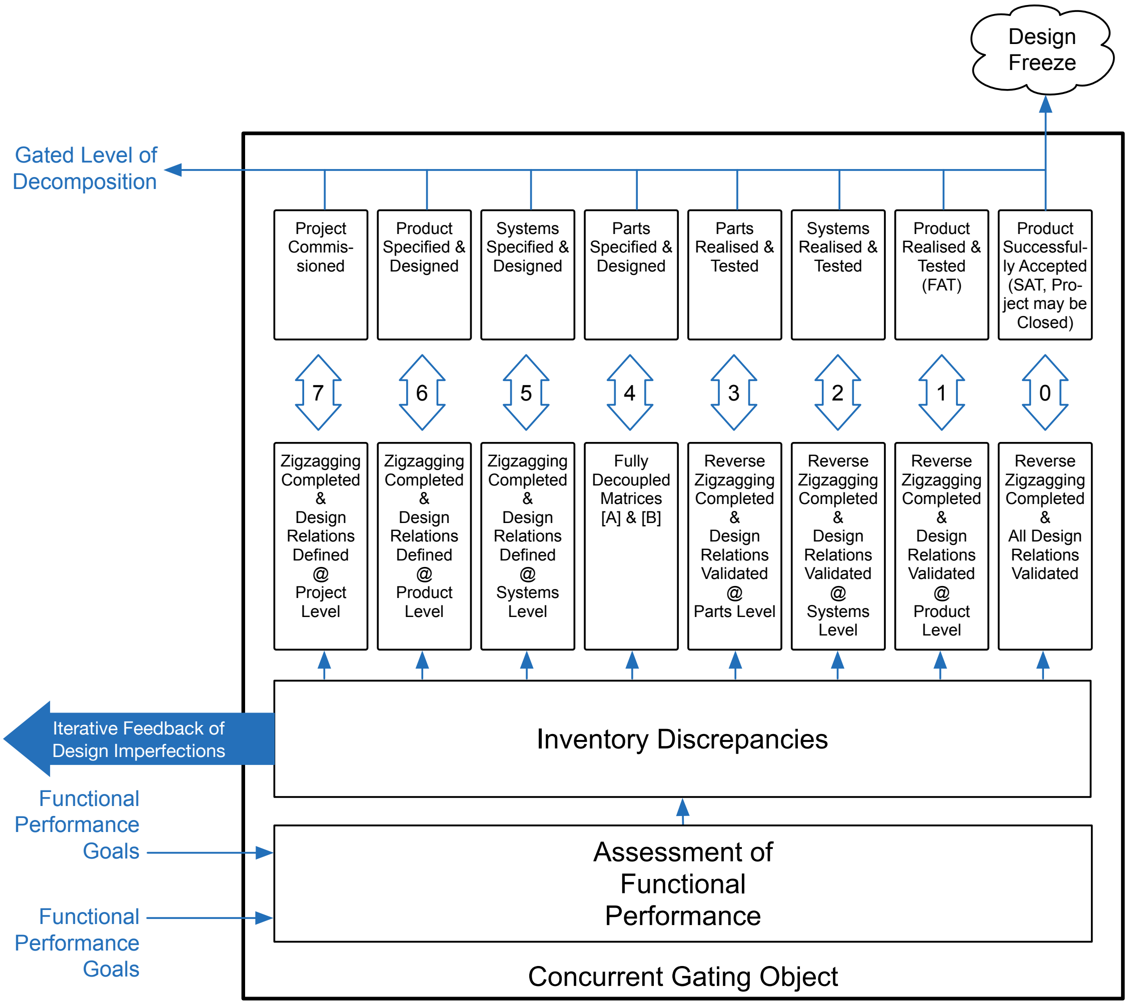


Figure 4.7 *Overview of the eight index stages*

The gating stages of the V-Modell XT are substantiated by the definition of a measure that can be quantified using AD. The numbers count the number of stages till the end. The Independence Axiom is satisfied at stage 4 and the Information Axiom at stage 0. Hierarchy is applied in four steps to add more detail to each axiom.

The procedure of *concurrent gating* is as follows:

- The first step is to compare the testing results with the functional reference of the model of the system. The last hierarchical level that was successfully addressed by the *zigzagging* process, determines the number of gates that may be closed. This is measured by: (i) successful decomposition, and (ii) successful definition of the design relations;
- If possible, the next gate is closed;
- After readjusting the number of closed gates, the feedback loop is closed and the decomposition/synthesis continues from where is was stranded.

The procedure of *zigzagging* is applied downward in the first four stages till proof of principle is reached. In the second half of the *concurrent gating*, the direction of *zigzagging* is reversed from bottom to top. In these four stages, it characterises till what level the system is robust. This second way of *zigzagging* is referred to as *reversed zigzagging*.

4.3.5 How the CμSD Framework Addresses the Key Limitations

The *CμSD framework*, combined with elements of AD, addresses the aforementioned limitations by providing the following capabilities: (i) concurrency is structurally applied by implementation of the domains of AD, and (ii) decomposition is substantiated by structured application of the design rules of AD. The former provides the capabilities that:

- All domains are addressed during every iteration;
- The domains are addressed successively, starting with FRs and followed by DPs and PVs, but since this is repeated every iteration, domains are addressed in

parallel. As such, the domains are well adjusted to each other before the next level of decomposition (the next management level) starts;

- The implementation is transparent, which will be exposed in more detail in the next Subsection.

Substantiation of decomposition and gating using AD takes care of:

- These processes are now based on design relations. Design relations require the designer to understand the design;
- The understanding that is required to define these relations, increases the certainty that all aspects of the design are recognised. Al recognised design relations need to pass testing successfully to close gates.

4.4 Case Study: Development of an Inkjet Printing Head and Reconfigurable Manufacturing System

The case concerns the development of a new inkjet printing system for industrial applications. The client has a long record on printing technologies but, due to technologically driven changes in his market, the intention is to address a new market segment. This is the market segment of ’Sign and Display’ printing. Sign and display printing is characterised by a large variety in printable materials e.g.: paper, polyester, foamboard, cardboard, and even wood. The client has identified what is needed to approach the market. A first analysis is shown in AD style in Figure 4.8 (the domains are represented in vertical direction).

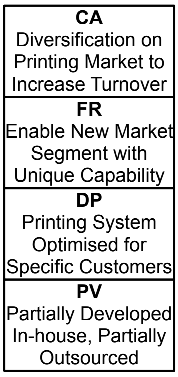
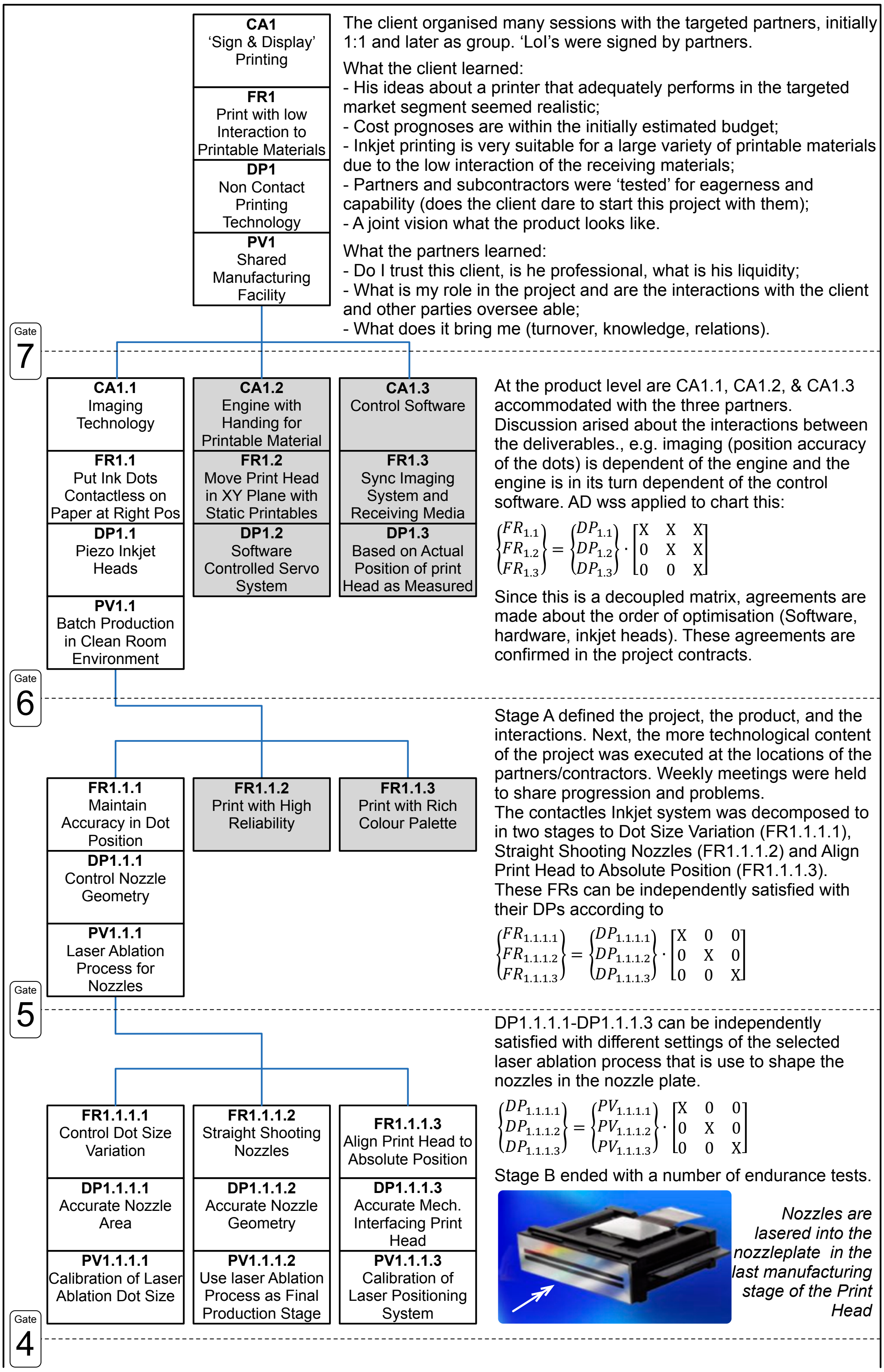


Figure 4.8 *Initial analysis of the baseline of the customer*

The project is ambitious; not only the market segment is new but also the proposed technology. Therefore, the goal is to find project partners that are willing to participate in the project, as well financially as with their technological expertise. Three partners and one customer were found: (i) a company that produces inkjet print heads, (ii) a mechatronics company aiming to build the printer engine, (iii) a control software company, and finally (iv) a launching customer willing to purchase a number of systems with reservation that a basic specification is realised. Figure 4.9 shows the project design with attributes CA1.1, CA1.2, and CA1.3 all outsourced to different partners. This case follows the project through its decomposition stages. To keep the case manageable in this paper, the focus is restricted to the accuracy of the inkjet print head (grey parts are not further decomposed). Figure 4.9 shows gates 7 to 4 of *concurrent gating* and the important design relations. Figure 4.10 shows gates 3 to 0 and explains how the design relations were validated by producing pilot series of products.



The client organised many sessions with the targeted partners, initially 1:1 and later as group. 'LoI's were signed by partners.

What the client learned:
- His ideas about a printer that adequately performs in the targeted market segment seemed realistic;
- Cost prognoses are within the initially estimated budget;
- Inkjet printing is very suitable for a large variety of printable materials due to the low interaction of the receiving materials;
- Partners and subcontractors were 'tested' for eagerness and capability (does the client dare to start this project with them);
- A joint vision what the product looks like.

What the partners learned:
- Do I trust this client, is he professional, what is his liquidity;
- What is my role in the project and are the interactions with the client and other parties oversee able;
- What does it bring me (turnover, knowledge, relations).

At the product level are CA1.1, CA1.2, & CA1.3 accommodated with the three partners. Discussion arised about the interactions between the deliverables., e.g. imaging (position accuracy of the dots) is dependent of the engine and the engine is in its turn dependent of the control software. AD wss applied to chart this:

$$\begin{Bmatrix} FR_{1.1} \\ FR_{1.2} \\ FR_{1.3} \end{Bmatrix} = \begin{Bmatrix} DP_{1.1} \\ DP_{1.2} \\ DP_{1.3} \end{Bmatrix} \cdot \begin{bmatrix} X & X & X \\ 0 & X & X \\ 0 & 0 & X \end{bmatrix}$$

Since this is a decoupled matrix, agreements are made about the order of optimisation (Software, hardware, inkjet heads). These agreements are confirmed in the project contracts.

Stage A defined the project, the product, and the interactions. Next, the more technological content of the project was executed at the locations of the partners/contractors. Weekly meetings were held to share progression and problems.
The contactles Inkjet system was decomposed to in two stages to Dot Size Variation (FR1.1.1.1), Straight Shooting Nozzles (FR1.1.1.2) and Align Print Head to Absolute Position (FR1.1.1.3). These FRs can be independently satisfied with their DPs according to

$$\begin{Bmatrix} FR_{1.1.1.1} \\ FR_{1.1.1.2} \\ FR_{1.1.1.3} \end{Bmatrix} = \begin{Bmatrix} DP_{1.1.1.1} \\ DP_{1.1.1.2} \\ DP_{1.1.1.3} \end{Bmatrix} \cdot \begin{bmatrix} X & 0 & 0 \\ 0 & X & 0 \\ 0 & 0 & X \end{bmatrix}$$

DP1.1.1.1-DP1.1.1.3 can be independently satisfied with different settings of the selected laser ablation process that is use to shape the nozzles in the nozzle plate.

$$\begin{Bmatrix} DP_{1.1.1.1} \\ DP_{1.1.1.2} \\ DP_{1.1.1.3} \end{Bmatrix} = \begin{Bmatrix} PV_{1.1.1.1} \\ PV_{1.1.1.2} \\ PV_{1.1.1.3} \end{Bmatrix} \cdot \begin{bmatrix} X & 0 & 0 \\ 0 & X & 0 \\ 0 & 0 & X \end{bmatrix}$$

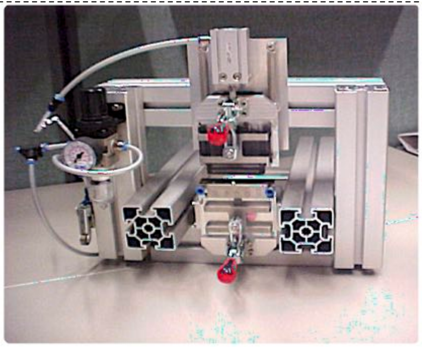
Stage B ended with a number of endurance tests.

Nozzles are lasered into the nozzleplate in the last manufacturing stage of the Print Head

Figure 4.9 *Concurrent System Decomposition and gates 7 to 4 of functional gating*

Gate
4

Varying the process windows was applied to make the project robust in stage 5. First, the DPs were varied within their tolerances and the effect on satisfaction of the FRs was studied. Example: during stage B, a test setup for bonding nozzle plates was developed. The quality of the bond proces could be consciously reduced and the effect on the FRs investigated. This gave the DP-FR tolerance relation. By producing small batches of 25 pcs, the tolerances on the production process could be evaluated which gives the relation between PV and DP and as such completes the reverse zigzagging process. Similar procedures were applied for all important processes.



Test setup to verify the positioning process for nozzle plate bonding was used to make a series of ink jet heads to get the statistical information about the process

Gate
3

One hierarchical step higher, at stage 6, the first production systems were integrated. This enabled increase of the batches for testing the system. batches of 10-12 products could be produced twice per hour. This did not only enable pilot production of print heads for testing the system, it also gave feedback about realised quality of the processes.

After integration, the production numbers of the test series numbers were increased using the automated equipment for the final manufacturing process

Gate
2

The printing system was going to be integrated at the site of one of the partners. The production equipment for the inkjet print heads was gathered at a shared production facility on another location over 2000km away. After the site acceptance tests (SAT), all equipment was shipped to the respective locations for integration of the printing system or manufacturing of the inkjet print heads.

Equipment that was made by one of the contractors is shipped after the SAT to the shared production facility

Gate
1

The communicative dynamics of stage D are of a different nature than that of stage B & C. The communication has changed from high certainty discussion to organisational with lots of little uncertainties.

The pressure for technological functionality remained high with the factory acceptance tests coming. With the FAT, all interactions were tested. Though some delays were caused, the many iterations during development had revealed many problems before they could escalate; it has prevented major disappointments.

At the shared production facility all equipment is installed and tested again

Gate
0

Figure 4.10 *Validating product and RMS by going through gates 3 to 0*

4.5 Discussion

The development case for an inkjet system for sign and display printing was considered to benefit from *concurrent system decomposition* and *concurrent gating*. The question arises what would have been the result if another method had been applied. Processes for industrialisation of microsystems are diverse and involve large investments. This makes objective reference expensive and heterogeneous.

4.5.1 Strengths of the Method for Indexing the Development Process

concurrent system decomposition successfully applies AD to enhance the *CμSD framework*. The use of design relations forces the designer to organise the product design and the required process technology, simultaneously combining the domains at the respective layers of decomposition. *Zigzagging* is applied to bring structure in this process; it ensures that requirements are tested in the most optimal order. The procedure continuously enables focus on the right layers and domains as the project evolves. New is the process of *reversed zigzagging*. It not only changes the direction of the *zigzagging* process from bottom to top but also reverses the order through the domains form CA-FR-DP-PV to PV-DP-FR-CA.

The application of AD supports *concurrent gating* for two reasons: (i) *zigzagging* applies the right order in which the tests are performed, and (ii) the design matrices provide the right criteria for testing as the design matrices force the designers to organise CAs, FRs, DPs, and PVs.

4.5.2 Weaknesses of the Method for Indexing the Development Process

The linearity of the *CμSD framework*, during the conceptual phase with this implementation of *functional gating*, may disappoint as much as it does in the V-Model.

It is because the designer is prioritising his work by the magnitude of the remaining risks; when a system is decomposed, the risk at any higher level is the aggregate of all risks at all lower levels. This means that when the last risk at the lowest level is reduced, all four gates may be closed straight after each other in succession. In practice, it seems less of a problem due to eased circumstances; uncertainties of many problems at a lower level may be estimated reliably without knowing them in detail yet. This is the case e.g.: (i) when a similar problem was solved in the past, (ii) when there is a variety of good solutions to address a particular problem, or (iii) commercial off the shelf parts are applied. Anyway, it is not realistic to suppose that the gating function is linear.

4.5.3 Limitations of the Method for Indexing the Development Process

It is still a point of discussion how rigid gating should be. The basic implementation of the Waterfall-Model dictates that the next stage should be started only when its preceding stage is reviewed and verified. This is one of the features that was criticised by Royce in his marked paper (Royce, 1970). It means that the scope of the designer is not far ahead to what's coming. Not looking forward is like an ostrich burying its head in the sand, and in particular during the conceptual phase (gates 7 to 4), it gives risks free rein to surprise the designer. It is unwanted because it increases the chances that closed gates need reopening. It seems better to allow designers to look ahead but force their focus to the largest remaining problem of the project (better explain to the designer what should be prioritised than to tell him what not to do).

Moving from one of the gates 3 to 0 backwards to the conceptual gates 7 to 4 is unwanted and should be prevented. In these cases, something has gone seriously wrong and work may be lost as the conceptual design will need reconsiderations. This problem increases with the hierarchical level of the project in which it takes place (from gate 3 to

4 if something is wrong with parts development, is less problematic than correcting a systems function that causes a fall-back from gate 2 to 5). Failing the FAT or SAT is usually disastrous; it will almost certainly lead to substantial actions for correction in the project and cause according delays.

4.5.4 Opportunities for Further Improvement

In the conceptual phase, boundaries are not stationary, but are influenced by internal and external occurrences. As a result, the project and its targets may change while it is under development. Together with the risk of non-linearity of *concurrent gating*, the gating function could use a further upgrade to make it more robust for these influences. The AD methodology was expanded with a ‘Theory of Complexity’ in 1999 (Suh, 1999; 2005b). This theory of AD could provide a solution for gating of the conceptual phase. This will be investigated in the next chapter.

4.6 Conclusions

The *CμSD framework* combines some of the best features of the V-Modell XT and Axiomatic Design; it addresses a number of the shortcomings of the V-Model but also applies strengths of the V-Modell XT to the Axiomatic Design methodology. As such it tries to combine the best of both worlds. The principle of *Reversed Zigzagging* is introduced to structure the testing procedure of the project. The method was applied to an industrial case; the development of an inkjet printing system. Though the result was generally positive, it also revealed room for further optimisations. Mainly the linearity of the conceptual phases and its gate function may be improved.

CHAPTER 5

APPLICATION OF INFORMATION AND COMPLEXITY IN AXIOMATIC DESIGN TO ENHANCE FUNCTIONAL GATING IN THE CμSD FRAMEWORK*

5.1 Introduction

The *μSD framework*, introduced in Chapter 3, was upgraded to the *CμSD framework* in Chapter 4 by adding functionality to support the concurrent nature of microsystem development. Both *functional system decomposition* and *functional gating* were enhanced using AD to determine project progression based on the capability of modelling the product and process design relations.

In this chapter, *concurrent gating* will be further enhanced by implementation of a measure that is not only based on 'what was achieved so far in the project', like successful decomposition and specification efforts, but also looks ahead at the 'remaining

---

** Parts of this chapter were published in:*

Puik, E. C. N., & Ceglarek, D. (2014). A Review on Information in Design. Presented at the 8th International Conference on Axiomatic Design ICAD2014, Lisboa, (Puik & Ceglarek, 2014a).

Puik, E. C. N., & Ceglarek, D. (2015). Axiomatic Product Design in Three Stages; A Constituent Roadmap that Visualises the Status of the Design Process by Tracking the Knowledge of the Designer, Presented at the ASME IMECE2015, Houston, (Puik & Ceglarek, 2015a).

Puik, E. C. N., & Ceglarek, D. (2016). A Different Consideration on Information and Complexity in Axiomatic Design. In N. P. Suh & A. M. Farid (Eds.), Axiomatic Design in Large Systems: Complex Products, Buildings Manufacturing Systems (1st ed.), (Puik & Ceglarek, 2016).

uncertainties in the project'. The goal is to make the gating process weighted and knowledge based, compared to closing gates with the uncertainty of reopening them in a later stage. This is called 'Intelligent Gating' (Table 5.1). *Intelligent gating* is implemented by application of 'Information in Design', based on the information theory of Shannon and Weaver, the same theory that is applied for information in the Information Axiom. Unfortunately, the information theory cannot be applied without further notice. In the current definition, information is only related to robustness and not to the conceptual stage of design. Therefore, it is investigated if it can be adapted for application over the total breadth of the *CμSD framework*.

Table 5.1 *Chapter 5 adds intelligent gating to the CμSD framework*

Object / Chapters	*Functional system decomposition*	*Functional gating*
Chapter 3	Structured Analysis Design Technique (SADT) in combination with Failure Modes and Effect Analysis (FMEA)	Three different methods: (1) Remaining uncertainties based on the FMEA (2) Qualitative Analysis (QA) with unidirectional coding (3) Qualitative Analysis using a Maturity Grid (MG) with a bi-directional coding scheme
Chapter 4	*Concurrent system decomposition* based on Axiomatic Design	*Concurrent gating* based on the completion of decomposed hierarchical levels (or tested levels)
Chapter 5	Same as in Chapter 4	*Intelligent gating* based on Information in Design

The chapter is organised as follows: Section 5.2 analyses how the mentioned theories can contribute to *intelligent gating* of the *CμSD framework*, and what the key issues are that prevent application. Section 5.3 will demonstrate that it indeed is possible to apply Information in Design as a measure for *intelligent gating*. Section 5.4 implements *intelligent gating* by decomposition of 'Total Information' of AD. Section 5.5

investigates how some specific elements of information could be addressed. Section 5.6, discusses the findings and lessons learned. Finally, Section 5.7 draws conclusions.

5.2 Analysis & Approach

5.2.1 Problem Statement

Though functionality of the *CμSD framework* was significantly enhanced by a concurrent approach of decomposition and gating, *concurrent gating* still has a number of shortcomings. The method could suffer from non-linearity, as was explained in the discussion of the last chapter. This non-linearity causes gates 7 to 4 (Figure 4.7) to be closed quickly after each other when final *zigzagging* secures ‘proof of concept’. Ideally, the gating function would be linear and continuous.

The principle of gating aims to define development stages in terms of its outputs, and the outputs of each stage represent the points along the development path. The current stage establishes the definition of the next one as a basis for further derived definitions (Rook, 1986). Therefore, if gates are reopened, the foundation of a larger set of derived stages is affected, and because of this, a lot of rework may be needed. However, an ideal gating method would register the consequences of such a change. These consequences could be inventoried and weighted against alternative options available to the designer. In this case, it would be possible to make an informed decision about reopening of a gate. This provides more flexibility in the decision-making process than simply restrict reopening of gates.

5.2.2 Current Situation

The last chapter demonstrated how the axioms in AD could be applied for *concurrent gating*. In principle, there are only two axioms, the Independence Axiom and the Information Axiom, typically satisfied in that order. This means that there are only three clearly defined statuses to be recognised for the design: (i) no axioms satisfied, (ii) only the Independence Axiom satisfied, or (iii) both axioms satisfied. Though the conceptual and robustness phases can be gated by application of the axioms in AD (respective satisfaction of the Independence and Information Axioms), a number of three stages does not provide sufficient resolving power for a gating function that can be applied to measure project progression. Therefore, the state of decomposition of the product design, and derived design relations, were used in Chapter 4 to increase the resolving power up to a number of eight stages. The decomposition tree is a good means to investigate which systems and parts of the design are affected if gates are reopened, however if design fallacies need to be corrected, it does not indicate what the consequences are in terms of uncertainties and extra work. For instance, if a gated project decision requires new systems to be developed from scratch, which later appears cause a lot of project uncertainties, the designer's initial focus will be on addressing the uncertainties. If gates could be interpreted softer, like a guideline instead of a binding instruction, the designer's perception would be open for alternative options that may be better for the project as a whole, e.g.: an overseeable change of a gated project decision could provide the opportunity to apply more proven technology. In Rook's influential paper, in which he proposes the V-Model, he mentions that exploration of the next stage is usually required before the current stage can be completed, and that new understandings may overrule old ones as the knowledge of the designer increases during the development project. Unfortunately, in practice, managers tend to be reluctant in reopening gates (Anitha et al., 2013).

5.2.3 Key Limitations

The critical shortcomings of *functional gating* are:

- Gating based on the level of decomposition is not very linear. Errors that are found in the later stages of design may need reconsideration at higher levels of decomposition. Closing gates may provide an unjustified sense of security;
- Gating is not based on true uncertainties in design, but on the level of decomposition in design, that may be unverified;
- Gates are often implemented with Boolean nature; this excludes many alternative solutions that could be integrally more efficient for the project.

5.3 Methodology to Implement Intelligent Gating to the CμSD Framework

As it is difficult to derive measures of uncertainty from the level of decomposition of design, another directive should be found that has better capability to determine the success rate of a microsystem development project. Ideally, it would allow for keeping gates open till alternatives can be explored by weighting the uncertainties of the options. As explained in the introduction, this way of gating is named *intelligent gating*. The word 'intelligent' is used because the process of closing gates has an inversed relation to remaining uncertainties in the design. Though the axioms in AD in principle are suitable for such a gating process, as shown in Chapter 4, it does not provide the targeted continuous measure. Instead, the decoupling process of the design matrix seems to have a Boolean character; it is decoupled or not (uncoupled being a special state of decoupling).

However, as the targeted condition indeed is Boolean, the process of decoupling advances gradually from ‘many design relations coupled’ via ‘few design relations coupled’ to ‘decoupled’. If the process of decoupling may be considered to have a continuous, analogue course it would enable continuous monitoring of development progression in the conceptual phase of design. However, it needs investigation if ‘the state of independence of the design’ may be considered as a continuous index.

The solution, as proposed here, is based on ‘Information in Design’. Information in design is related to information in the communications theory as defined by Shannon (Shannon, 1948). An attempt to apply information in design, as a measure of independence of the design, was not made before. To do so, the definition of Information in AD needs to be extended.

Subsection 5.3.1 brings existing kinds of information within AD together from literature. Subsection 5.3.2 demonstrates that ‘Useful Information in AD’ is responsible for satisfaction of the FRs. Subsection 5.3.3 demonstrates that information as a measure indeed can be applied to monitor the independence of the design. Finally, Subsection 5.3.4 explains how the key limitations, as inventoried in the previous section can be addressed by this method.

5.3.1 Current Status on Information in AD

Information in AD is derived from the information theory using a measure of ‘Boltzmann Entropy’ according to Shannon & Weaver (Shannon & Weaver, 1949; Suh, 1990). In this definition, Information is related to chaos in the design; a well-structured and understood design contains little information, therefore a chaotic and poorly understood design will have a large information content. Information uses a logarithmic representation as introduced by Hartley to make information additive instead of multiplicative (Hartley, 1928). According to the information theory, information is inversely related to the probability of success of a goal being met. In AD, the goal is met when DPs are causing FRs and constraints to be within tolerances.

The total amount of information in a design is called 'Total Information'. Total Information was split into two parts, 'Useful' and 'Superfluous' information (Suh, 1990). *Useful information* relates solely to the satisfaction of particular tasks. These tasks are specified in terms of the FRs and constraints. *Superfluous information* does not affect the relation between DPs and FRs. As information is additive due to the logarithmic function, the following breakdown of *total information* can be made (Figure 5.1).

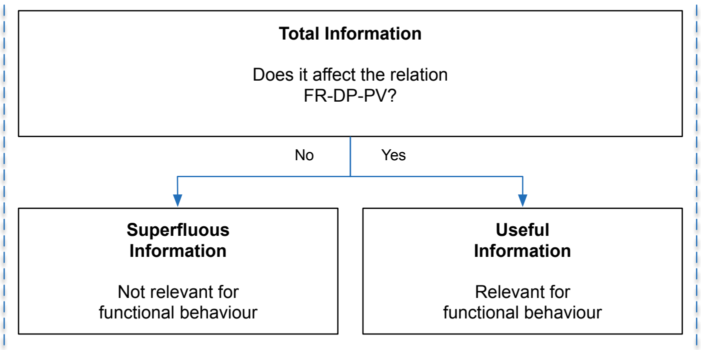


Figure 5.1 *Breakdown of total information*

Every product design in progress will have an 'Information Content'. The information content is a measure of the probability of success of achieving the specified FRs (Suh, 1990). The probability of success is obtained by considering all FRs to be satisfied in their mapping to DPs. Then, the joint information content is determined by taking the sum of all individual 'Informations'. The result gives the information content of the design. The Information Axiom dictates that the information content of a system should be minimised, and thus maximising the probability of FRs to be satisfied. This means for the information breakdown of Figure 5.1 that *superfluous information* is no information from the axiomatic perspective and may be ignored. On the other hand, *useful information* indeed must be properly eliminated from the design because elimination of *useful information* from the design is a prerequisite to satisfy all FRs.

5.3.2 Elimination of Useful Information from a Design

As elimination of *useful information* leads to satisfaction of FRs, the question is how this can be achieved. A first and most straightforward hypothesis would be to assume that *useful information* could be eliminated by satisfaction of the Information Axiom.

Statement: Elimination of *useful information* cannot be guaranteed by satisfaction of the Information Axiom alone.

Proof: According to good-AD-practice, the information content of a design can be calculated with (Suh, 1990)

$$I = log\left(\frac{system\ range}{common\ range}\right) \tag{5.1}$$

and if it concerns multiple FRs, the different information contents should be summarised. Satisfaction of the Information Axiom can only take place if all system ranges are within the common ranges (Suh, 1990; ElMaraghy et al., 2012). However, this does not satisfy the Independence Axiom; the design needs to be independent too. With only addressing the Information Axiom, the design could therefore still be a coupled design, and some FRs may not be satisfied. If there are unsatisfied FRs, *useful information* is not completely eliminated; the statement is true. Elimination of *information in design* as addressed by the Information Axiom will cause a design to be robust by guaranteeing overlap between system and common ranges, but it does not guarantee independence of design.

This investigation leads to understanding that *useful information* cannot be eliminated by satisfaction of the Information Axiom alone. This implies that a certain part of *useful information* is addressed by the Independence Axiom and that this axiom indeed is related to information. The question arises what kind of information this is.

In the books about AD (Suh, 1990; 2001; 2005b), the Independence Axiom was never associated with information according to Boltzmann entropy, neither was imaginary complexity. However, imaginary complexity was considered to have a stochastic nature for some problems (Suh, 1999; 2005b). Further, the book about complexity shows a number of examples that are clearly explaining how knowledge of the designer is related to the quality of design outcomes. One example is a case where the designer does not realise that the design is a *good design* with a decoupled matrix. The designer uses trial and error to test many different sequences of DPs to satisfy the FRs, needing to test n! sequences, thinking that the design is quite complex. This situation describes exactly the characteristic behaviour of Boltzmann entropy in a design.

5.3.3 Information Related to the Independence Axiom

The question is if the Independence Axiom is related to Boltzmann entropy and if this is the case, how it is embedded.

Statement 2: The Independence Axiom is related to Boltzmann entropy.

Proof statement 2: The information theory of Shannon and Weaver states that information is ‘related to the number of alternatives that remain possible to a physical system’ (Shannon & Weaver, 1949). The ‘number of alternatives’ indicates that the current design is not fully restricted within its delineated boundaries. Further, Weaver explains, ‘information does not relate to what the design is as much as what the design could be'. In an incomplete design, many alternatives in which the design can manifest itself are still open. Only a certain amount of these alternatives lead to satisfaction of FRs. The other alternatives lead to unsatisfied FRs. For an ignorant designer, this process has a stochastic nature; it increases Boltzmann entropy and as a result also information in design. Therefore, not only a lack of robustness causes information in design, but also every lack of boundaries that are needed to restrict the system to operate correctly.

Example 1: In a fully robust system, the Information Axiom is satisfied because the system ranges match the common ranges. If the designer lacks understanding of the design, and hereby the design matrix is coupled, he will be surprised of the inexplicable system behaviour when he tries to set up the system. To the designer, the system seems to operate randomly until he gains knowledge of the system. What first appeared random, shows to behave in a structured manner, but only after acquisition of the appropriate knowledge.

Example 2: A designer overlooks a DP during the design process and as a result he assumes that the design matrix is decoupled conforming good-AD-practice. In a later stage this DP, that should have been properly 'fixed', appears to drift away from its initial value. The drifting DP may cause coupling of the design matrix and randomly deprives satisfaction of the FRs.

Explanation: The statement claims that information is not solely restricted to the Information Axiom. Example 1 explains that the dissatisfaction of the Independence Axiom may introduce features with a stochastic nature and therefore it also deals with information in design. This information is related to missing structure of the design that is a requisite to make a design independent. Gell-Mann & Lloyd call this missing structure a ‘lack of regularities in the system’. The lack of regularities increases entropy in the system and ‘the smaller the entropy, the less spread there will be among the entities that follow these regularities’ (Gell-Mann & Lloyd, 1996). A lack of regularities in the design will increase its chaotic behaviour and thus increase information. The definition of well-chosen FRs, the process of selecting matching DPs, decoupling the relations between FRs and DPs, making sure that all DPs are relevant, and ensuring that all relevant DPs are known, are all regularities that contribute to a more predictable behaviour of the design and hence they eliminate information from the design.

Example 1 and 2 can be clarified further by experiments that were described by Shannon and Brillouin (Shannon, 1948; Brillouin & Gottschalk, 1962). This experiment studies the transfer of a message in the English language over a telegraph line. The total character set exists of 26 characters of the alphabet and a space between words. Initially, the transfer per character is studied when no a priori knowledge of the English alphabet is present. The information content for all characters is the same and is calculated at 4.76 bits. This number can be roughly confirmed when realising that five bits of information give a total of $2^5 = 32$ combinations; so, a total of 27 combinations are expected to come just under five bits. For the second experiment, knowledge is made available that the a priori probability of occurrence of the characters in the English language is not equally distributed; e.g. the space and the character ‘E’ appear more frequently than others. Availability of this knowledge reduces the total information needed to transfer characters. Reconstruction of a corrupted message can be performed on a basis of statistical knowledge of the English character distribution instead of mere coincidence, thus increasing the chance on a successful outcome. The information per character indeed appears to be lower and is determined to be 4.03 bits. For the third experiment, the knowledge of the English words and grammar is made available to the receiving end. This knowledge helps rejecting unsuccessfully reconstructed messages and in this way further increasing the chances on a successful reconstruction of the message. Depending on the situation, the actual amount of information is estimated to be between one and two bits. This example clarifies that every type of knowledge based condition, imposed on the possible freedom of choice, immediately results in a decrease of information. The same applies to the synthesis of a product design where every definition of an FR and its DP limits the possible variation of the behaviour of the system and thus reduces information or entropy in the design. Adding regularities in a design decreases information; it quantifies the extent to which an entity is taken to be regular, non-random, and hence predictable. For AD, this is not only limited to the Information Axiom since the description of rule-based features for the 'Structure of the Design' also adds-up to the predictability of the product design and therefore also reduces information. Finally, decoupling of the design matrix is a process that eliminates wrong outcomes in a structured manner. The remaining stochastic process has no other options than to operate within the remaining boundaries of the system. In a *good design*, all remaining boundaries lead to a successful outcome and thus satisfaction of FRs.

5.3.4 How Intelligent Gating Addresses the Key Limitations

Uncertainty in design can be expressed by information in design as a measure. This is as well the case for the Independence Axiom as the Information Axiom. Information in design is additive; the uncertainties of multiple issues in the design may be added to a score. This score represents the total amount of remaining uncertainties in the design. It can be applied to compare alternative solutions for design problems and selecting the ones that give the best chances to satisfy the FRs. The application of information in design could address the key issues that were inventoried in Subsection 5.2.3:

- Due to the additive nature of information in design, it may be expected to be at least reasonably linear;
- If *functional gating* can be based on information in design, it is also based on uncertainties in design;
- Since information can have any value from 0^+ to ∞, it can be applied as a continuous measure.

In the next section, information is decomposed further to make it applicable for *intelligent gating*.

5.4 Total Decomposition of Information

This section decomposes information in AD according to the analysis of the former paragraph. Subsection 5.4.1 starts with *useful information* since this is the highest kind of information relevant for satisfaction of the FRs. It also defines a new kind of information that is related to the Independence Axiom and it renames the current definition of information in AD to prevent confusion. Subsection 5.4.2 completes the decomposition and defines two sub-kinds of information in design. Section 5.4.3 summarises the expanded definition of information in design.

5.4.1 Decomposition of Useful Information

The reasoning that information is in principle related to both the Independence Axiom and the Information Axiom makes *useful information* the aggregate of these kinds of information conforming

$$I_{Useful} = I_{Related\ to\ Axiom\ 1} + I_{Related\ to\ Axiom\ 2} \tag{5.2}$$

where both kinds of information are the result of irregularities in the design; the Independence Axiom dealing with the structure of the design, and the Information Axiom dealing with robustness in the design. The information related to the Independence Axiom disappears when the design matrix is decoupled and the information related to the Information Axiom disappears when a design becomes robust. As a result, *useful information* measures the lack of total regularities and therefore the 'Ignorance of the designer'; this is exactly according to the conclusion of Gell-Mann & Lloyd, which leads to the following equation

$$I_{Useful} = IGN_{Designer} \tag{5.3}$$

where $IGN_{Designer}$ is the total ignorance of the designer under proviso that there was enough time to apply the designer's knowledge to the design. As indicated in Equation (5.2), $I_{Related\ to\ Axiom\ 1}$ is a different kind of information as defined for the Information Axiom in AD. Consequently, a new definition is needed to differentiate these two kinds of information.

Definition 1: The information caused by the irregularities in the structure of the design is called 'Unorganised Information' since it only exists when the design matrix has not yet been organised. *Unorganised information* is information that resides in the system because not all FRs, DPs and PVs are known and/or the design matrix is not uncoupled or decoupled.

Definition 2: The information that is related to robustness of the design, which is traditionally indicated by the Information Axiom in AD, is further referred to as 'Axiomatic Information' ($I_{Related\ to\ Axiom\ 2}$).

The breakdown of total information as shown in Figure 5.1 can be expanded by applying this definition and is shown in Figure 5.2.

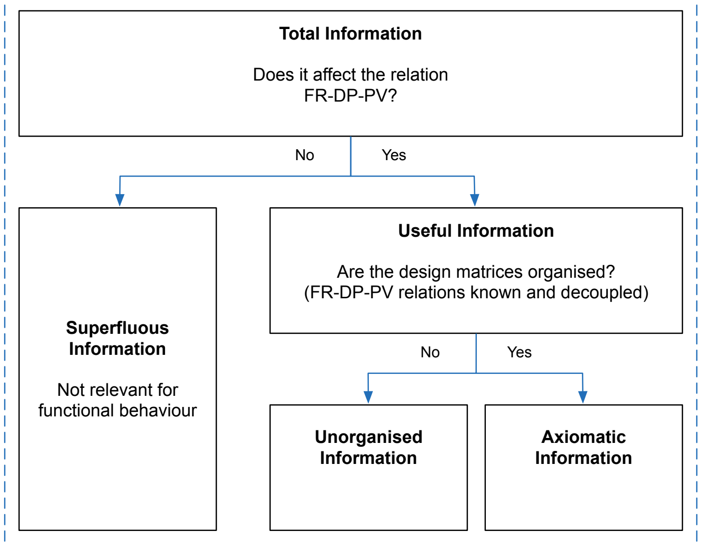


Figure 5.2 *Expanded breakdown of total information*

Unorganised information is determined by organisation of FRs and DPs in the design matrix and their decoupling but it has no impact on the common range of the system, they are situated at the same hierarchical level.

5.4.2 Decomposition of Unorganised Information

If a design matrix is properly developed, hence, all FRs and DPs are known and decoupled, only *axiomatic information* is left in the system. *Axiomatic information* typically gives feedback to the designer about his lacking knowledge. If a system range does not satisfy the design range, the designer will notice that a particular FR is unsatisfied. The designer will also know what DPs are responsible for the problem because of his understanding of the design matrix. This is not the case for *unorganised information*; lacking knowledge does not automatically come to the surface and information may remain hidden. The first example of Subsection 5.3.3 shows a situation in which the designer does know that he is missing knowledge to set up the system. In this case, a design shows inexplicable system behaviour to the designer, which warns the designer that he does not yet fully understand the design. The second example shows a different case. The designer misses a DP, but is not aware of this problem. His lacking knowledge is essential to prevent malfunction in the future, when changing circumstances that are not clear to the designer enable the DP to cause problems. Missing knowledge hinders the designer to make the right choices for the essential regularities in a design and therefore, *unorganised information* may manifest itself in at two different appearances; 'Unrecognised' and 'Recognised'.

Definition 3: 'Unrecognised Information' is a part of *unorganised information* that is not recognised by the designer and therefore remains hidden in the system. It is addressed by finding the right FRs, DPs, and PVs.

Definition 4: 'Recognised Information' is the part of *unorganised information* that is recognised by the designer but, as the knowledge to address the problem is lacking, it cannot yet be eliminated from the design. It is addressed by preparation of the design matrix and decoupling it.

The next paragraph will give an overview of all kinds of information that are covered in this chapter.

5.4.3 Overview of Information in Design

This chapter has explained seven kinds of information. An overview of these different kinds of information are shown in Figure 5.3:

- *Total information*; the total information content of the design (full entropy of the design);
- *Superfluous information*; information that does not affect the relation between FRs and DPs;
- *Useful information*; the part of *total information* that affects the relation between FRs and DPs;
- *Axiomatic information*; the part of *useful information* due to a discrepancy in design ranges and system ranges according to the original definition of information for the Information Axiom;
- *Unorganised information*; a specific kind of *useful information* that is caused by insufficient relational regularities of the design (FRs and DPs). *Unorganised information* is related to the Independence Axiom;
- *Unrecognised information*; a specific kind of *unorganised information* that is not recognised by the designer and therefore remains unaddressed;
- *Recognised information*; a specific kind of *unorganised information* that is recognised by the designer but the knowledge to address the problem in an appropriate manner is lacking.

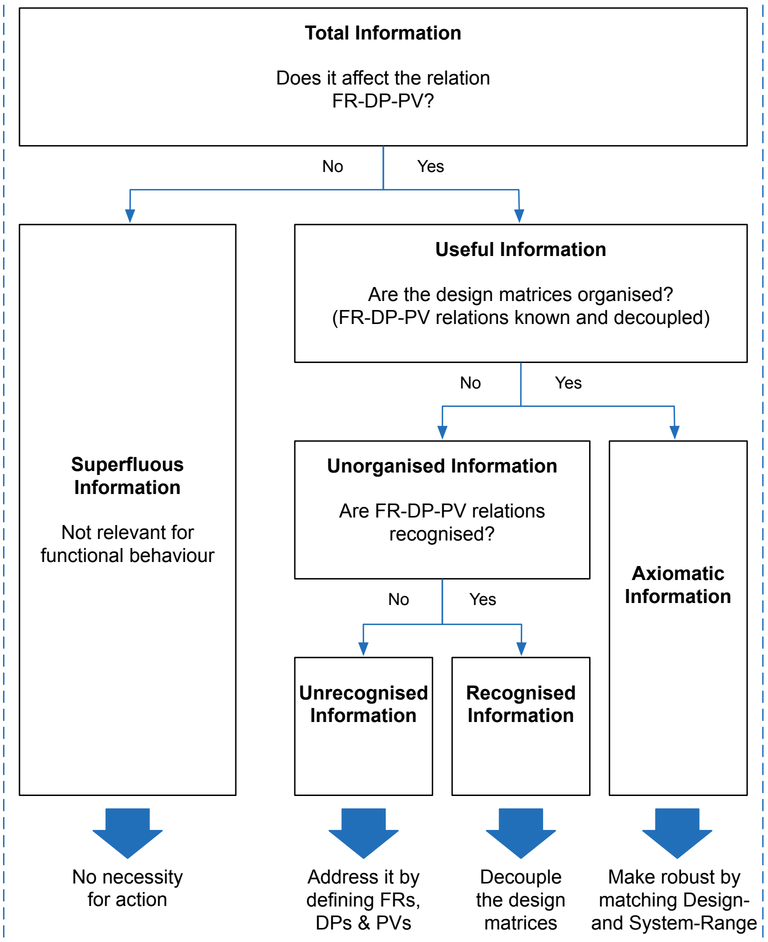


Figure 5.3 *Final breakdown of total information*

Figure 5.3 completes the break-down of information in AD. A number of three kinds of information should be addressed to ensure a *good design*:

- *Unrecognised information* should be addressed by completion of the design relations and decoupling as complete understanding of the design relations leaves no room for ignorance of the designer. Once *unrecognised information* is recognised, it instantly changes to *recognised information*;
- *Recognised information* is known to the designer and should be addressed conforming good-AD-practice;
- *Axiomatic information* is eliminated by matching the system and the design ranges.

5.5 Application of Information in Design for Intelligent Gating

The former section has determined three kinds of information in design that together cover the whole development process. These kinds of information have a continuous character. The next step is to build an object for the *CμSD framework* that provides in project monitoring and implements *intelligent gating* in a way that is: (i) continuous, (ii) uncertainty based, (iii) linear, and (iv) concurrent.

The application of information in design has impact on the way the project is monitored. This impact changes the way gating is implemented in the project. So far, gating was based on the status of the parts of the design that were decomposed and understood by the designer. The order in which the project was gated was determined by the hierarchy of the V-Modell XT; Project-Product-Systems-Parts. The portion of the project that was decomposed, and of which the design rules were known was used as the foundation for the actual project status. When information in design is applied, the order changes. Information is related to the portion of the project of the design that is not understood. This is because insufficient regularities were applied and because of that there are remaining uncertainties and thus information in the design. Instead of basing the gating process on ‘what the designer knows’, it will be based on ‘what the designer does not know’. The process of *intelligent gating* works the other way around compared to *functional and concurrent gating* that were used so far. In this process, there is an essential role for *unrecognised information* because it can have a large impact on the design process.

Recognised information and *axiomatic information* are related to problems that are known to the designer; the designer is aware of these problems but they were ‘just’ not solved till now due to reasons of priority and available time. What should happen to the design can be inventoried, prioritised, and subsequently addressed. However, *unrecognised information* is by definition not known to the designer (since it is unrecognised so far). As such, it cannot be addressed by the designer as well. The difficulty is that *unrecognised information* indeed is present in the design and should be addressed at some point. A second difficulty is that the impact on the design, when *unrecognised information* becomes clear to the designer, could principally not be overseen in advance. The impact may be large and the design may need big changes to address the problem. *Unrecognised information* is a designer's nightmare; it will confront him at a surprising moment in time and may have impact on the structure of the design. Therefore, *unrecognised information* is best tracked down and addressed as soon as possible.

5.5.1 How to Track Down and Address Unrecognised Information

To investigate what can be done to find *unrecognised information*, the 'Cynefin Framework', is applied. The Cynefin framework is an analytical decision making framework that was developed by Snowden at IBM (Kurtz et al., 2003).

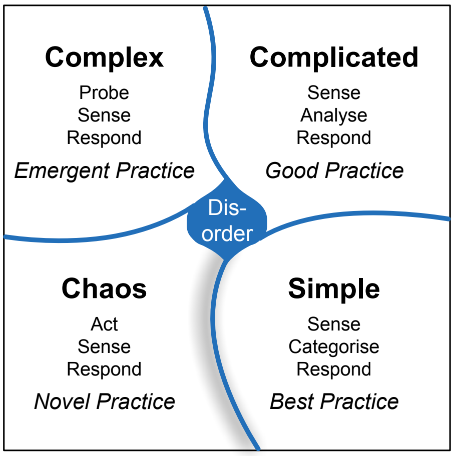


Figure 5.4 *The four contexts of the Cynefin framework*

Cynefin relates to a ‘place of multiple belongings’. The Cynefin framework originated in the practice of knowledge management. Snowden calls it a ‘sense-making’ model where data precedes framework and patterns emerge from the data instead of the other way around. It consists of four contexts, basically fields of action, in which an organisation or system can be found, and a fifth space when the actual context is unknown (Figure 5.4). Knowledge is in the Cynefin framework the most important parameter to determine the context where an organisation, system, or problem is currently located. When knowledge is acquired, the context changes. Appendix D explains the contexts of the Cynefin framework for broader understanding.

For *unrecognised information*, the ‘Complex Context’ of the Cynefin framework is of importance (Kurtz et al., 2003); it is the domain of complexity theories, which study how patterns emerge through the interaction of many entities. Emergent patterns can be perceived but not predicted; this phenomenon is named ‘Retrospective Coherence’. In the *complex context*, structured methods like many existing methods for systems engineering are ill prepared. Once a pattern has stabilised, it can be understood and even may appear logical, but there are many of such patterns. Since patterns are connected with each other, they may repeat for a while, but it is never sure if they continue to repeat, because the underlying sources of the patterns are not open to inspection. This context accurately describes the world of a designer that is developing new technology.

The decision model in the *complex context* is to create ‘Probes’ to make the patterns or potential patterns more visible before taking action. The patterns are observed or ‘Sensed’ and responded to by: (i) stabilising desirable patterns, (ii) by destabilising those we do not want, and (iii) by seeding the *complex context* so that patterns we want are more likely to emerge. This procedure is continued till more and more patterns are understood and can be connected to a larger whole.

Creating probes to make patterns or potential patterns visible is exactly what is the basis of the *CμSD framework*; it comes down to performing safe-fail-testing to the design and as such understand the patterns. Connecting the patterns is what AD does, by combining separate design relations to design matrices, a broad understanding of the design is acquired. Basically, *unrecognised information* is only found in these ways:

- Testing the design to: (i) study what is necessary to stimulate behaviour of functions as expected, (ii) study what is necessary for functions to stop acting up negatively, and (iii) study what is necessary to bring the design to a higher level;
- Completing the models of product and processes to understand how it operates in detail.

By application of these methods, unrecognised information, may be found though only complete understanding and a full set of regularities would guarantee this. *Unrecognised information* changes to *recognised information* when it is found and may then be addressed by applying the right design relations.

5.5.2 Application of Information in Design for Intelligent Gating

With the current knowledge, the outline for *intelligent gating* is defined:

- Elimination of *unrecognised information* from the design has the priority over all other kinds of information;
- When *recognised information* is eliminated too, the design is uncoupled or decoupled and the Independence Axiom is satisfied and proof of concept is accomplished;

- When *axiomatic information* is eliminated from the design the system may be considered robust;
- The information related to the axioms, *unorganised information* for the Independence Axiom and *axiomatic information* for the Information Axiom may be considered to be continuous and may gradually be addressed till it is eliminated from the design;
- All kinds of information are additive. As such the information in the product design may be added with the information in the process design. The aggregate information is a measure for the total project status.

Table 5.2 shows when and how these kinds of information are addressed during the product development process.

Table 5.2 *Overview in which phase unrecognised, recognised, and axiomatic information in design should be addressed*

Phase / Information	Exploration	Conceptual	Robustness
Unrecognised Information	Find as much <u>unrecognised information</u> as possible Find FRs, DPs, and PVs	Do everything to make sure no *unrecognised information* is left	Stay alert for things that are not understood
Recognised Information	Start addressing elementary problems and new design artefacts first	Address all remaining <u>*recognised information*</u> by decoupling the design	Guard decoupling
Axiomatic Information	No need for action	Inventory key tolerances	Make all design relations statistically robust by addressing all <u>*axiomatic information*</u>

5.6 Discussion

Feasibility for applying information in design for *intelligent gating* required investigations in the information theory which underlies AD. Based on the definitions of Shannon & Weaver, Brillouin, and Gell-Mann & Lloyd, the statement that the Independence Axiom deals with information may be considered to be true. It leads to a new definition of information within AD. Accepting the definition means that *useful information* is the basis for AD since it covers every aspect that is needed to satisfy the FRs. But it also means that both the Independence Axiom and the Information Axiom are addressing information in design. However, both axioms address different kinds of regularities and therefore deal with different kinds of information; regularities in the product design, that deal with conceptual design issues, are different from the regularities that deal with robustness.

5.6.1 Strengths of this Approach

The analysis of information in design proves that information is indeed related to the Independence Axiom. The satisfaction of the Independence Axiom is basically a discrete process as it will be decoupled step by step, and there are only a limited number of design relations. However, the information theory may be applied for its foundation, which makes satisfaction of the Independence Axiom in principle a continuous process. As such, it can be used as a continuous measure to follow the project progression as it evolves. The related kind of information is defined as *unorganised information*, because it relates to the project stadium in which the design matrices are not yet organised.

Due to the additive nature, the amounts of information that are caused by project uncertainties can be added to form a *total information* content of the overall design. It may be expected to be at least reasonably linear. As such, it can be applied for *intelligent* *gating*; which in this case is based on uncertainties in the design. It approaches the design process the other way around; gating is not based on what was accomplished so far in the project, but it is based on the difficulties that can be foreseen in the remainder of the project. It is not said that this approach prevails over the other, but it might help to focus on *unrecognised information* more effectively; as this is the most important kind of information to address, there is room for enhancement in ways to address it.

5.6.2 Weaknesses of this Approach

Basically, the *intelligent gating object* is not implemented yet, but instead of this, a framework was developed. Gating should rigidify potential change of decisions that are considered well-founded and that qualify as boundaries for the remaining branches in the decomposition tree. The question is how the gating function should be addressed in *intelligent gating*. One option is to keep gates open if branches have many uncertainties, as changes are more likely to take place for these cases. Another way is to indeed apply gating but, while doing this, distribute as much freedom for design to branches with large uncertainties and less freedom for branches that are relatively certain. In any case, the focus is forward to the remaining development process, and as such focuses on what is to come, instead of what seems accomplished.

As implemented now, the gating process has only three stages indicating that respectively exploration, conceptualisation, and robustness are completed. In between these stages, the continuous measure of information in design may be applied. However, manual determination of all uncertainties in the design, and calculating their information contents to summarise them, would be quite laborious and was not tested yet. Better ways to determine the information content of the design should be developed.

5.6.3 Other Considerations

Generally, the gating function can be approached in two ways: (i) the designer is focussed on what so far was well organised in the design (what is good and does not need to be changed) or, (ii) the designer is focussing forward (what are the remaining risks to address). With the decomposition of information in design, as introduced in this chapter, both are supported by AD. The former is addressed by the *zigzagging* process, and the latter is addressed by inventorying remaining information in design. Both methods have particular advantages and disadvantages. In the first situation, the designer gets a good indication what was accomplished so far in the project, however, if gates need reopening the project is likely to be set back in time. In the second situation, the designer has an indication of the remaining problems till the end of the project, which is convenient for planning project resources, but how can the information in design be determined if the project contains many new technological artefacts that might contain unrecognised information. Ideally, a combination of both is applied, in which traditional gating secures decisions that have been made, but where substantial risks may be foreseen and adequately addressed or avoided.

Another consideration is the choice for hard or soft gating. Hard gating defines clear strategies for development of the next stages but could lead to less optimal project choices since the project rigidifies due to the inflexible decisions. Only a single hierarchical level is under investigation which provides a good overview of the project (though it may be unsubstantiated confidence). Soft gating is flexible and keeps options open to take the best possible solutions. However, project control is complicated because there is less uncertainty about the project course and, as a complication, the designers risk losing focus on specific tasks to complete. Moreover, it is difficult to oversee more than a few hierarchical levels. As such elementary advantages of the process of decomposition, in order to gain overview and focus, are easily lost.

5.6.4 Opportunities for Further Improvement

Find an intelligent implementation for *intelligent gating*, as explained in the former subsection.

To increase adoption in more industrial areas, the method could benefit from a low threshold and efficient user interface e.g.; the method would benefit from a way to visualise project progression and especially what may be expected of the remainder of the project.

5.7 Conclusions

This chapter showed a framework for *intelligent gating* to enhance the *CμSD framework*. *Intelligent gating* does not investigate what was accomplished in the project, but instead it looks forward into the future by evaluating the remaining project risks. A measure of uncertainty for the many artefacts that are still to be addressed form the input for the gating process. It is based on the information theory of Shannon that also forms the basis for information in Axiomatic Design. Information in design is additive and as such it can be applied to determine the overall risk of the project.

The chapter has developed the framework for *intelligent gating*. Some aspects remain open e.g., if *intelligent gating* should be combined with elements of *concurrent gating* to get the best of both worlds. Also, if the gating process should be implemented ‘hard’ (Boolean) or ‘soft’ (flexible but not as clear) is an issue for further investigation.

PART 2: APPLICATIONS OF RISK BASED, CONCURRENT SYSTEMS ENGINEERING MODELS FOR MICROSYSTEM DEVELOPMENT

CHAPTER 6

APPLICATION OF INFORMATION IN DESIGN TO VISUALLY MONITOR THE MICROSYSTEM DEVELOPMENT PROCESSES AND INCREASE UNDERSTANDING AS IT PROGRESSES*

6.1 Introduction

This chapter builds further on Chapter 5. It applies the measure that is based on information in design to realise an 'Axiomatic Maturity Diagram' that can be applied to visualise the project. The capabilities of the Axiomatic Maturity Diagram are to: (i) monitor project progression to follow the targeted strategy, (ii), find and analyse errors in the design process, and (iii) plan and execute effective recovery from errors.

The chapter is organised as follows. Section 6.2 defines the problem statement, inventories the current situation in the field, and evaluates the key limitations. Section 6.3 presents a methodology to visualise the project based on information in design. Two cases, that are described in Appendix E, are evaluated in Section 6.4. Finally, Section 6.5 discusses the results and Section 6.6 draws conclusions.

---

* *Parts of this chapter were published in:*

Puik, E. C. N., & Ceglarek, D. (2014). A Theory of Maturity. Presented at the 8th International Conference on Axiomatic Design ICAD2014, Lisboa, (Puik & Ceglarek, 2014b).

Puik, E. C. N., & Ceglarek, D. (2015). The Quality of a Design Will Not Exceed the Knowledge of Its Designer; an Analysis Based on Axiomatic Information and the Cynefin Framework. Procedia CIRP, 34, 19–24, (Puik & Ceglarek, 2015b).

6.2 Analysis & Approach

As modern development processes are complex and comprehensive, the general goal of manager and designer is to build understanding of (their part of) the project. Uncertainties in the design lead to uncertainties in project execution and disrupt the relationship between project investments and project progression; not knowing where the project stands, makes managers and designers uncomfortable. *Intelligent gating* is expected to increase understanding as it may reveal how uncertainties in the project are developing. Especially when the project status and its progression can be visualised, as a universal language, it enables monitoring of the project risks and serves as a platform for discussion.

6.2.1 Problem

Chapter 5 has shown that the axioms are both related to information in design and because of this, the conceptual status of the project and its robustness may be seen as continuous indices. Table 6.1 inventories their relations.

Table 6.1 *Relation between information in design and the axioms*

Phase Related to	Exploration	Conceptual	Robustness
Axiom	Address Independence Axiom, starting with definition of FRs, DPs, and PVs	Complete satisfaction of the Independence Axiom needed to complete this phase	Complete satisfaction of the Information Axiom needed to complete this phase
Kind of information	Address *unrecognised information* as soon as possible and as thorough as possible	Address *unorganised information (aggregate* of *unrecognised* and *recognised information)*	Address *Axiomatic Information*

There has been discussion in literature if the axioms may be considered independent from each other. In Appendix C, this is investigated; the conclusion is that the axioms are independent, and they address different kinds of information. However, the Independence Axiom may be disruptive to the Information Axiom. To address the disruptive behaviour, the guidelines of AD advise to address the Independence Axiom first, followed by the Information Axiom. The reason is that the Independence Axiom determines the design matrix, and by doing this, it sets the boundaries for the Information Axiom. The boundaries are the design matrices and the design relations. Note that for concurrent design of microsystems it concerns both matrix [A] and [B] (Product design matrix and the process design matrix). Setting the design matrices determines the structure of the product and process design, making sure it is a 'Viable Design'. Optimising the design relations by reducing the statistical variance, making it statistically robust, turns the *viable design* into a *good design* and the design relations operate as they should. From this perspective the more general concept arises that the Independence Axiom is about 'doing the right things' and the Information Axiom is about 'doing things right' (Puik & Ceglarek, 2014a). Though these statements are not meant to be inexhaustible, they contribute well to general understanding of how these kinds of information address the product design. The method for visualisation should take this order into account.

6.2.2 Current Situation

In the current situation, the axioms are typically applied as intended. The order to address the Independence Axiom before the Information Axiom is a standard guideline of AD. The matrix, though it originates from mathematical analysis, does have a visual function; not only are coupled, decoupled, and uncoupled matrices easily recognised due to their represented shape, the matrix also gives a very quick insight in the complexity of a product design. However, these are only momentarily statuses of the design, it does not show how the project evolves.

6.2.3 Key Limitations

The key limitations are divided in three categories: (i) currently there is no visualisation of project progression based on the gating function, (ii) there is little ability to prove insight to different kinds of project dynamics, and (iii) there is no visualisation of errors in the design and how to recover:

(i) No visualisation of project progression based on the gating function:

- The ability to visualise conceptual and robustness phases of product design;
- The visualisation of the absolute status of progression the project.

(ii) Little ability to prove insight to different kinds of project dynamics:

- What is the most efficient development path in terms of investment (SMEs);
  - Optimised development path for project lead time (microsystems, semi-conductor industry);
  - Lowest chance for development errors (safety systems, medical industry).

(iii) No visualisation of errors in the design and how to recover:

- What are the consequences of design errors;
- What are the alternatives for recovery?

6.3 Methodology to Visualise Progression of the Development Process Based on Information in Design

6.3.1 The Axiomatic Maturity Diagram

The Axiomatic Maturity Diagram is based on the information content in a product design as represented by the Independence Axiom and the Information Axiom (Puik & Ceglarek, 2014b). The diagram, shown in Figure 6.1, uses two axes, one for each axiom, plotting the degree in which the axioms are satisfied. The diagram in itself has no axiomatic properties but it takes its name from the fact that it applies the axioms as premises.

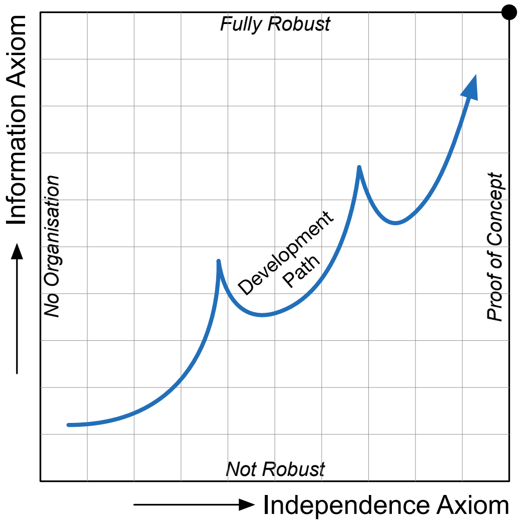


Figure 6.1 *The Axiomatic Maturity Diagram.*
The horizontal axis plots the Independence Axiom, the vertical axis plots the Information Axiom. The development path is arbitrary

The horizontal axis is the 'axis of organisation' starting at 'No Organisation' and ending with 'Proof of Concept'. Proof of concept indicates that the product design is a *viable design*; the design matrix is decoupled and therefore *unorganised information* has become equal to zero. As was shown in Figure 5.3, this implies that both kinds of sub-related information (*unrecognised information* and *recognised information*) have been eliminated. The vertical axis represents robustness of the design from ‘Not Robust’ to ‘Fully Robust’. As was explained in Chapter 5, a fully robust system implies that the *axiomatic information* has become equal to zero (the traditional information in AD coming forth from a bad common range). The lower left-hand corner indicates a high level of ignorance; the designer has little knowledge how to satisfy FRs with his DPs and therefore the ‘Axiomatic Maturity’ is low. The upper right-hand corner shows low information content and maximum probability of FRs being satisfied. This is the area of high axiomatic maturity. Development of products starts in the lower left-hand side and moves to the upper right-hand side. Products are fully mature when they reach the upper right-hand corner of the Axiomatic Maturity Diagram, as marked with a dot. The diagram is plotted in Figure 6.1. The shown development path is arbitrary. The axes of the Axiomatic Maturity Diagram are swapped in comparison to the real and imaginary axes in the complexity diagram of (Suh, 2005b). Two reasons apply to deviate from that definition; firstly, because the Independence Axiom and the Information Axiom are simply plotted in that order, and secondly because the level of independence, as set by the Independence Axiom, never moves backwards (as long as no knowledge of the designer is lost, it will typically increase). By choosing this way of plotting, the path of evolving maturity takes the form of a mathematical function. This makes reading the Axiomatic Maturity Diagram more natural. Note that the information in design is inversely related to the satisfaction of the Axioms; information is gradually reduced as the axiomatic maturity increases (from lower left to upper right).

6.3.2 Presumed and Legitimate Position in the Axiomatic Maturity Diagram

At any moment of development, the designer may presume an actual position in the diagram according to the current status of the design, but this position may differ from the real and legitimate position of the design; the presumed and legitimate positions may have discrepancies. A discrepancy is caused by a lack of knowledge of the designer because he has missed some essential design artefacts. As a result, the designer rates the level of engineering of the current product design higher than it actually is good for. When he finds the design error that causes the discrepancy, the problem can be addressed. However, if it is not discovered, the discrepancy will present itself at some point in the remaining part of the development process or after market introduction as a surprise to the engineers. The presumed position in the diagram needs to be corrected and that may lead to a project delay. Discrepancies between the presumed and legitimate position in the Axiomatic Maturity Diagram are the result of *unrecognised information* and due to its disruptive character, it may have large impact on the remaining product development process. Therefore, the goal is to discover discrepancies between presumed and legitimate positions as early as possible.

6.3.3 Determination of the Legitimate Position

Finding *unrecognised information* is the key challenge for product designers and there is no method that comprehensively enables this. However, it is possible to apply methodologies that objectively determine the position of a design in the Axiomatic Maturity Diagram. This forces the presumed position to be based on facts instead of gut feeling. It will contribute to a higher degree of realism of the designer. A number of methods that focus on the conceptual design have been described in literature. These methods could be used to apply regularities to the design and in order to reveal *unrecognised information* (some of these methods were introduced in Chapter 2); Tay & Gu apply the hierarchical product topology of the design from the functional and physical domains into a relational data model (Tay & Gu, 2002). Chen et al. expand this method with a production framework (Chen et al., 2012) and the architecture framework for manufacturing system design of Benkamoun et al. also use the axiomatic domains and the hierarchical structure (Benkamoun et al., 2014). Zhang & Chu have developed an approach for the design of product and maintenance by combining AD, QFD and FMEA (Zhang & Chu, 2010). Suh has also reported a sequence of steps to follow that are based on FMEA (Suh, 2004). Finally, the *CμSD Framework* of Chapter 4 may be applied to add regularities using iterative improvement cycles.

As *unrecognised information* only exists in its hidden state, it instantly changes to *recognised information* when it is discovered. In the form of *recognised information*, the designer can address it by completing and decoupling the design matrix. Quantification of *recognised information* may be done with the 'Independence Measure' in Acclaro Design™ as described by Do (Do, 2015).

Axiomatic information is easier to quantify. It does not blur the perception of the designer with discrepancies between perceived and legitimate positions. The common ranges of the system can be quantified with the known statistical methods such as methods for six sigma (Yang & EI-Haik, 2008) and Taguchi (Taguchi et al., 2005). Remaining risks could be quantified by FMEA (Suh, 2004; Puik et al., 2013c) or Qualitative Analysis (Puik et al., 2013b).

6.3.4 Ideal Development Path in Product Design

Product development, as indicated above, will start somewhere at the lower left hand side and will move diagonally upwards. The exact starting point will depend on the complexity of the project definition. A high-tech project that is new to the world might start with high amount of ignorance in the deep lower left corner. A project that aims to develop according to the First-Right-Time philosophy should start without *unrecognised information* and starts further to the lower right-hand side of the diagram. Also, the chosen path may be dependent on the amount of risk that is acceptable to the company, e.g.; the most efficient development path in terms of investment (SME), a path that reduces lead time (semiconductor industry), or a path that minimises development errors (medical or avionics). As explained in Section 6.2.1, it is preferred to start with the Independence Axiom followed by the Information Axiom due to the disruptive character of *unorganised information*, thus:

- Define FRs and find all relevant DPs to address *unrecognised information*;
- Decouple the design matrix to address *recognised information*;
- Match the design ranges and system ranges to guarantee an adequate common range to address *axiomatic information*.

This leads to a preferred path that first moves to the right and then angles upwards. It is plotted in the left-hand graph of Figure 6.2.

Depending on the preferred project strategy, a more or less risky path could be followed. In case of the rather conservative and slow but safe path of the Waterfall-Model, (Royce, 1970) the procedure of following Independence and Information Axioms in that order would be persistent (Figure 6.2 right-hand graph). A slightly more risky path that in practice enhances the development speed of projects is the path of ‘Simultaneous

Engineering' (Bullinger & Warschat, 2012). This gives the designer more room to start early work on robustness, process technology, and other life cycle elements. This merges the work on Independence and Information Axioms and possibly shortens project lead time.

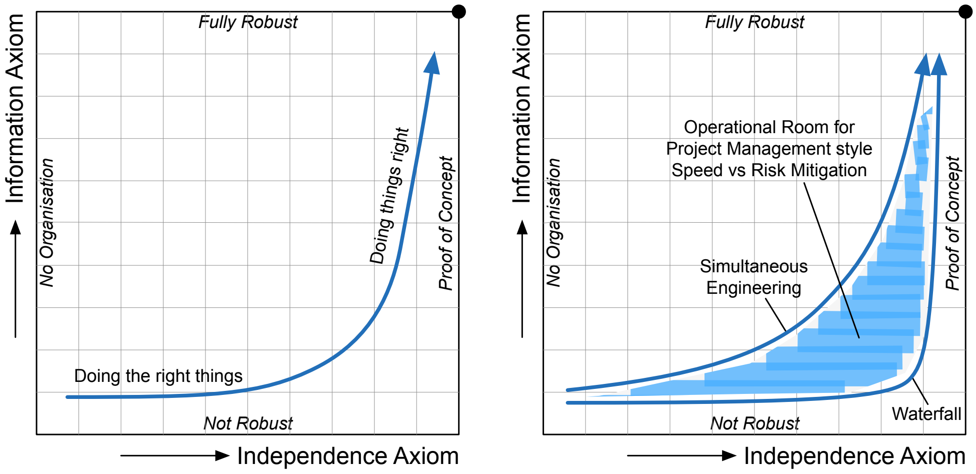


Figure 6.2 *Left; preferred development path through the Axiomatic Maturity Diagram, as indicated in literature, first moves to the right to satisfy Axiom 1. After this, Axiom 2 is satisfied in an upward direction. Right; depending on the nature of the project, a different strategy may be followed. The right lower curve would represent a waterfall management approach, while the upper would represent the path in case of a simultaneous engineering strategy*

6.3.5 Consequences of Typical Errors

Unexpected errors in the development process are mostly related to the discovery of *unrecognised information*. This reveals discrepancy between the presumed and legitimate position. It will divert the development path in the Axiomatic Maturity Diagram. Depending on the kind of error, a discontinuity will appear. This discontinuity is the result of the conversion of *unrecognised information* to *recognised information*. It may show as a kink in the development path or a jump to a different position in the diagram, depending on:

- Availability of a solution to address the problem;
- Robustness of the current design being affected or not.

If a solution to a newly revealed problem is available, it will cause a jump in horizontal direction because *unrecognised information* is converted to *recognised information* and that is addressed right away. If robustness of the design is affected, this means that the design matrix changes and robust DPs are replaced by non-robust DPs. This will cause a drop in vertical direction because *axiomatic information* increases. The following typical design errors could occur:

No decoupling of the design matrices: The first typical problem is the example that was applied in Chapter 5, Subsection 5.3.3 where relevant FRs and DPs are known but the design matrix is coupled. As a result, the designer will have problems setting up the system and it will show inexplicable system behaviour. It is possible to optimise the design conforming the Information Axiom and have adequate common ranges, but *recognised information* remains in the system. An example is the combination lock as described by (Suh, 2004; 2005a): If a combination lock is to be opened without knowing the code, it is a matter of trial-and-error to open it. Even if the instruction manual is available it is not possible to open it without further knowledge (being the code). The designer knows he is missing essential knowledge.

As shown in Figure 6.3, the result depends on whether the DPs need replacement. If this is the case, replacing DPs will lead to a fall back in satisfaction of *axiomatic information* on the vertical axis. If the design matrix is decoupled and the known DPs can be maintained, the effects may be minimal; all information will be eliminated and the mature state is the result.

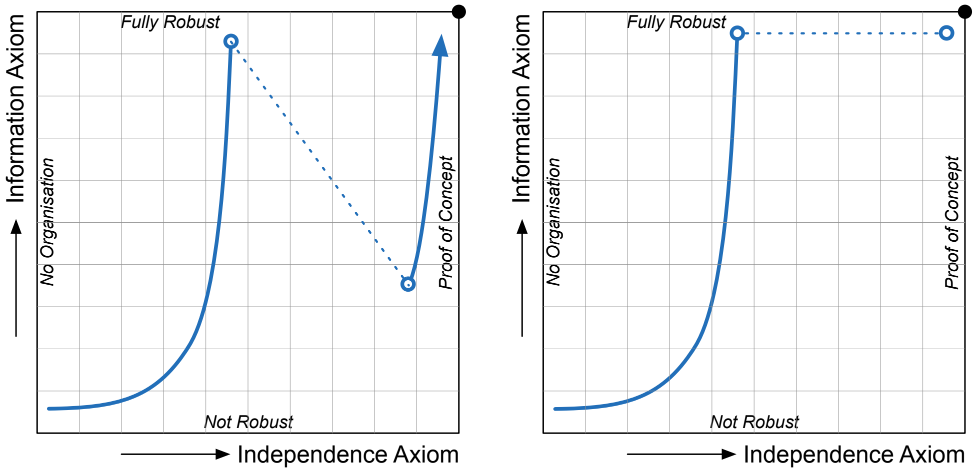


Figure 6.3 *A coupled design matrix does not conflict with satisfaction of the Information Axiom. However, if decoupling of the matrix needs replacement of DPs, the Information Axiom is not automatically satisfied for the new DPs and efforts may be lost (left). The second option shows a luckier situation that the DPs can be maintained. In this case the impact on the design is minimal (right)*

Wrongly chosen DP: Another typical problem is the second example of Subsection 5.3.3. A wrongly chosen DP leads to the situation that the DP does not satisfy the related FR. It will seem to the designer that the design matrix is understood and decoupled, but in fact this is not the case. Time and effort are spent to match the system and design ranges of this DP, but since the DP has no effect, these efforts do not succeed. The designer may deduce that something is wrong but he does not know that the particular DP is causing the problem. As such, this situation leads to *unrecognised information*. To correct the problem, the designer needs to locate the wrong DP. As a result, the design matrix will need corrections and to address the related FR, a new and relevant DP will need to be installed. Figure 6.4 plots the possible discontinuities when a wrong DP in the design matrix is discovered.

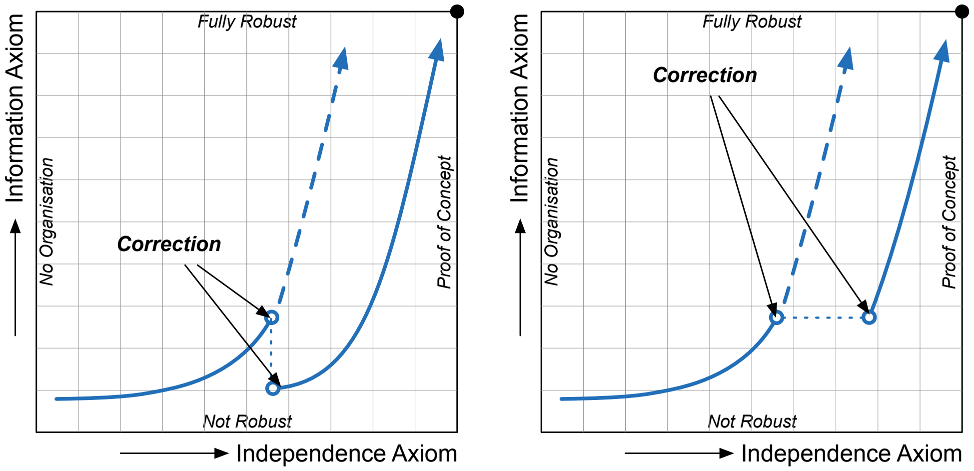


Figure 6.4 *Discovery of a wrong DP leads to a discontinuity in the development process. In the unlucky situation that an obsolete DP was already optimized, efforts are lost and the new DP again needs optimization and a correction takes place (left). In a lucky situation, the problem can be solved with minor efforts. In this case, the related unrecognized as well the recognized information disappears (right)*

Non-matching system and design ranges: A non-matching system and design range for one or more of the design relations between FRs and DPs leads to the situation that the Information Axiom cannot be fully satisfied. Note that the definition of *axiomatic information* is based on joint probability (quantified product of all probabilities, appendix C) or the sum of all information in the design relation. Therefore, the mature state is only reached if all system- and design-ranges are matched (Figure 6.5). In this case, there is no discrepancy between presumed and legitimate positions.

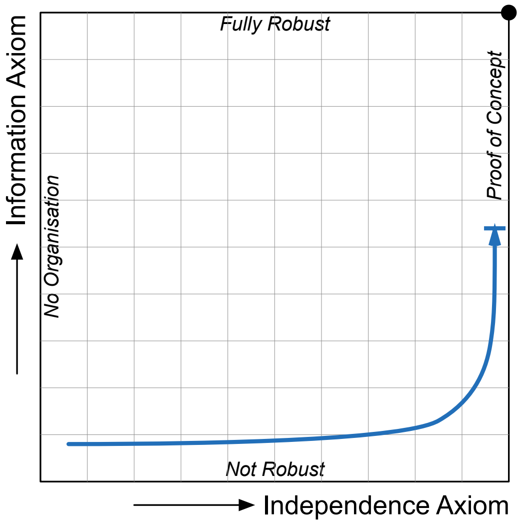


Figure 6.5 *Non-matching system- and design ranges prevent the mature state from being reached. The design will not become robust*

6.3.6 How the Axiomatic Maturity Diagram Addresses the Key Limitations

The Axiomatic Maturity Diagram has the capability to address the key limitations as follows:

- If the designer indeed has not chosen the right structure, the horizontal axis will start rising early. It leads to discrepancies in presumed and legitimate positions in the Axiomatic Maturity Diagram;
- If the designer does not succeed in making the design robust it will not reach the fully robust status marked with the dot Figure 6.5;
- It was shown what errors can occur, how to recover from it, and what the consequence are in terms of work that needs to be redone Figure 6.3, Figure 6.4, Figure 6.5;
- The different development paths for SMEs, semiconductor industry, safety systems or medical industry, follow different trajectories as indicated in Figure 6.2.

6.4 Two Cases that Apply the Axiomatic Maturity Diagram to Explain Design Errors in Retrospect

Two cases that help explaining design errors by application of the Axiomatic Maturity Diagram are included in Appendix E. These cases do not specifically address microsystems, but show how the diagrams may be used in practice. It also demonstrates the generic character of the method.

6.5 Discussion on the Application of Information to Monitor Development Processes

Based on information in design it is possible to trace product development. Three kinds of information each show a typical course that characterises the state of design:

- *Unrecognised information* leads to discontinuities in the development path;
- *Recognised information* prohibits product development to reach proof of concept (in horizontal direction);
- *Axiomatic information* prohibits product development to reach the state of robustness (in vertical direction).

6.5.1 Strengths of this Approach

In learning organisations, universities but as well companies, visualisation of the design process can serve as a tool to explain the origin of errors made in projects to students and novice designers. Causes and consequences become clear lessons for future design projects and it will contribute to the learning experience of the designer (design team). Communication, supported by these visual means, could function as a universal language to widen the scope of personnel, increasingly being capable of understanding what went wrong, for students, engineers, but also managers, and executives.

The innovative contribution in this chapter is largely carried by the concept of *unrecognised information*. The impact of this kind of information is substantial; it can make or break the process of product design due to the discontinuities that can set the design back and might appear as a total surprise to the designer.

6.5.2 Weaknesses of this Approach

The concept of *unrecognised information* also uncovers the largest weakness of this analysis; *unrecognised information*, as the name indicates, is hard to recognise and that is also the reason why it remains hidden. Good understanding of the design e.g. by mathematical, quantitative or qualitative modelling, increases the chance to perceive *unrecognised information*. The reason is that good understanding leads to well-chosen regularities in design and this eliminates information in general. Providing a graphical overview of the product status does not necessarily reveal missing information, but it may help to understand the stages in the development process and how to act accordingly. Faulty scenarios, eventually from the past, can be analysed, characterised, and corrected. This learning experience might help understanding of future projects if similar patterns occur and are indeed recognised by the designer. However, in the execution of design projects it is never completely clear if discrepancies in the Axiomatic Maturity Diagram are latent. Till now, this cannot be guaranteed.

6.5.3 Further Remarks

The order in which the three kinds of information are addressed is preferably the same as in the bullet list above. The safest way is to address *unrecognised information* as soon as possible by functional modelling of the system or testing the (preliminary) design with the *μSD or CμSD Framework*. This changes *unrecognised information* in *recognised* *information* so it can be addressed. Further, *recognised information* and *axiomatic information* will be addressed in that order conforming good practice of AD. The principle of simultaneous engineering proposes a parallel approach of *recognised information* and *axiomatic information* up to some extent. This consciously trades speed of development for development risks. The right path to choose should be an executive decision.

6.6 Conclusion

This chapter presents a way to apply the different kinds of information for a graphical analysis of the product development process. The Axiomatic Maturity Diagram is applied for the visual representation. *Unrecognised information* may present itself with discontinuities in the diagram and causes bends or sudden jumps in the development path. There are two causes for jumps; a fall back in robustness of the design due to the fact that optimised DPs become obsolete, or the recognition of a problem for which a solution is available. In the first situation, a vertical drop is the result and the second case leads to a horizontal progression. If no jump is caused, the information is converted to *recognised information*, which may be addressed by the decoupling of the design matrix. It results in a steady move to the right in the diagram. A preferred path is found by addressing *unrecognised information*, *recognised information*, and *axiomatic information* in that order, analogue to good practice in AD. Finally, the visualised analysis in the Axiomatic Maturity Diagram contributes to the understanding of imperfections during the execution of projects. Because of this it is especially suitable for learning environments. The strengths are not particularly recognised in the prediction of imperfections in projects; this remains a challenge for future investigations.

CHAPTER 7

ASSESSMENT OF RECONFIGURATION SCHEMES FOR RECONFIGURABLE MANUFACTURING SYSTEMS BASED ON RESOURCES AND LEAD TIME*

7.1 Introduction

The current requirement of reducing lead-times for the development of microsystems and their manufacturing systems has changed the way these manufacturing systems are developed. Instead of developing them from scratch, which was the standard up to the 20th Century, modern ways to develop manufacturing systems are to compose them of standardised elements. Not only can these Reconfigurable Manufacturing Systems (RMS) be assembled from modular parts, they can also be modified in a later stadium when it needs adaptions for later generations of the product.

This chapter addresses the reconfiguration process of RMS. As such, it applies the capability of the method for *intelligent gating* to assess what needs to be done to

---

** Parts of this chapter were published in:*

Puik, E. C. N., Telgen, D., Moergestel, L., van, & Ceglarek, D. (2014). Classification of Reconfiguration Resources and Lead Time for Reconfigurable Manufacturing Systems. In F. Chen (Ed.). Presented at the International Conference Flexible Automation and Intelligent Manufacturing, San Antonio, (Puik et al., 2014b).

Puik, E. C. N., Telgen, D., Moergestel, L., van, & Ceglarek, D. (2016). Assessment of Reconfiguration Schemes for Reconfigurable Manufacturing Systems Based on Resources and Lead Time. Robotics and Computer-Integrated Manufacturing (Puik et al., 2017a).

successfully reconfigure the RMS and fulfil the actual need for manufacturing. The assessment is then applied to compare different scenarios for reconfiguration.

Section 7.2 defines the problem statement, inventories the current situation in the field, and evaluated the key limitations. Section 7.3 presents a methodology to assess alternatives for reconfiguration by looking forward and inventory the needed resources and lead time. It calculates and compares necessary resources to execute the options for reconfiguration. Section 7.4 studies a case that compares different *reconfiguration schemes*, applied for the 3D measuring probe of Chapter 3. Finally, Section 7.5 discusses the results and Section 7.6 draws conclusions.

7.2 Analysis & Approach

This section analyses the state-of-the-art of how manufacturing systems are generally assessed in the field. It will be compared to the other options that were mentioned in the introduction of the thesis (DMS, AMS, FMS).

7.2.1 Problem Statement

As RMS are a logical addition to Dedicated Manufacturing Systems (DMS), Adjustable Manufacturing Systems (AMS), and Flexible Manufacturing Systems (FMS) it is desirable to know how these systems relate to each other. Zhang et al. conducted an analytical comparison of DMS, AMS, FMS & RMS (Zhang et al., 2006a), rating given systems on ‘Adaptability’ and ‘Reconfiguration Time’ as shown in Figure 7.1. The overview in Figure 7.2 was obtained by combining the data in a two-dimensional representation. RMS are not adapted as quickly as FMS as the change of structure comes with more overhead than the change of software. RMS score well on adaptability;

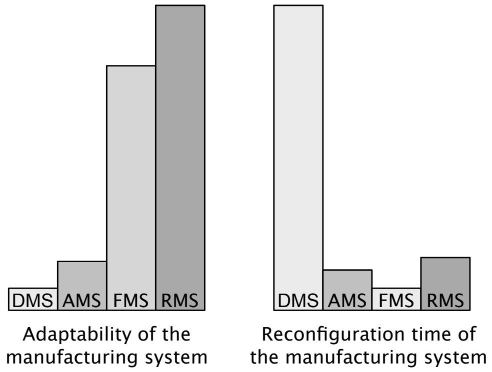


Figure 7.1 *Adaptability of (left) and reconfiguration time of manufacturing systems (right)*

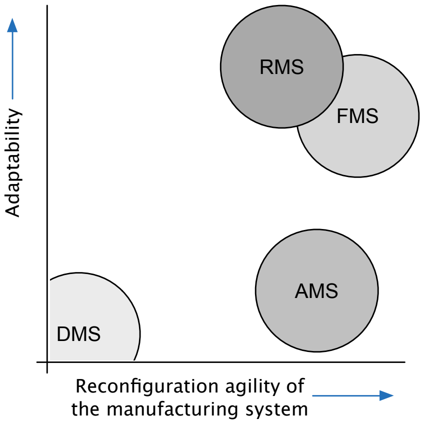


Figure 7.2 *Comparison of DMS, AMS, FMS and RMS. Note that the horizontal axis is reversed compared to Figure 7.1*

changing the structure of the system adds broad possibilities for implementation of optimised process technology. However, the exact position of the RMS in the graph may vary depending on the kind of reconfiguration. The manner in which (re)configuration is performed determines how much renewal is applied in the system.

A direct consequence of this discrimination is the amount of work that is required to carry the reconfiguration through. The lead time and resources needed for implementation will be lower if reconfiguration takes place with known and tested *process modules*. If new *process modules* need to be developed, lead time and reconfiguration efforts are considerably larger, but the added capability of the system may be expected to increase noticeably too. Therefore, the ‘reconfiguration scheme’ of an RMS needs to be carefully investigated ahead to find the adjustment of the system that adequately matches production demand. Schemes refer to the systematic layout of a particular setup and how it is composed from individual (modular) *process modules*. A method for inventorying and quantification of risks in the reconfiguration process could help to find the right balance in optimisation on the short, the mid, or the longer term.

7.2.2 Current Situation

Assessment of manufacturing systems from an economic perspective is a widely-investigated topic that over the years has created profound improvement of manufacturing efficiency. Increasing dynamics in manufacturing have led to increased need for changeability on the shop floor. Assessment, that initially was of a pure economic nature, is in modern management extended with the consideration how that valuation affects potential strategic benefits, e.g. being able to adapt to rapidly changing product demands (Kuzgunkaya & ElMaraghy, 2009). In this context is respectively inventoried: (i) adaptability assessments that compare a range of systems (DMS, AMS, FMS, or RMS), (ii) the assessment of the adaptability of FMS, and (iii) assessment of adaptability of RMS.

(i) Methods for Comparison of DMS, AMS, FMS, and RMS

Since radical changes in the structure of DMS and AMS are not intended nor foreseen, methods for assessment of change of DMS and AMS are minimal in literature. A survey by Hollstein et al focusses on possibilities rather than limitations by inventorying how far existing (dedicated) manufacturing systems can be upgraded with economically feasible interventions (Hollstein et al., 2012). Michaelis & Johannesson choose a comparable approach but add the principle of co-design to fit the development of new products to the opportunities and limitations of existing equipment (Michaelis & Johannesson, 2012). Zhang & Chu compare the economic performance of the systems by focussing on the time needed to carry through changes (Zhang & Chu, 2010). Kuzgunkaya & ElMaraghy go further and compare FMS and RMS based on a combination of key parameters e.g. economic considerations (Kuzgunkaya & ElMaraghy, 2009), structural complexity, and responsiveness. Amico et al extend comparison on economic considerations with the theory of ‘real options’ to define a payoff function that can be used to compare different systems (Amico et al., 2006). Nassehi et al apply ‘formal methods’, mathematical techniques for the specification, design and verification, to check the consistency of manufacturing processes (Nassehi et al., 2012).

(ii) Assessment of the Adaptability of FMS

Though the adaptability of FMS is of a different nature than that of RMS, related work investigates the change of a number of FMS on the shop floor, which is a similar but higher-level approach compared to the reconfiguration of RMS. Abdel-Malek & Wolf focus on the most efficient mapping of FMS in a factory. In their approach, FMS can be moved, changed, or upgraded similarly like *process modules* in a RMS (Abdel-Malek & Wolf, 1991; 1994). This is benchmarked using Key Performance Indicators (KPI) to compare different alternatives and select the best option. Lotfi presented a linear programming model for the optimisation of a number of objectives i.a., financial aspects, flexibility, and group homogeneity (Lotfi, 1995). Yan et al. use a what they call ‘Life Locus Tree’ that is not only modelling the birth of the FMS but also focusses on use, adaption, and expansion (Yan et al., 2000).

(iii) Assessment of the Adaptability of RMS

With the start of the new millennium, assessment of the adaptability of RMS has gained attention. Spicer developed a method for economic evaluation of RMS and presents design principles for RMS to optimise the number of *process modules* in a system (Spicer et al., 2007). Abdi & Labib introduce a method called Analytic Hierarchy Process, a tactical tool to decompose the match between a manufacturing system and a family of products to a hierarchical order (Abdi & Labib, 2003; 2007). The model has the ability to support product family justification in a wide variety of RMS. Farid &

McFarlane assess reconfigurability of distributed manufacturing systems with a ‘Design Structure Matrix’ to define reconfiguration ease (Farid & McFarlane, 2006; 2008; Farid, 2014). The matrix is inspired by Axiomatic Design and defines the number of ‘Degrees of Freedom’ of the system. A higher number of DoF increases the number of transformation processes and increases flexibility of the system. Hasan et al. present a mathematical reconfigurability model based on the ‘Multi Attribute Utility Theory’ (Hasan et al., 2013). This model is capable of computing the effort for converting an existing machine configuration to a new configuration. The model discriminates between modules added, removed, or changed. Weighting factors are applied to vary the impact of change, but no further dependency to the kind of change is applied. The method as proposed by Hasan et al. is the only method for assessment of RMS that, though in a basic way, takes the difficulty of the reconfiguration process into account.

7.2.3 Key Limitations

The *reconfiguration scheme* has significant influence in needed resources and lead time of the reconfiguration process. Mainly new developments make heavy demands on resources and lead time. Therefore:

- *Reconfiguration schemes* of RMS need to be carefully investigated ahead to find the adjustment of the system that adequately matches production demand;
- A method for inventorying and quantification of risks in the reconfiguration process could help to find the right balance in optimisation on the short, the mid, or the longer term;
- Such methods, to assess the *reconfiguration schemes* of RMS, do not exist yet.

7.3 Methodology to Apply the Framework of Functional Gating as a Measure for Comparison of Reconfiguration Schemes

The method for assessment of *reconfiguration schemes* as presented in this chapter is based on the method for *concurrent gating* of Chapter 4. Figure 7.3 shows the seven steps as defined there.

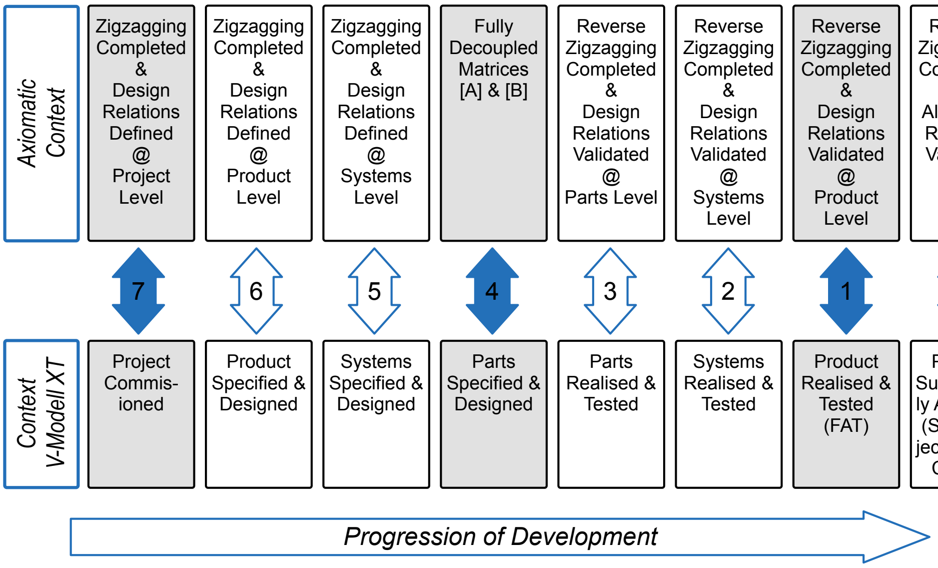


Figure 7.3 *Development of RMS in seven steps from the embryonic stage to an independent and robust design*

New equipment schemes can in some cases be assembled without development of new *process modules*. However, in the majority of situations this is not possible because modules that are well specified need to be applied outside their specified operating window or product-specific modules need to be developed. The development needed to upgrade the *process modules* causes new risks in the configuration process of the RMS. The index method will be applied to discriminate between these situations.

The method for *concurrent gating* of Chapter 4 divides the development process into seven stages by defining relatively accurate nuances between the stages based on decomposition. For the classification of reconfiguration resources, a number of seven stages seems not necessary; the overhead of determining accurate value judgement for a reasonably large number of modules would make the method laborious. Therefore, the number of seven stages is reduced to a limited number of three stages. The applied stages are (indicated blue/grey in Figure 7.3):

(a) 'Repeat': *Process modules* have been applied in the past and are well documented, processes are used within their specified operating windows;

(b) 'Adapt': *Process modules* have been applied in the past and are well documented, processes are used outside defined operation windows or in an alternative manner that has not been tested;

(c) 'Expand': *Process modules* do not yet exist and have to be developed (the library of modules is expanded).

(a) Reconfiguration that 'Repeats' application of known and tested *process modules*, which might be considered as the most moderate change, will cause the index level to drop to '*Reverse zigzagging* completed, design relations validated at Product level' (Figure 7.3). Successful testing after integration will bring the RMS up to the final level again. It does not matter if a single or a larger number of modules are replaced; system level tests are always required to bring the system back to production level;

(b) The classification of the reconfiguration type 'Adapt' needs examination of the definition. It states 'Fully decoupled matrices [A] and [B]' which means that all prior testing becomes obsolete and verification and validation should be executed again in the new situation;

(c) Reconfiguration type 'Expand' per definition needs to go through all development levels of the index method as the index method was designed for new module development. Module development is starting completely on the left-hand side in Figure 7.3.

Since the three stages repeat, adapt & expand have been derived from the more nuanced method for *concurrent gating*, classification is expected to be quick and clean. Therefore, the most appropriate stage for process module development can be defined without confusion or discussion.

7.3.1 Calculation of Resources & Lead Time for Initial Configuration

The simplified method with three straightforward stages can be applied for assessment of the *reconfiguration schemes* of modular manufacturing equipment. There will be typically more than a single good scheme to configure a manufacturing system; many alternative configuration schemes exist that all have particular advantages and drawbacks with respect to e.g. reliability, complexity, configuration time, reconfiguration time, and cost. By calculation of the resources and lead time of the respective configuration schemes, comparison between the alternative solutions can be made. In this way, pros and cons can be objectively considered.

Total resources needed for (re)configuration of a particular manufacturing system can be calculated by summation of the total efforts to integrate all *process modules* of that manufacturing setup. The number of steps to reach a robust and independent design as indicated in Figure 7.3 is a measure for the resources needed per module. Depending if modules are reused, adapted, or expanded; this would be a number of respectively 1, 4, or 7 steps. However, required resources are not always proportional to the number of steps. Therefore, weighting factors are applied to correct for eventual discrepancies. The result is a score for the total amount of efforts needed to (re)configure a manufacturing setup, expressed as 'Reconfiguration Resource Units' (RRU), according to

$$RRU_{(Re)conf} = \sum_{i=FirstModule}^{LastModule} \mathrm{S}_i \cdot \mathrm{W}_{RR\,i} \tag{7.1}$$

that represents the resources needed to complete reconfiguration of a manufacturing system. In this equation, S is the number of index steps for each module and W_{RR} is the weighting factor to correct for discrepancies between the number of index steps and the true reconfiguration resources. W_{RR} can be any non-negative real ($\mathbb{R}^{+} \cup \{0\}$). The weighting factors can be optimised based on experience.

Comparably, the minimum lead time (mLTU), or the best possible time to complete the reconfiguration, may be determined by the longest development trajectory of any of the new *process modules*. Note that the mLTU is dimensionless and therefore expressed in 'minimum Lead Time Units'. By multiplication of this number and the average lead time per index step, the true minimum lead times for reconfiguration can be calculated. Again, weighting factors are applied for estimation of the lead time per module depending on the choice for repeat, adapt, or expand. The dimensionless minimum lead time for (re)configuration of the total system is found by taking the upper bound of all individual dimensionless lead times according to

$$mLTU_{(Re)conf} = \max\{\mathrm{S}_{FirstModule} \cdot \mathrm{W}_{LT\,FirstModule}, \ldots, \mathrm{S}_{LastModule} \cdot \mathrm{W}_{LT\,LastModule}\} \tag{7.2}$$

and it should be noted that the weighting factors may be chosen in the same manner as the weighting factors for calculation of resources, as more drastic changes require more time. However, the weighting factors W_{RR} and W_{LT} for reconfiguration resources and lead time can be optimised separately if needed to increase the accuracy of the calculation. This is typically done when experience with the method grows.

7.3.2 Calculation of Resources & Lead Time if Reconfiguration Is Planned Ahead

If reconfiguration takes place on a regular basis, and a number of future readjustments can be foreseen, the calculation of resources can be applied further ahead in time. Engineers may decide to choose a more flexible configuration scheme for the initial setup since later reconfigurations may benefit when they need to be converted for newer products. The joint efforts for reconfiguration can be reduced on the long run by well-timed investments in equipment flexibility. The total of all reconfiguration resources can be calculated by summation of the efforts required for the initial system and the efforts required for future reconfigurations. The method to determine the efforts for a single scheme, by summation of weighting factors as given in Equation (7.1), is expanded by a second summation of the future reconfigurations according to

$$RRU_{MultipleReconf} = \sum_{j=FirstConfiguration}^{LastConfiguration} RRU_{(Re)conf\ j}$$

$$RRU_{MultipleReconf} = \sum_{j=FirstConfiguration}^{LastConfiguration} \sum_{i=FirstModule}^{Lastmodule} \mathrm{S}_{i,j} \cdot \mathrm{W}_{RR\ i,j} \tag{7.3}$$

where the index i counts the number of *process modules* and index j counts the number of (re)configurations including the initial configuration. The best possible dimensionless lead time for all reconfiguration actions together could be calculated comparably by summation of the upper bounds of the respective reconfigurations

$$mLTU_{MultipleReconf} = \sum_{\mathrm{j}=FirstConfig}^{LastConfig} \max\left\{\mathrm{S}_{FirstMod,j} \cdot \mathrm{W}_{LT\ FirstMod,j}, \ldots, \mathrm{S}_{LastMod,j} \cdot \mathrm{W}_{LT\ LastMod,j}\right\} \tag{7.4}$$

which provides the minimal dimensionless lead time spent to all (re)configuration actions. Equations (7.3) & (7.4) enable comparison of alternative choices for equipment configuration and a number of successive reconfigurations. It provides information how a choice for a certain scheme escalates to future systems and so affects the reconfiguration resources and lead time compared to alternative schemes. The gain for an alternative scheme can be calculated by subtracting the reconfiguration resources of the alternative scheme from the basic scheme according to

$$RRU_{Gain} = \left(\sum_{j=FirstConfig}^{LastConfig} \sum_{i=FirstModule}^{Lastmodule} S_{i,j} \cdot W_{RR\,i,j} \right)_{Basic\ Scheme} - \left(\sum_{j=FirstConfig}^{LastConfig} \sum_{i=FirstModule}^{Lastmodule} S_{i,j} \cdot W_{RR\,i,j} \right)_{Alternative} \tag{7.5}$$

where the first term represents the (re)configuration resources of a basic scheme and its derivative reconfigurations and the second term represents the resources for an alternative scenario. The gain in best lead time can be determined in the same manner

$$mLTU_{Gain} = \left(\sum_{j=FirstConfig}^{LastConfig} \max \left\{ S_{FirstMod,j} \cdot W_{LT\ FirstMod,j}, .., S_{LastMod,j} \cdot W_{LT\ LastMod,j} \right\} \right)_{Basic\ Scheme} - \left(\sum_{j=FirstConfig}^{LastConfig} \max \left\{ S_{FirstMod,j} \cdot W_{LT\ FirstMod,j}, .., S_{LastMod,j} \cdot W_{LT\ LastMod,j} \right\} \right)_{Alternative} \tag{7.6}$$

in which a positive gain in dimensionless lead time is in favour of the alternative scheme.

7.3.3 How the Framework of Functional Gating Addresses the Key Limitations

The novelty of the presented solution here is that it not only compares different strategies for possible manufacturing solutions, being able to compare their economic strengths, but also supports substantiated development of the RMS by inventorying the need for new *process modules* to be developed. This brings the complexity of the reconfiguration process versus the envisioned expansion of the RMS into the equation.

7.4 Case: Calculation Example for Reconfiguration of an RMS for 3D Measuring Probes

The methods for the calculation of reconfiguration resources and lead time were applied on a true case; the configuration of a manufacturing machine for the assembly of 3D measuring probes. 3D Measuring probes are made for geometrical measurement of high tech products with high accuracies and small tolerances.

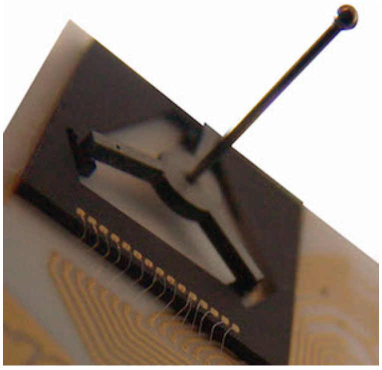


Figure 7.4 *3D Probe for geometrical measurements in the nanometre range*

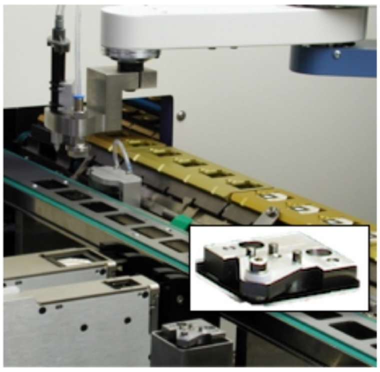


Figure 7.5 *Conventional (passive) alignment setup*

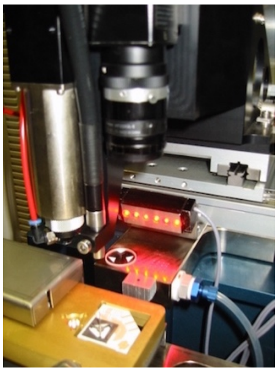


Figure 7.6 *Flexible configuration with computer vision*

The probes are applied in geometric coordinate measuring machines to enable measurements in the nanometre range (Haitjema & van Seggelen, 2003). The heart of the probe consists of a surface-micro-machined silicon ‘Die’ (semiconductor-chip) that contains sensitive strain gauges to convert mechanical nanometre movements to electrical signals. A stylus is applied to measure the object by gently touching the surface and subsequently transfer the sensing movement to the Die. The probe is shown in Figure 7.4.

The intention of the project was to invest in a production system with a family of products in mind. However, when the initial system was configured, it was not yet clear how many siblings the family was going to have. Keeping all options open for maximum flexibility appeared a costly option. However, limiting the flexibility of the equipment, due to rigid design choices, decimates the advantages of the reconfigurable framework of the production system. By application of assessment of application schemes, a number of alternative configuration schemes were developed being able to compare resources and lead time.

The production engineers had a number of technically sound options to configure the production system. These options were brought down to two different strategies; the application of 'passive' alignment or 'active' alignment. Passive alignment is the traditional variant where the tool that determines the position of the parts is a passive tool (mating surfaces, or alignment pins as shown in the inset of Figure 7.5). Active alignment applies active systems for determination of the part's position. Mostly used is a computer vision system that applies a camera and pattern recognition system for finding the position of the parts. This position is communicated with the robotic manipulator for placement (Figure 7.5 and Figure 7.6).

7.4.1 Configuration of the Initial Manufacturing system

Alternative configuration schemes of the manufacturing system are obtained by combining different *process modules*. The options were narrowed down to a single active and passive scheme. Passive alignment would be the most obvious way for the company; it could be realised by straightforward mechanical components. The active alignment system is more comprehensive; it applies a computer vision system, consisting of camera with suitable optical system for lens and illumination of the object, computer with appropriate software and a number of fixtures to bring the parts in place. The vision system had not been applied before and the physical properties of the optical system made the vision system initially more complex. In the perspective of the 'simplified classification' method of this chapter (Subsection 7.3):

- Passive alignment can be applied by adapting an existing process module;
- Active alignment with the vision system can be applied if the library of *process modules* is expanded.

These two solutions were compared by calculating the values for RRU and mLTU conform Equations (7.1) & (7.2). The values for S are determined by the number of steps that are bridged in the axiomatic model of Figure 7.3, the result is shown in Table 7.1.

Table 7.1 *Weighting factors to calculate reconfiguration resources and lead time*

Applied parameters for S & W					
For Calculation of Reconfiguration Resources			For Calculation of Lead Time		
	S	WRR		S	WLT
Repeat	1	1	Repeat	1	1
Adapt	4	1	Adapt	4	1
Expand	7	1	Expand	7	1

As this is the first case to be addressed by this method for comparison, there is no sensible experience yet by application of weighting factors W_{RR} and W_{LT}. Note that the intrinsic distribution of the method, by counting the number of steps to completion, already introduces a basic principle of weighting even if no weighting factors are applied. Therefore, active weighting was at this stage not yet applied: weighting factors for optimisation of reconfiguration resources and the best lead time were all set to 1.

A tailored pick & place system to manufacture the probe was configured from standard *process modules*. Table 7.2 shows the configuration scheme in case of passive alignment.

Table 7.2 *Configuration scheme for passive alignment*

Scheme 1 Initial setup for passive alignment						
Process Module	Class	S	W_{RR}	W_{LT}	$S \cdot W_{RR}$	$S \cdot W_{LT}$
Base Frame	Repeat	1	1	1	1	1
Manipulator	Repeat	1	1	1	1	1
Gripper	Adapt	4	1	1	4	4
Feeder A	Repeat	1	1	1	1	1
Feeder B	Repeat	1	1	1	1	1
Nest for Carrier	Adapt	4	1	1	4	4
Alignment Body	Adapt	4	1	1	4	4
Reconfiguration resources according to Eqn (1)					16	
Minimum dimensionless lead time according Eqn (2)						4

Table 7.3 *Configuration scheme for active alignment*

Scheme 2 Initial setup for active alignment						
Process Module	Class	S	W_{RR}	W_{LT}	$S \cdot W_{RR}$	$S \cdot W_{LT}$
Base Frame	Repeat	1	1	1	1	1
Manipulator	Repeat	1	1	1	1	1
Gripper	Adapt	4	1	1	4	4
Feeder A	Repeat	1	1	1	1	1
Feeder B	Repeat	1	1	1	1	1
Nest for Carrier	Adapt	4	1	1	4	4
Alignment Body	Expand	7	1	1	7	7
Reconfiguration resources according to Eqn (1)					19	
Minimum dimensionless lead time according Eqn (2)						7

The same exercise was applied for the configuration scheme of an active alignment system; this is shown in Table 7.3. The summation in Table 7.2 and Table 7.3 is executed according to Equation (7.1) for the configuration resources and according to Equation (7.2) for calculation of the minimum dimensionless lead time. In both situations, a total number of 7 *process modules* are applied. It is the last module that is different for both configuration schemes; instead of repeating a known module, the vision system is developed as a new module and the library of *process modules* is expanded. It can be seen that both the initial reconfiguration resources and the lead time for configuration are higher for the second configuration scheme. Based on this sole configuration only, the first configuration scheme would be the better one. Resources are saved and lead time is shorter (16/4 compared to 19/7).

7.4.2 If Reconfiguration Is Planned Ahead

The calculation in the last subsection does not evaluate further than the initial configuration of the system. If this is the case, an alternative configuration scheme that invests in flexibility will be less interesting due to the overhead that comes with it. However, this will change if reconfiguration is planned further ahead. Table 7.4 shows the same comparison if a first and a second reconfiguration indeed are carried through.

The two respective reconfigurations have the same character, which is usually the case if parts are modified but the structure of the product is not changed. Again, the solution with passive alignment is compared to the active alignment system with the vision camera. The calculations in the table are the result of Equations (7.1 – 7.4). The outcome shows that after the first reconfiguration the two options score exactly the same. Reconfiguration resources and lead time for both schemes are equal (26/8). However, after the second reconfiguration, scheme 2 appears more efficient. This scheme thrives better in a large number of reconfigurations. Also, all further reconfigurations within the same family of products will be in favour of scheme two. This can be seen in Table 7.5; the gain in reconfiguration resources is calculated by Equations (7.5) & (7.6). Due to the fact that in this example the weighting factors for RRU and mLTU have been chosen the same, the RRU_{gain} and $mLTU_{gain}$ show the same values.

***Table 7.4** Comparison of scheme 1&2 including first and second reconfiguration.*

Reconfiguration Scheme for Passive Alignment													
Initial Setup				First Reconfiguration					Second Reconfiguration				
Process Module	Class	$S{\cdot}W_{RR}$	$S{\cdot}W_{LT}$	Class	$S{\cdot}W_{RR}$	$S{\cdot}W_{LT}$	Σ_{RRU}	Σ_{mLTU}	Class	$S{\cdot}W_{RR}$	$S{\cdot}W_{LT}$	Σ_{RRU}	Σ_{mLTU}
Base Frame	Repeat	1	1	Repeat	1	1	RRU & mLTU after 1st re-configuration according to Eqs (3 & 4)		Repeat	1	1	RRU & mLTU after 2nd re-configuration according to Eqs (3 & 4)	
Manipulator	Repeat	1	1	Repeat	1	1			Repeat	1	1		
Gripper	Adapt	4	4	Repeat	1	1			Repeat	1	1		
Feeder A	Repeat	1	1	Repeat	1	1			Repeat	1	1		
Feeder B	Repeat	1	1	Repeat	1	1			Repeat	1	1		
Nest for Carrier	Adapt	4	4	Repeat	1	1			Repeat	1	1		
Alignment Body	Adapt	4	4	Adapt	4	4			Adapt	4	4		
RRU according to Eqn (1)		16		RRU Eqn (1)	10		26		RRU Eqn (1)	10		36	
mLTU according to Eqn (2)			4	mLTU Eqn (2)		4		8	mLTU Eqn (2)		4		12

Reconfiguration Scheme for Active Alignment													
Initial Setup				First Reconfiguration					Second Reconfiguration				
Process Module	Class	$S{\cdot}W_{RR}$	$S{\cdot}W_{LT}$	Class	$S{\cdot}W_{RR}$	$S{\cdot}W_{LT}$	Σ_{RRU}	Σ_{mLTU}	Class	$S{\cdot}W_{RR}$	$S{\cdot}W_{LT}$	Σ_{RRU}	Σ_{mLTU}
Base Frame	Repeat	1	1	Repeat	1	1	RRU & mLTU after 1st re-configuration according to Eqs (3 & 4)		Repeat	1	1	RRU & mLTU after 2nd re-configuration according to Eqs (3 & 4)	
Manipulator	Repeat	1	1	Repeat	1	1			Repeat	1	1		
Gripper	Adapt	4	4	Repeat	1	1			Repeat	1	1		
Feeder A	Repeat	1	1	Repeat	1	1			Repeat	1	1		
Feeder B	Repeat	1	1	Repeat	1	1			Repeat	1	1		
Nest for Carrier	Adapt	4	4	Repeat	1	1			Repeat	1	1		
Alignment Body	Expand	7	7	Repeat	1	1			Repeat	1	1		
RRU according to Eqn (1)		19		RRU Eqn (1)	7		26		RRU Eqn (1)	7		33	
mLTU according to Eqn (2)			7	mLTU Eqn (2)		1		8	mLTU Eqn (2)		1		9

Table 7.5 *Gain in reconfiguration resources and lead time. Note that reconfiguration '0' is the initial configuration of the production system*

Reconfiguration No							
	0	1	2	3	4	5	6
RRU_{Gain}	-3	0	3	6	9	12	15
$mLTU_{Gain}$	-3	0	3	6	9	12	15

7.5 Discussion

The case shows that assessment of *reconfiguration schemes* can be applied to compare alternative setups of reconfigurable equipment. The method also shows that it can be applied in an early stage of development.

7.5.1 Strengths of the Application of Reconfiguration Schemes

The framework enables reasonably objective determination of the development status of *process modules*; because only three options are applied, reuse, adapt, and expand, and there is substantial difference between these options, there is little discussion how modules should be classified. Therefore, a quick and effective resource estimation of the impact of the redesign can be made. The integrity of the Axiomatic Design matrix acts as a determinative parameter for the impact when redesigning *process modules*. The design matrix can be explored with relatively little effort. It enables the method to be applied in dynamic environments, for internal or external quotations and for determination of response-times to market demand. Therefore, the method seems reliable to assess resources and lead time when reconfiguring RMS. It also seems possible to apply the method as a systems engineering tool for effective comparison of alternative *reconfiguration schemes*.

7.5.2 Weaknesses of the Reconfiguration Schemes

The strength of the method for assessment of RMS also comes with a downside. Especially for new *process modules*, it can be difficult to estimate the required amount of work for development of those modules. When there are several *process modules* to expand or adapt, the accuracy of the method may be compromised.

When comparing schemes for the initial configuration, according to Equations (7.1 & 7.2), the result tends to be in favour of the conventional and less flexible solution. The reason is that implementation of flexibility comes at a cost; direct comparison will always be in favour of a system optimised for a single product due to the fact that it has no flexibility requirements to address. The second set of equations, assuming that a number of reconfigurations will be carried through, as calculated by Equations (7.3 & 7.4), may be too tolerant due to the uncertainty that these future reconfigurations indeed take place while the method just assumes that this is the case.

Minimum dimensionless lead time, as calculated by Equation (7.4), compares the total (dimensionless) time that is spent to multiple reconfigurations when equipment is converted a number of times. Comparably, Equation (7.6) determines the gain in lead time over these conversions. These parameters can be applied to estimate the total loss of turnover due to the execution of reconfigurations, since systems under reconfiguration are not producing. However, the outcomes of Equations (7.4 & 7.6) are under restriction that unlimited resources can be applied to the reconfiguration process. Mainly for large reconfiguration projects, the number of workers involved is against the limit and resources have to be applied sequentially. The equations may not deliver the desired result in these cases.

7.5.3 Limitations of the Reconfiguration Schemes

The number of gradations when modifying *process modules*, repeat, adapt, or expand as here proposed, is limited to a number of three. The thought behind it is that quick comparison does benefit from a simple framework. The simplification leads to straightforward application of the method which enhances agility; a deliberate choice was made to let agility prevail over accuracy. Experienced users may decide for a different balance and expand the assessment method to use all seven stages of the gating framework.

7.5.4 Other Considerations

Despite the fact that accuracy cannot be guaranteed under all circumstances, the method can be used to compare configuration schemes in the early stage of projects since other methods, insofar as they exist, have their shortcomings too.

There are statistical effects that may improve the accuracy of the method. At first, when applied to configuration schemes with a large number of *process modules*, accuracy is expected to improve due to the averaging effect. Secondly, when applied to situations where the number of repeated modules is high, the accuracy is also expected to improve due to the larger certainty of estimation of *process modules* that are reused versus modules that need sweeping changes.

During application, many (sub)systems as applied in comparable configuration schemes will be similar. These systems may be either, included in, or excluded from the resource calculations as needed. In any case, all inaccuracies of calculations that concern identical systems in different configuration schemes will not influence the accuracy of comparison of different alternatives. This explains that the power of the tool comes more to its own in comparison of resources, for competing *reconfiguration schemes*, than in absolute calculation of required resources. The dimensionless character of RRU and mLTU is in these cases no longer object of consideration.

The nature of the reconfiguration process influences the character of the manufacturing system. When reconfiguration just concerns actions of the type 'repeat', it is doubtful if the structure of the equipment will change. According to the definition (Koren et al., 1999), the RMS is then used as an FMS and only delivers 'Generalised Flexibility' (ElMaraghy, 2006). If the system is reconfigured with a number of *process modules* that are expanded, the system is used as a true RMS and delivers 'Customised Flexibility'. The former is considerably less adaptable than the latter but offers better agility due to the less complicated development efforts needed. This effect is shown in Figure 7.7. This observation confirms the findings of this chapter; increased adaptability comes at the price of higher resources and increased lead time.

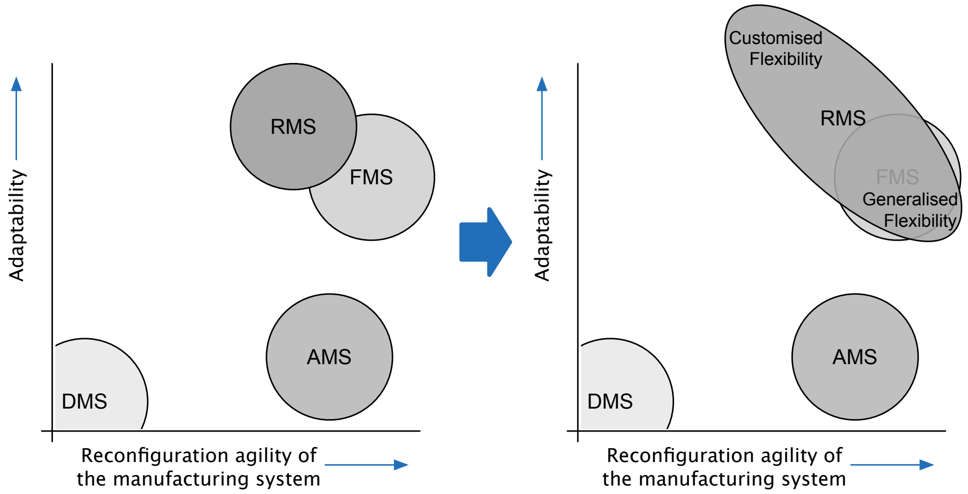


Figure 7.7 *The character of the reconfiguration process determines the difference between 'Generalised Flexibility' and 'Customised Flexibility'*

7.5.5 Opportunities for Further Improvement

Weighting factors that were implemented for the case with the measurement probe were all set to 1 (Table 7.1). The reason was that the intrinsic distribution of the index method, by counting the number of steps to completion (Figure 7.3), already introduced a basic principle of weighting even without the weighting factors set to a specific value. However, it is unlikely that the actual work to execute the reconfiguration matches the 1, 4, or 7 steps of the index method. It may be expected that the accuracy of the procedure can be improved as the understanding of the process increases. The weighting factors provide experienced users the ability to refine the process according to their findings.

In many situations, the certainty that future reconfiguration will take place is low; it cannot be guaranteed if and when it will happen. The current equations cannot be applied for these situations because the certainty that a future reconfiguration takes place is not yet implemented in the model. If the future reconfiguration is incorporated in the comparison, and does not take place in reality, an investment in flexibility may not be earned back. Otherwise, if it is not incorporated but it does take place, a better scheme could have been chosen at an earlier stage. To overcome this problem, the probability that a reconfiguration actually takes place could be brought into the model.

7.6 Conclusions

The assessment method for inventorying the reconfiguration of RMS can be successfully applied to compare different solutions when reconfiguration is planned ahead. The effective ways to quantify the impact on resources enables quick comparison for engineers and management. The method is particularly suitable to get a quick indication of the needed resources, albeit rough, but presented in the conceptual phase of development of product and manufacturing means. The method for assessment is based on the structure of *concurrent gating*, which gives a solid scientific substantiation. For convenience, a simple and practical framework is built around the basis of Axiomatic Design, that distinguishes three levels of reconfiguration for the modular parts of the RMS; repeat, adapt, and expand. It simplifies application considerably, but it also lowers the accuracy of the method.

CHAPTER 8

PRODUCT DESIGN IN THREE PHASES; A CONSTITUENT ROADMAP THAT MONITORS THE STATUS OF THE DESIGN PROCESS BY TRACKING KNOWLEDGE OF THE DESIGNER*

8.1 Introduction

This chapter combines the best characteristics of the models that were presented in this thesis. In one general model, it enables *intelligent gating*, and *iterations in the design process* to respectively support project control, and agility. This model is called 'Constituent Roadmap' because it connects to existing models that may be applied within its context. The *constituent roadmap* is based on information in design and AD plays a central role.

This chapter is organised as follows; Section 8.2 analyses the problem to be addressed. Section 8.3 proposes the *constituent roadmap*. In Section 8.4, the *constituent*

---

[*] *Parts of this chapter were published in:*

Gielen, P., Sillen, R., & Puik, E. C. N. (2012). Low cost environmentally friendly ultrasonic embossed electronic circuit board (pp. 1–7). Presented at the 2012 4th Electronic System-Integration Technology Conference, IEEE, (Gielen et al., 2012).

Puik, E. C. N., & Ceglarek, D. (2015). Axiomatic Product Design in Three Stages; A Constituent Roadmap that Visualises the Status of the Design Process by Tracking the Knowledge of the Designer, Presented at the ASME IMECE2015, Houston, (Puik & Ceglarek, 2015a).

Puik, E. C. N., & Ceglarek, D. (2015). The Quality of a Design Will Not Exceed the Knowledge of Its Designer; an Analysis Based on Axiomatic Information and the Cynefin Framework. Procedia CIRP, 34, 19–24, (Puik & Ceglarek, 2015b).

roadmap is demonstrated by the development of a ‘Micro Hydrogen Sensor’. Section 8.5 discusses the findings, and Section 8.6 draws conclusions.

8.2 Analysis and Approach

Monitoring project progression, during project execution, is valuable for project managers and engineers. This section inventories the importance to track project progression based on actual knowledge of the design.

8.2.1 Problem

In the previous chapters were discussed: (i) *functional* and *concurrent gating* based on the level of decomposition, in other words, measuring ‘implemented knowledge to the design’, and (ii) *intelligent gating* that measures knowledge ‘still to be acquired and implemented to complete the project’. The latter has advantages, ‘what is still to come in a project’ can count on more interest than ‘what has been done so far in a project’. Obviously, it is also the more difficult option; it is more difficult to predict the future than look back at what was achieved so far.

Intelligent gating requires understanding of what is still to come in the project. This understanding should be based on knowledge of the current design and what is needed to complete the design. Knowledge is the central theme that enables *intelligent gating*.

When it comes to knowledge, the acquisition of knowledge prioritises over the implementation of knowledge for two reasons: (i) knowledge should be acquired before it can be implemented, and (ii) knowledge acquisition is a more difficult process than knowledge implementation. The first statement does not need further explanation, however the second is explained here. Knowledge implementation can be executed by spending time and efforts to structure the design; a potentially laborious process that comes with little uncertainties and can be accurately planned ahead. However, knowledge acquisition suffers from similar peculiarities as finding *unrecognised information*. Like *unrecognised information*, the right knowledge reveals itself in an unpredictable manner. As such, planning of knowledge acquisition comes with substantially more uncertainties, causes more risks in project execution, and is considered a more difficult process.

Knowledge is more than being familiar with a list of FRs and the related decomposition tree, stronger, these are just ‘things worth knowing’ and they hardly involve having knowledge e.g.; if the background why this is the case is missing, it misses its added value. It is therefore important that knowledge is anchored in the project and that the designer, who owns the knowledge, is present to ensure its proper use.

8.2.2 Current Situation

The Waterfall- and V-Models do not really anchor knowledge, even gated decisions may need reconsideration if required by residual project risks. In Subsection 2.4.1 an inventory was made of methods that have the capability of modelling the design. Of these methods, the investigations of Benkamoun, and Komoto & TomiYama are particularly valuable for this research since they add the capability of actively securing knowledge in the model itself or in the direct periphery of the model. Benkamoun’s ‘Architecture Framework’ (Benkamoun et al., 2014) increases the relations between the system representation and a rational and systematic design process. As such it could be applied for progress determination of the project. Unfortunately, the model is not equipped for this functionality in its present form.

Komoto & Tomiyama describe a product modelling framework called 'System Architecting CAD' (Komoto & Tomiyama, 2012). SA-CAD tracks system decomposition, it models parameter relations, and performs consistency management of the parameters. An interesting aspect is that SA-CAD could eventually store design knowledge used in system architecting independently from specific engineering disciplines. Unfortunately, this is not implemented yet.

8.2.3 Key Limitations

- Presence of the knowledge is the basis of a reliable design provided that it is relevant and well applied. This makes relevant and complete knowledge a binding condition to enable a satisfactory design process;
- Application of knowledge is not enough; knowledge should be anchored in the project;
- The current methods, that do track knowledge in the design are difficult to apply for enhancement of intelligent gating.

8.3 Methodology to Implement Knowledge Based Intelligent Gating

To address the key limitations, the method for *concurrent gating* will be combined with the decomposition of information in design. These two methods will form the basis of the *constituent roadmap*.

Subsection 8.3.1 will inventory how knowledge can be embedded, and Subsection 8.3.2 will define the three phases of the roadmap. Finally, Subsection 8.3.6 will explain how the *constituent roadmap* addresses the key limitations.

8.3.1 Knowledge in Axiomatic Design; How Is It Applied & Where Is It Located

Earlier investigation, based on information in design (Puik & Ceglarek, 2014a), has led to the belief that 'the quality of a design never exceeds that of the designer'. A *good design* in the hands of an ignorant designer will not be recognised as such (Suh, 2005b); this is shown by the example of an ignorant designer that does not understand how a design matrix was optimised and what the FR-DP-PV relations are. Optimisations in the perception of the designer will in the best-case lead to reinvention of the wheel; they might lead to a different *good design*, but probably will degrade the level of the design. As a result, the design matrix will be unclear, at least not decoupled, and the information content of the design will increase. Ergo, it cannot be assumed that that an ignorant designer will produce a *good design* without understanding the design relations and the decoupling of the design matrix.

It was briefly explained in the introduction that the designer is usually not a single person. In practice, it is a group of people, with various specialisms, to address problems with various natures. Typical groups of people that are involved in the development process according to the domains in AD:

- The marketer, who translates 'Customer Attributes' (CAs) into 'Functional Requirements' (FRs);
- The product designer who relates FRs to 'Design Parameters' (DPs);
- The production engineer, who searches for 'Process Variables' (PVs) that match the DPs.

The state of a design is kept in the axiomatic domains by defining its design relations (CAs, FRs, DPs, and PVs). The relations are intrinsically static; only by external activity of the mentioned marketer, product designer, and process engineer, the relations of the design will change (Figure 8.1).

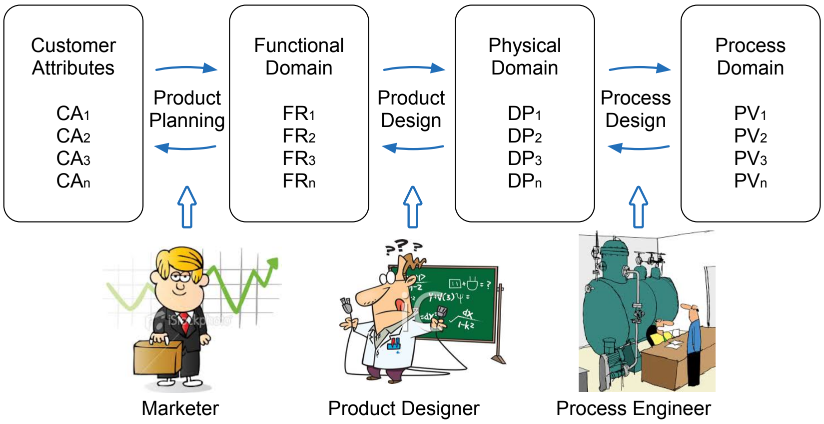


Figure 8.1 *Connections of the domains in AD*

To investigate how knowledge is applied, in order to reduce information content of the design, it is essential to understand where information and knowledge are located. AD defines the basis for this; missing or unclear relations between the domains increase the information content of the design. Clear relations, that are defined in uncoupled design matrices (or decoupled with knowledge of the order to tune the DPs), reduce the information content. Respectively the marketer, the product designer, and process engineer of Figure 8.1 apply knowledge to the design in order to regulate the relations between the domains and thus reduce the amount of information. When a sloppy marketer fails to define good relations between CAs and FRs, the FRs will not represent the qualities that a customer expects of the design. As a result, DPs and PVs will be incorrect too. Information in design can originate at either one of the three places where relations between the domains are established; product planning, product design, or the process design. Information typically tends to escalate through the successive domains in the right-hand direction (when the method of *zigzagging* is applied on a faulty FR).

Though the designer applies knowledge to the design, that knowledge is not transferred into the design. It is knowledge 'application' that leads to reduction of the information content, not the knowledge itself. The knowledge itself remains with the designer. Relevant knowledge and understanding as applied to a *good design* has provided optimal relations between the domains and, as a result, the information content in the design has disappeared.

Summarising; in order to produce a *good design*, knowledge must be applied to the relations of the domains. Recapitulating the kinds of knowledge:

- Applied knowledge is transferred to the design;
- Knowledge is implemented is in the design matrices;
- Knowledge stays with the designers and may be secured in reports for later use.

In addition to the last item, knowledge may be lost by various reasons e.g. staff turnover, poor administration, forgetfulness, etc., but can be written down in reports for application by future generations. This secures the essential data about the design and makes it easier for a successive designer to understand the design considerations by levelling his knowledge to that of the initial designers.

8.3.2 Product Design in Three Phases

Figure 8.2 shows decomposition of *useful information* as presented in Chapter 5. A *good design* has no *useful information* because knowledge of the designer is applied to the design and it completely regulates the behaviour of that design; by definition all FRs will be satisfied in this situation. To get rid of *useful information*, all three *unrecognised, recognised,* and *axiomatic information* should be addressed and eliminated. These kinds of information are coupled to respectively the explorative phase, conceptual phase and the robustness phase (conforming Table 6.1 on page 152).

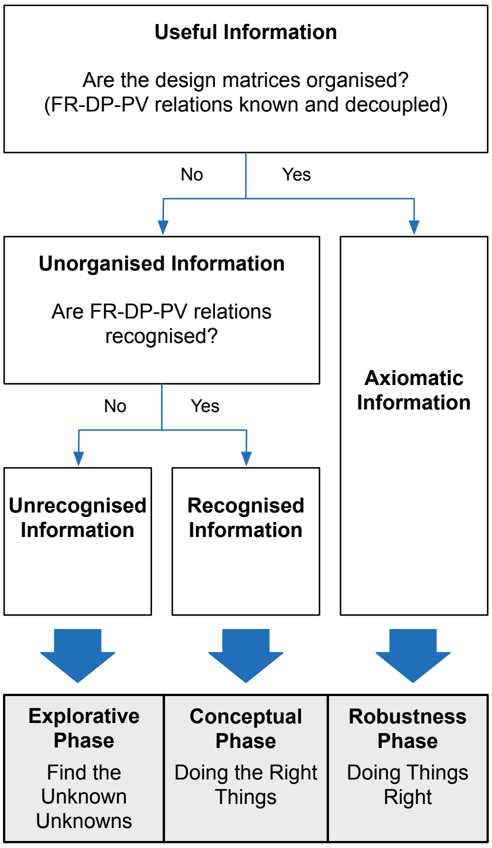


Figure 8.2 *Breakdown of information in design and how it leads to the successive phases of the Constituent Roadmap*

Chapter 6 explained how AD applies the order for the application of the axioms. Conforming this definition, *unrecognised, recognised, and axiomatic information* are addressed in that same order:

- *The explorative phase* looks for coherence in the product design process. The aim is to obtain an overview of FRs, as complete as possible at this early stage of the project. All DPs that interact with each FR are inventoried. Successively this should be continued for each DP and its PVs. The goal is to gain understanding of the design as soon as possible, and by doing this, find as much *unrecognised information* as possible.

Exploration ends with an ‘Image on the Future System’ conforming Banathy’s model (Figure 2.7);

- *The conceptual phase* validates the concept by completion and decoupling of the design matrix. Functionality is tested by iterative improvement cycles conforming the method of Chapters 3 & 4. *Unorganised information*, the aggregate of *unrecognised* and *recognised information*, is eliminated from the design;

  The conceptual phase ends with a ‘Model of the Future System’, also according to Banathy’s model;
- *Finally, the robustness phase* tunes PVs and DPs to satisfy the FRs within all operating windows of the system. This leads to elimination of all *axiomatic information*.

  This phase ends with a ‘Validated and Verified System’.

Table 8.1 gives an overview of the status of progression of the successive phases during product development. In the next section, these three phases will be applied in a general roadmap for concurrent product design.

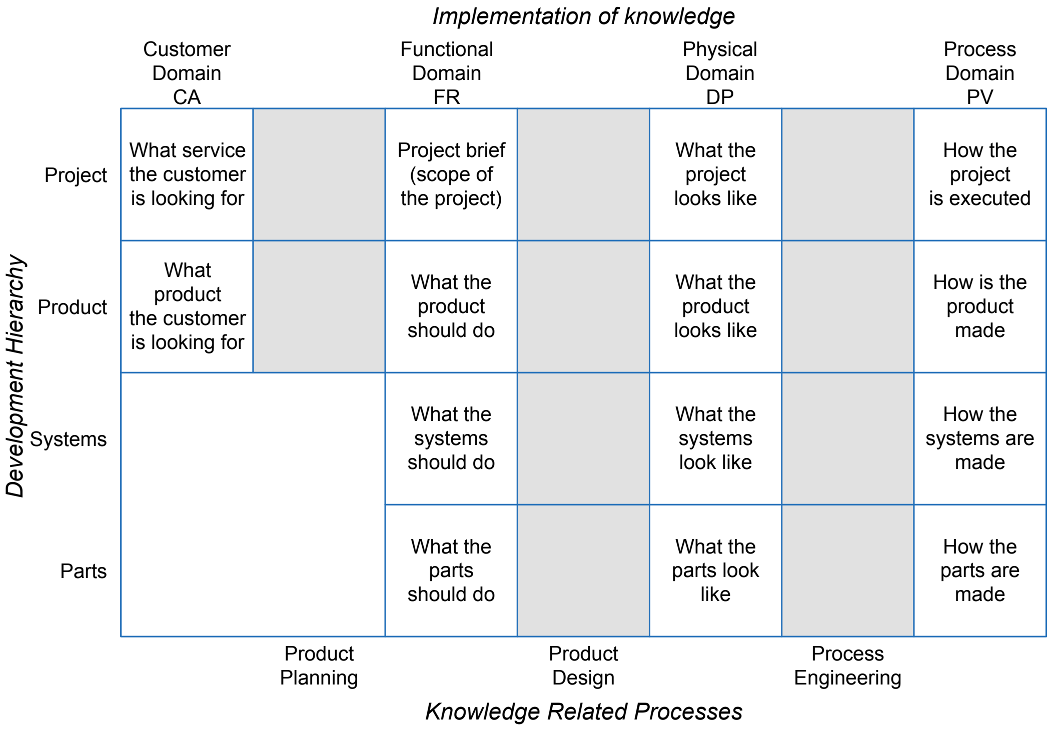
Table 8.1 *Relation between information in design and the axioms*

Phase / Status Progression	No Information Addressed	*Unrecognised Information* Addressed	*Recognised Information* Addressed	*Axiomatic Information* Addressed
Phase	No Phase Completed	Explorative Phase Completed	Conceptual Phase Completed	Robustness Phase Completed
Design Relations	Some relations known	FR-DP Rel. known DP-PV Rel. known	FR-DP Rel. known DP-PV Rel. known	All design relations fully robust
Design Matrices	Some Elements Known	Some Elements Known	Design Matrices Decoupled	Design Matrices Decoupled

8.3.3 Constituent Roadmap of Product Design

The particular feature of the *constituent roadmap* is that it tracks the implementation of knowledge but, at the same time, also tracks and archives the knowledge related processes to understand how knowledge is applied to the product design. The *constituent roadmap* is intended to be modular in itself and integrates with all models that were so far reported in this thesis e.g.: Quality Function Deployment, Qualitative Analysis, Failure Modes and Effect Analysis, Morphological Matrix, and Structured Analysis Design Technique.

The structure of the *constituent roadmap* consists of a merged matrix of 7 x 4 cells that gathers applied knowledge in the odd rows and gathers knowledge in the even rows. Its structure is based on the knowledge application model of Figure 8.1. The first part of the *constituent roadmap*, in which the odd rows are defined, is shown in Figure 8.3; there is strong resemblance with the axiomatic domains and their hierarchy. Decomposition is implemented on the vertical axis. This is analogue to the decomposition as applied in the V-Model. In this case, the levels are the same as in Chapter 4 and originating from the German V-Modell XT. Note that the customer domain is included as well. So far in this thesis, the functional domain was always the starting point for the design process. The *constituent roadmap* proposes a slightly different approach; the customer domain will be leading for the definition at Project and Product levels, but not for the Systems and Parts levels, where the functional domain is maintained as starting point. The reason is that the customer may not be expected to have the capability to deliver substantive technical input for the highly technological levels. Figure 8.4 shows how knowledge related processes are located in the even columns of the *constituent roadmap*.



Implementation of knowledge

	Customer Domain CA		Functional Domain FR		Physical Domain DP		Process Domain PV
Project	What service the customer is looking for	▨	Project brief (scope of the project)	▨	What the project looks like	▨	How the project is executed
Product	What product the customer is looking for	▨	What the product should do	▨	What the product looks like	▨	How is the product made
Systems			What the systems should do	▨	What the systems look like	▨	How the systems are made
Parts			What the parts should do	▨	What the parts look like	▨	How the parts are made
		Product Planning		Product Design		Process Engineering	

Development Hierarchy

Knowledge Related Processes

Figure 8.3 *The odd rows of the Constituent Roadmap focus on applied knowledge*

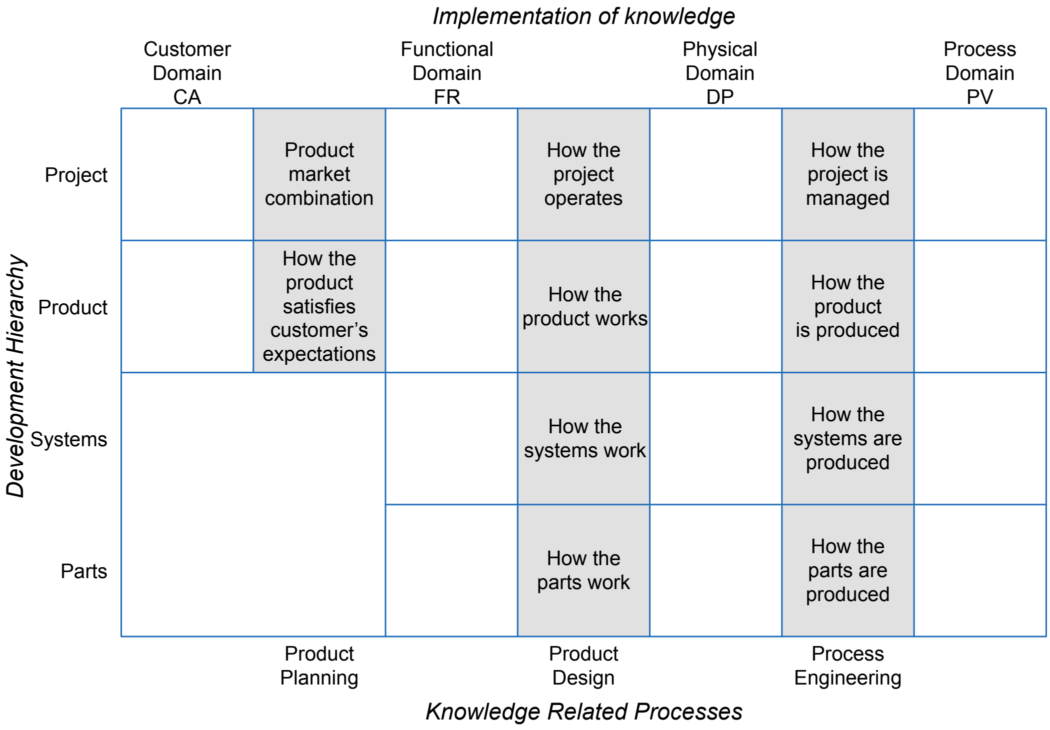


Figure 8.4 *The even rows focus on knowledge needed to understand the design*

The implementation of knowledge may be secured by documenting all models that underlie the design. These models can be system engineering models, but also numerical, marketing, and organisational models. The documentation does not only address success factors but also less successful options; knowledge about the current design will be secured in the project documentation as well as knowledge of the alternatives for the design that were not applied will also be secured including the reason to decide so.

To finalise the *constituent roadmap*, the two matrices will be combined to: (i) construct the design relations (CAs, FRs, DPs, and PVs), (ii) capture what knowledge was applied to the design, and (iii) to capture to what implementation of knowledge (or solutions) this has led (Figure 8.5).

8.3.4 The Check Matrix

The ‘Check Matrix’ is applied to track progression of the *constituent roadmap*. Its structure is based on the status of the even rows of the roadmap as shown in Figure 8.5. The corresponding design relation is represented by a number in the *check matrix* e.g.; the relation of the FRs and DPs at the ‘parts’ level in Figure 8.5 is indicated with ‘0’. This number can vary from ‘0’ to ‘3’ as the maturity of the design relation increases. The number represents the design phase (explorative, conceptual, robustness) with which that particular design relation could maximally comply:

‘0’	If it would not comply with any completed design phase;
‘1’	If it complies with a completed explorative phase;
‘2’	If it complies with a completed conceptual phase;
‘3’	If it complies with a completed robustness phase.

As such the number indicates the status of the knowledge relation between the domains. It will be used as a measure of progression considering elemental relations of the product development process.

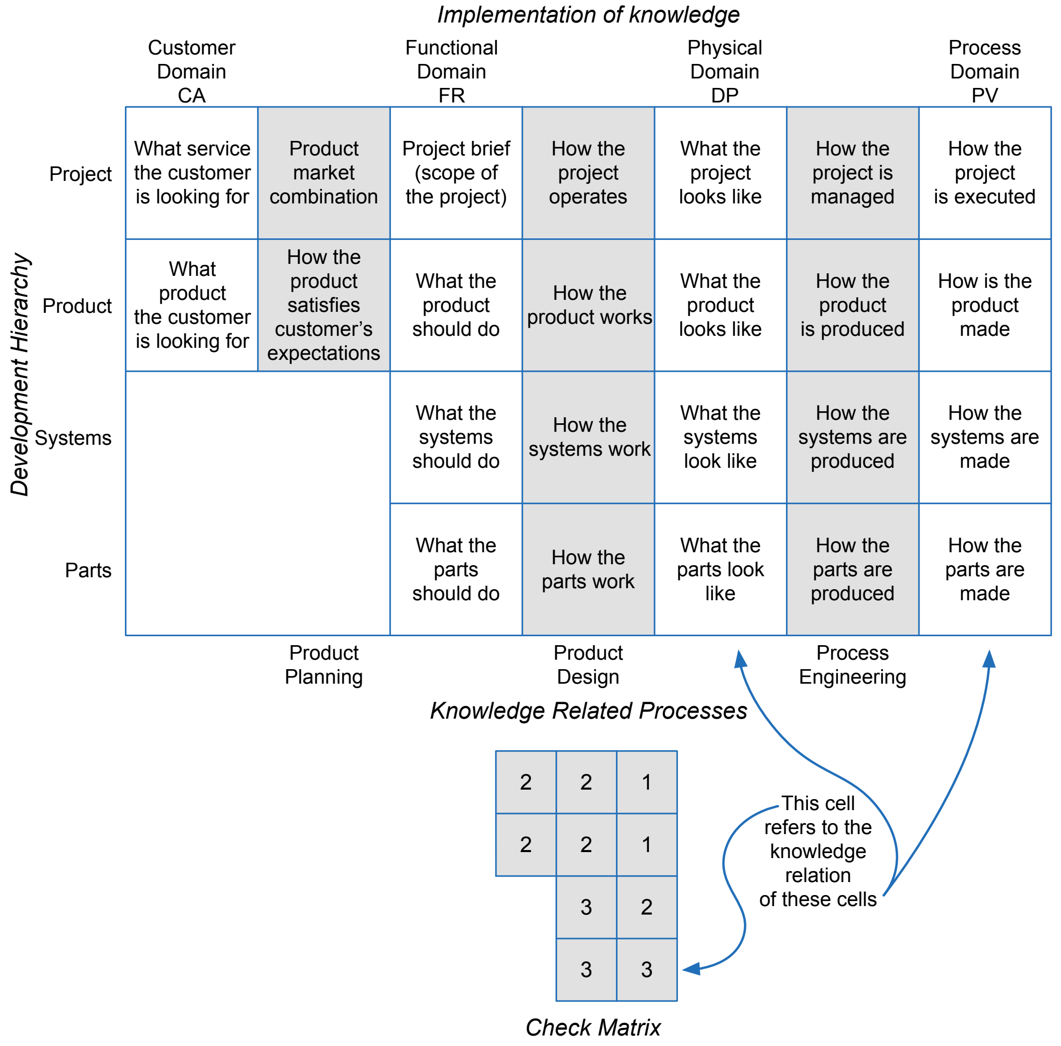
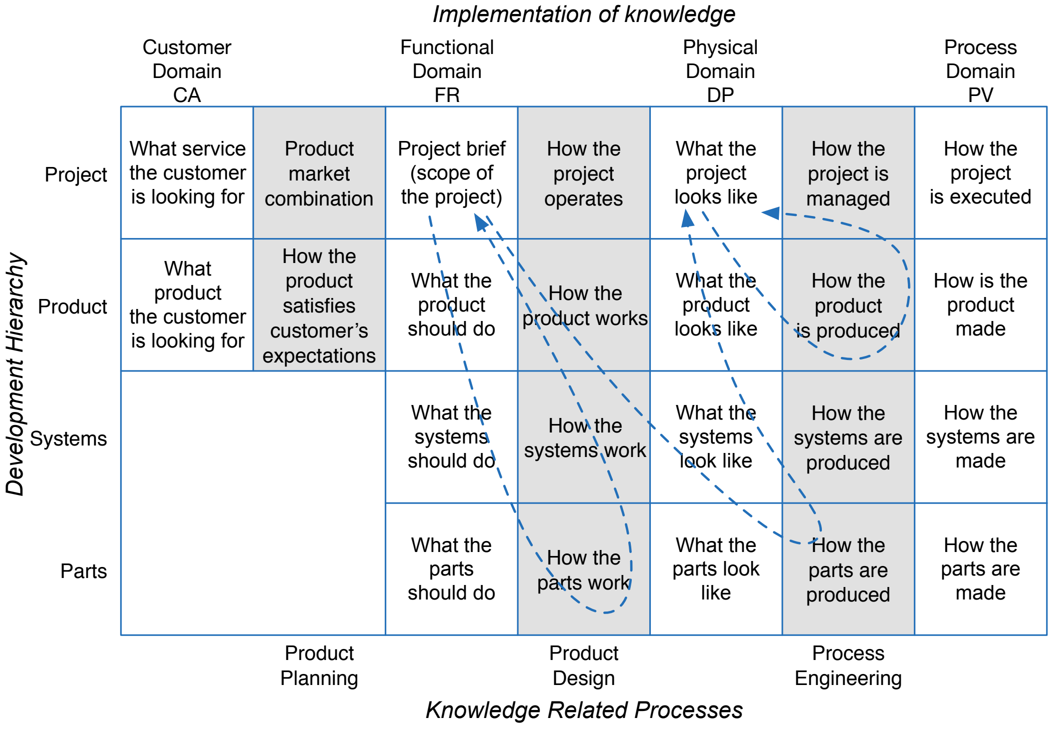


Figure 8.5 *The check matrix gives an overview of the knowledge status of the product*

8.3.5 Procedure of Application

The procedure of application follows the *CμSD framework*. Requirements specification starts at the highest level and takes place simultaneously with the decomposition of the product. Functional performance is verified by the execution of tests using the iterative approach. In the explorative phase, many uncertainties exist, and the order of addressing uncertainties is risk adjusted. The FMEA can be applied, as demonstrated in the rationale of Section 3.3.3, to determine a prioritised list of problems to address. These problems can be addressed in order of priority; by a process called ‘Yo-yoing’.



Implementation of knowledge

Development Hierarchy

	Customer Domain CA		Functional Domain FR		Physical Domain DP		Process Domain PV
Project	What service the customer is looking for	Product market combination	Project brief (scope of the project)	How the project operates	What the project looks like	How the project is managed	How the project is executed
Product	What product the customer is looking for	How the product satisfies customer’s expectations	What the product should do	How the product works	What the product looks like	How the product is produced	How is the product made
Systems			What the systems should do	How the systems work	What the systems look like	How the systems are produced	How the systems are made
Parts			What the parts should do	How the parts work	What the parts look like	How the parts are produced	How the parts are made
		Product Planning		Product Design		Process Engineering	

Knowledge Related Processes

Figure 8.6 *The full Constituent Roadmap and the process of yo-yoing*

Yo-yoing is specifically applied in the explorative phase of development. *Yo-yoing* executes the iterations of *CμSD framework,* but now its movement addresses all domains, instead of only the functional domain. As such, *yo-yoing* describes a motion through the *constituent roadmap* that is more erratic than the process of *zigzagging*. *Yo-yoing* moves not only up and down but if needed in lateral direction to address the most significant uncertainty left in the project. Like the process of decomposition, it starts at a high hierarchical level (Project or Product) where the overview on the project is maximal but jumps from there to the place where project uncertainties are highest. It performs an explorative search in order to check if knowledge is present to successfully connect the FR with a DP, or a DP with a PV. From there it bounces back to the highest level to check for the next largest risk. In this way, it pokes around in the matrix to address the largest development risks one by one. This is shown in Figure 8.6. When large amounts of quick tests need to be done to check if knowledge on a certain topic is available, as is the case in the explorative phase, *yo-yoing* offers a solution. However, if a structure needs to be tested in a sound manner, the process of *zigzagging* is preferred. The process of *zigzagging* is applied for the final check to make sure that the design is decoupled. *Zigzagging* verifies the design relations layer by layer, from left to right, until the structure of the design is completely understood. In this case for the *constituent roadmap*, *zigzagging* is the final check to validate the conceptual design, and close the conceptual phase. It is applied to check if: (i) the relations between FRs, DPs, and PVs are present, (ii) if the matrices are decoupled or uncoupled, (iii) to determine if knowledge was correctly applied to the design, and (iv) if all unused DPs and PVs are fixed. *Zigzagging* is a thorough process with a strictly defined path through the domains. If an error in the design is found, *zigzagging* stops and starts again from the point where the error is corrected. Because of this thoroughness, *zigzagging* may be a slow process compared to the agile process of *yo-yoing*. However, both serve a specific goal; *yo-yoing* is applied to learn as much as possible about the design in an as short as possible timeframe to eliminate as much *unrecognised information* as possible. *Zigzagging* is a sound process that checks if the structure of the design is completely understood.

In the next paragraphs, the three phases of the *constituent roadmap* are described as it would evolve during project execution. The *check matrix* plays an essential role in this process.

Rationale for the explorative phase: The first phase forms the exploratory part of development. The goal is to determine preliminary design relations and to find out which knowledge is relevant for the association of CAs, FRs, DPs, and PVs. It is important that no parameters are missed as such a knowledge gap may lead to *unrecognised information* in the design.

The *check matrix* starts with zeros for all relations. At first, the mission of the project (e.g. the project brief) is decomposed to form an image of the future product. A wide scope of the designer is required to make sure that no issues in the periphery are forgotten. *Yo-yoing* is applied to address immature relations in the product design using the risk-adjusted approach of the *CμSD framework*; the order of addressing the relations in the *check matrix* is determined by severity of the risks (rationale of Subsection 3.3.3 or coding may be applied in the context of qualitative risk analysis (rationale of Subsection 3.3.4). During on-going decomposition, CAs, FRs, DPs, and PVs are determined. All relations must be accompanied with a notion how satisfaction is realised e.g. how a DP relates to an FR in order to satisfy it. The *check matrix* is applied to gather the results; if all FR-DP relations of a hierarchical level are present, the respective cell of the *check matrix* is set to '1'. When all zeros have disappeared from the *check matrix*, phase one is completed. Phase one may be repeated for alternative product solutions.

In the situation that parts of the design are reused from earlier designs, the design uncertainty on the reused parts may be low. In this case, the *check matrix* me be advanced to the level that is believed realistic for that part. The procedure for assessment as was presented in Chapter 6 may be applied to determine reusability of these parts. Note that this assessment procedure was developed for RMS, so it may be applied for the DP-PV relations without further adoptions. If it is to be applied for the FR-DP relations, the method may need adjustments; adaptation to this situation should be straightforward because the concept remains unchanged.

Rationale for the conceptual phase: The second phase of conceptual validation leads to a *viable design* (proof of concept). It is based on decoupling of the matrix. Typically, the earlier mentioned set of tools may be applied to define the relations of CAs, FRs, DPs, and PVs (Figure 8.7).

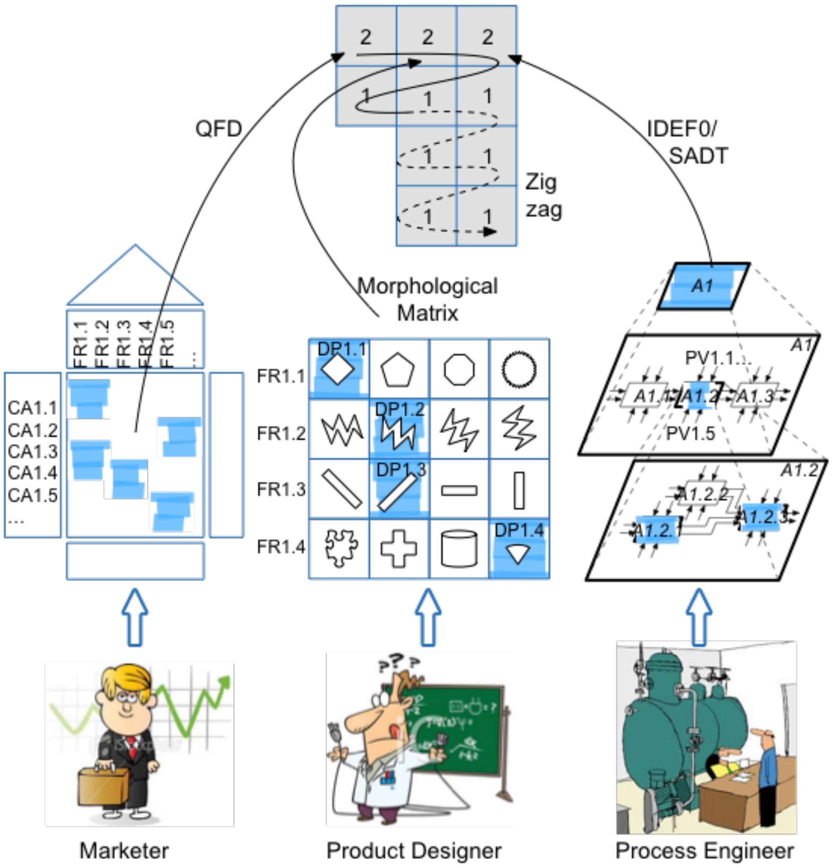


Figure 8.7 *Combining known methods for systems engineering with the Constituent Roadmap during the phase of conceptual validation*

Analogue to the explorative phase, the approach is iterative and *yo-yoing* may be applied to analyse straggling relations throughout the hierarchy and across the domains. Towards completion of the conceptual phase, *yo-yoing* is replaced by *zigzagging*.

It is possible to start the conceptual phase with more than a single product concept. However, at the end, a single most promising solution is selected for continuation to the next phase.

The respective index of the *check matrix* is increased to ‘2’ to indicate that the design relations are fully understood and decoupled. When all cells of the *check matrix* have advanced to a minimum of ‘2’, the system is conceptually solid, which means that there is a complete model from the functional and manufacturing perspective.

Rationale for the robustness phase: The final phase is executed conform good axiomatic practice. The product is made robust by matching the design range and the system range within the common statistical frameworks of Six Sigma (Yang & EI-Haik, 2008) or Robust Design (Taguchi et al., 2005) (explained Appendix B). It ensures reliable satisfaction of FRs with DPs incorporating their tolerances (same for the other design relations). This phase applies the process of *reversed zigzagging*, since the verification of robustness starts at the bottom level of the hierarchy. When *reversed zigzagging* reaches the highest hierarchical level, relations are proven robust, the *check matrix* is upgraded to ‘3’. The product is fully engineered when the complete *check matrix* is set to ‘3’. This final phase completes the implementation of all relevant knowledge in the design. The result is a *good design* (verified and validated system).

Recording of Design Information: During all three phases, the *check matrix* has a purpose in recording design information. Information that was applied to upgrade the check values, may be stored in a project database at the end of each phase (e.g. the output of the models conforming Figure 8.7). The structure, in which data is stored, can be comparable to the *check matrix*, since knowledge application concentrates at these nodes.

8.3.6 How the Constituent Roadmap Addresses the Key Limitations

The *check matrix* forms the basis of a reliable design; it is composed of the crossroads in the design that apply knowledge in the role of the product planner, the product designer, or the process engineer. Analysis of the *check matrix* is analogue to analysis of the status of the design and it follows whether knowledge was applied to understand the design relations. In essence, the *check matrix* forces the designer to gather the right knowledge and use it to structure the design. This addresses the key limitation that monitoring of the design should be based on knowledge.

The structure of the *check matrix* is also the structure that should be leading for the process of project documentation. When systems engineering models are applied e.g., by using the rationales of the *CμSD framework*, or by applying the models conform Figure 8.7, these models will intrinsically structure and secure the applied knowledge. Future project members can use the models to understand why elementary project decisions were taken.

It can be stated that the *check matrix* is a direct reflection of the knowledge required in the project. The underlying thought is that consistent design relations indicate that knowledge is applied to essential parts of the project.

8.4 Case: Design of World's Smallest Hybrid Hydrogen Sensor System

The *constituent roadmap* was applied to a new product development case; a microsystem development project that was funded by the European Committee in the perspective of a call that aims to develop ground breaking new technology for zero-power ICT systems (ECSFP, 2009). The Project 'SiNAPS' (Semiconducting Nanowire Platform for Autonomous Sensors), a granted proposal from this call, was aiming to develop a wireless 'Smart Dust' sensor system to be hooked-up to the Internet of Things.

8.4.1 Problem Definition

The project-call for zero-power ICT systems, describes its aims and how these aims should be satisfied to qualify for the reserved development resources. Basically, the call defines the CAs for the project. Project ideas for the call need to be ‘sold’ conforming customer-supplier relations by submitting a proposal that will be reviewed by the ‘customer’, in this case a line-up of delegates on behalf of the European Committee. Project SiNAPS was granted because its project proposal satisfactorily described a project within the scope of the call. SiNAPS sets an ambitious goal; the proposal describes how a miniature sensor device may be developed, that will be acknowledged as the smallest hybrid sensor system in the world (Fagas et al., 2014). Not only the way the system will be developed, realised, and integrated is accurately described in the project proposal, also the respective contributions of the participating organisations and the roles of their members with proven qualifications are explained (Fagas, 2008). Basically, the proposal conceptually defines the highest hierarchical level of the project, including high level FRs, DPs and PVs.

The technological description of the project proposal defines a sensor system that is composed of a number of microchips in a small sensor casing. Due to its hybrid nature, the parts need to be assembled one by one in a package. In total, a number of six chips (further referred to as ‘Dies’) are integrated to form an autonomous measurement platform with a target volume smaller than 4 mm^3. The concept of the SiNAPS sensor system is shown in Figure 8.8.

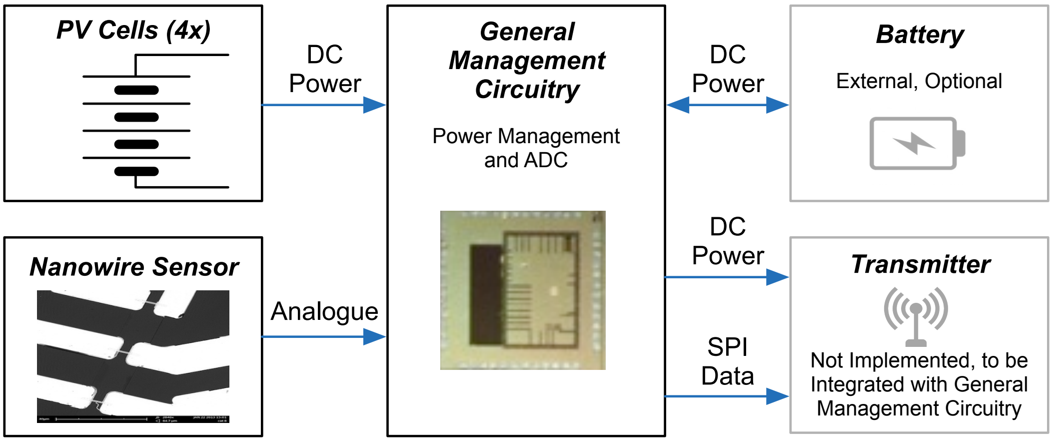


Figure 8.8 *SiNAPS Block Diagram of Sensor System*

Further, the project starts with a relatively empty slate. All silicon Dies will be designed from scratch, enabling the project to be flexible and unconventional. The radio frequent transceiver of the system (Figure 8.8) will not be implemented in order to apply maximum resources to the measurement system and power management.

8.4.2 Application of the Constituent Roadmap in the Explorative Phase

The explorative phase started with determination of the CAs, FRs, DPs, and PVs. All results were gathered in the *constituent roadmap*. Table F.1 of Appendix F shows the completed constituent roadmap for the initial status of the project, and Table 8.2 below shows the *check matrix* at this stage. What strikes in Table 8.2 are the high scores for the knowledge relations at the Project and Product levels. The scores of ‘2’ indicate that FRs, DPs, and PVs are basically decoupled and therefore already meet the conditions needed to close the conceptual phase. This high level of completion was the result of the relatively far developed project proposal, of which high demands were made to obtain funding.

Table 8.2 *Initial check matrix of the constituent roadmap*

	Knowledge about CA-FR Relation	Knowledge about FR-DP Relation	Knowledge about DP-PV Relation
Project	2	2	2
Product	2	2	2
System		0	1
Part		0	0

For the sensor system, two alternatives are available for implementation in the sensor system; both alternatives comply with the project proposal. The first sensor option, a Nanowire based biosensor for various chemical and biological measurements, is completely new and not operational yet. The second sensor option is a Nanowire based sensor that measures a single gas. This sensor uses an operating principle that is based on ‘Palladium Nanowires’ to measure hydrogen concentrations. The sensor has a less complex design that is more mature. This is confirmed when the system and part levels are inventoried. The *check matrix* is shown in Table 8.3 and the complete overview again in Table F.2 in Appendix F.

Table 8.3 *Initial check matrix for alternative hydrogen sensor*

	Knowledge about CA-FR Relation	Knowledge about FR-DP Relation	Knowledge about DP-PV Relation
Project	2	2	2
Product	2	2	2
System		1	1
Part		1	1

Basically, if the hydrogen sensor is chosen, the project could pass the exploration phase right away since a full set of design relations is known. However, in order to see if the more ambitious biosensor can be brought up to the required level, the development of the two sensors is kept in parallel for some time. After six months, substantial progression was made for the sample disposal of the biosensor, as shown in Figure 8.9. However, the selectivity of the system was not at the required level yet. The biosensor was dropped from the project and the hydrogen sensor was upgraded to the main line of research. The explorative phase of the project was closed. The *check matrix* remained as in Table 8.3.

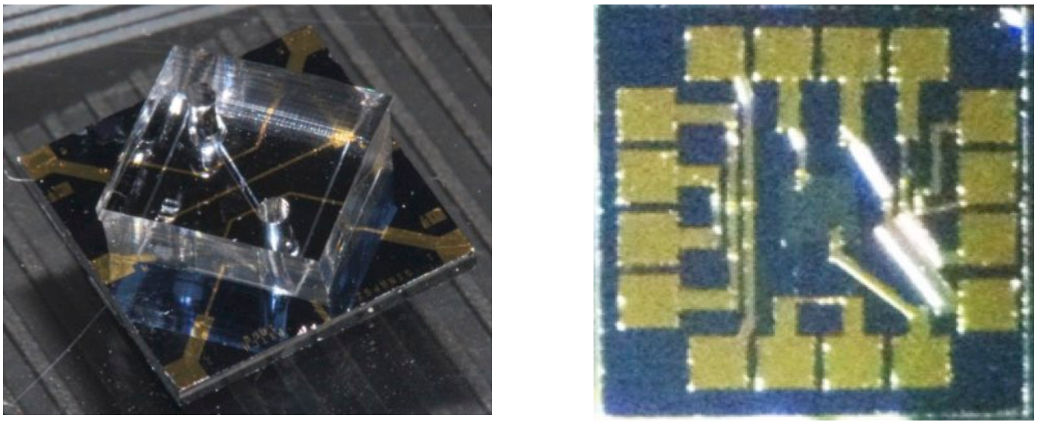


Figure 8.9 *Biosensor with sample disposal (left), Nanowire hydrogen sensor (right)*

8.4.3 Application of the Constituent Roadmap in the Conceptual Phase

The conceptual phase had the goal to concurrently develop the sensor design and its production processes. Since all Dies were using standard CMOS technology (a project constraint), standardised ways for production could be used by the application of validated design rules. The status of the *constituent roadmap*, including the final FRs, DPs, and PVs, for the conceptual phase are given in Table F.3 in the appendix. Due to the successful decomposition, the sensor consisted of six separate components that could be individually optimised. However, there appeared coupling in the sensor system itself due to the fact that the number of FRs exceeded the number of DPs for the sensor chip. FR3.2.2 ‘Measure Sensitively’ and FR3.2.3 ‘Measure with Quick Response’ are satisfied by only a single DP ‘Diameter of the Nanowire’. Though this indeed did lead to dependent behaviour, it did not jeopardise the functionality of the system; reduction of the diameter of the Nanowires leads to better sensitivity as well as to a quicker response. Therefore, DP3.2.2 could be used to satisfy both FR3.2.2 as FR3.2.3. Characterisation on older generation of Palladium Nanowire Dies had already shown that the coupling should not lead to problems (Offermans et al., 2009). Therefore, the coupling was accepted in the design.

The PV-cells and the palladium Nanowire Sensor Die were functionally tested in a number of iterative cycles. The Power Die (general power management circuitry), that was designed fully according to CMOS specification was software tested by applying the method of ‘Boundary Scanning’ (a debugging method that applies IO simulations to test the functionality of the chip design). To conclude the conceptual phase, the design matrices were derived (Table 8.4) and the *check matrix* was updated (Table 8.5).

Table 8.4 *Design matrices for the conceptual validation*

Knowledge about FR-DP Relation	Knowledge about DP-PV Relation
$\begin{Bmatrix} FR1 \\ FR2 \\ FR3 \\ FR4 \end{Bmatrix} = \begin{bmatrix} X & 0 & 0 & 0 \\ X & X & 0 & 0 \\ 0 & 0 & X & 0 \\ 0 & X & 0 & X \end{bmatrix} \cdot \begin{Bmatrix} DP1 \\ DP2 \\ DP3 \\ DP4 \end{Bmatrix}$	$\begin{Bmatrix} DP1 \\ DP2 \\ DP3 \\ DP4 \end{Bmatrix} = \begin{bmatrix} X & 0 & 0 & 0 \\ 0 & X & 0 & 0 \\ 0 & 0 & X & 0 \\ 0 & X & 0 & X \end{bmatrix} \cdot \begin{Bmatrix} PV1 \\ PV2 \\ PV3 \\ PV4 \end{Bmatrix}$
$\begin{Bmatrix} FR3.1 \\ FR3.2 \\ FR3.3 \end{Bmatrix} = \begin{bmatrix} X & 0 & 0 \\ X & X & 0 \\ 0 & 0 & X \end{bmatrix} \cdot \begin{Bmatrix} DP3.1 \\ DP3.2 \\ DP3.3 \end{Bmatrix}$	$\begin{Bmatrix} DP3.1 \\ DP3.2 \\ DP3.3 \end{Bmatrix} = \begin{bmatrix} X & 0 & 0 \\ 0 & X & 0 \\ 0 & 0 & X \end{bmatrix} \cdot \begin{Bmatrix} PV3.1 \\ PV3.2 \\ PV3.3 \end{Bmatrix}$
$\begin{Bmatrix} FR3.1.1 \\ FR3.1.2 \end{Bmatrix} = \begin{bmatrix} X & 0 \\ 0 & X \end{bmatrix} \cdot \begin{Bmatrix} DP3.1.1 \\ DP3.1.2 \end{Bmatrix}$ $\begin{Bmatrix} FR3.2.1 \\ FR3.2.2 \\ FR3.2.3 \end{Bmatrix} = \boxed{\begin{bmatrix} X & 0 \\ 0 & X \\ 0 & X \end{bmatrix}} \cdot \begin{Bmatrix} DP3.2.1 \\ DP3.2.2 \end{Bmatrix}$	$\begin{Bmatrix} DP3.1.1 \\ DP3.1.2 \end{Bmatrix} = \begin{bmatrix} X & 0 \\ 0 & X \end{bmatrix} \cdot \begin{Bmatrix} PV3.1.1 \\ PV3.1.2 \end{Bmatrix}$ $\begin{Bmatrix} DP3.2.1 \\ DP3.2.2 \end{Bmatrix} = \begin{bmatrix} X & 0 \\ 0 & X \end{bmatrix} \cdot \begin{Bmatrix} PV3.2.1 \\ PV3.2.2 \end{Bmatrix}$

Table 8.5 *Check matrix for proof of concept*

	Knowledge about CA-FR Relation	Knowledge about FR-DP Relation	Knowledge about DP-PV Relation
Project	2	2	2
Product	2	2	2
System		2	2
Part		2	2

With all elements of the *check matrix* at a minimum of ‘2’, the conceptual phase was completed.

8.4.4 Application of the Constituent Roadmap in the Robustness Phase

The third and last phase of the *constituent roadmap* focusses on the robustness of the design. Chapter 7 has shown that the complexity of a system may be addressed by testing the system under various windows of operation, or by complete understanding of the design relations (matrix and statistical relation). In this case, a combination is applied.

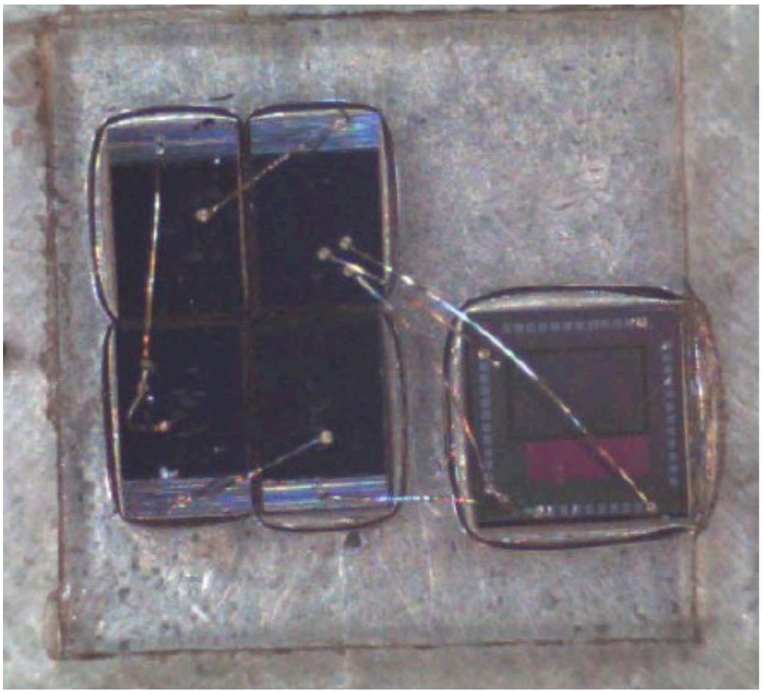


Figure 8.10 *Test setup to characterise PV cells and Power Manager Die*

The PV cells of the sensor were tested in a setup with the charge circuit of the Power Manager as shown in Figure 8.10. In parallel, the hydrogen sensor was separately tested by connecting it to the ADC circuit. After testing, when it proved to be fully functional, the sensor system was fully assembled.

All results were carefully written down in reports, and because this concerned a research project, most of it was reported in academic papers. Table 8.6 shows the papers that were produced during validation of the final phase of the project. All lower levels were reported in academic papers. However, the project level was not published in academic papers but confidentially reported to the customer, (European Committee). The project report was officially accepted.

Table 8.6 *Provided documentation after robustness phase*

	Knowledge about CA-FR Relation	Knowledge about FR-DP Relation	Knowledge about DP-PV Relation
Project	Final Report SiNAPS Acceptance letter European board	Final Report SiNAPS	Final Report SiNAPS
Product	Acceptance letter European board	(Fagas et al., 2014)	(Fagas et al., 2014)
System		(van der Bent et al., 2015; Khosro Pour et al., 2012)	(Lafeber et al., 2014; Khosro Pour et al., 2013; Tong et al., 2010)
Part		(van der Bent et al., 2010)	(Tong et al., 2010)

The final sensor is shown in Figure 8.11. The Dies are placed on a substrate of just over 4 mm^2. Combined with low thickness of the chosen concept, that was under 1 mm in total, the goal of max 4 mm^3 was met.

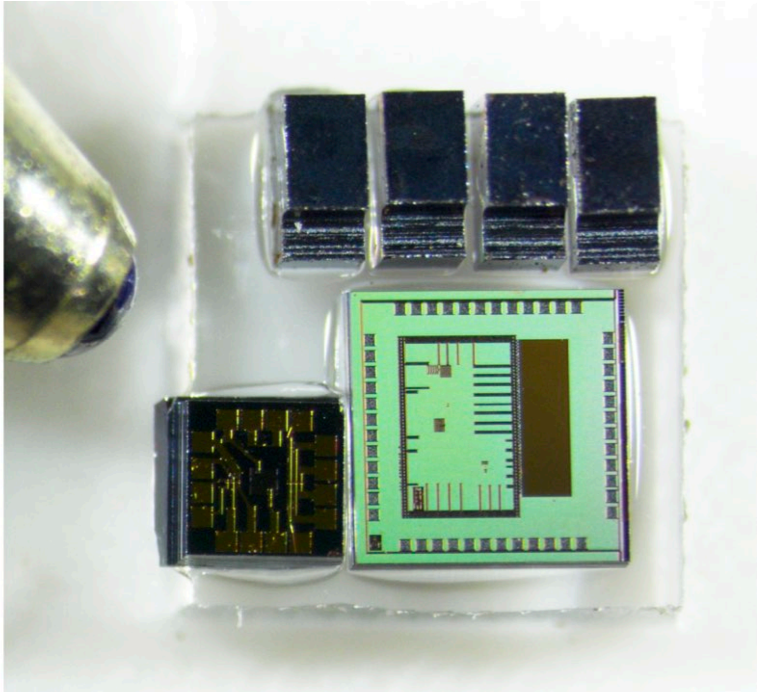


Figure 8.11 *World's smallest hydrogen sensor after completion*

8.5 Discussion

In essence, the *constituent roadmap* is a knowledge-tracking model. In the first phase, it monitors the presence of necessary knowledge during the search for coherence, in the second phase, it monitors if this knowledge was applied to acquire a model to support conceptual validation. Finally, in the third phase, it checks whether knowledge was applied to implement solutions according to good practice. The *check matrix* visualises progression in a consistent way throughout the development process. This enables easy access for less-experienced users. The roadmap is based on information in AD, in which the unorganised component, related to the Independence Axiom, was split into an unrecognised and a recognised part. Users are stimulated to designate the relevant knowledge in the design, before making final conceptual choices when the milestone 'proof of concept' approaches. This increases the resolving power of the method.

8.5.1 Strengths of the Constituent Roadmap

The way of knowledge stocktaking for the product and its domains, in relation to the hierarchy of the V-Model (V-Modell- XT) is new. Also new is the way this knowledge is visualised in the *check matrix* that focusses on all knowledge related processes. Together it combines accessibility of the scientifically forceful method of Axiomatic Design, yet dealing with the plenitude of the product design in a structured way. Due to the visual nature of the model, it is suitable as a universal language to improve understanding between all stakeholders in the organisation, whether they are managers, staff or technicians. Especially since the roadmap is simple to apply; its use is attractive to all parties.

The method retains its solid academic basis due to the application of AD. Many ways to combine AD with other systems engineering methodologies have been reported in the past. The *constituent roadmap* may benefit from these combinations comparably. However, the application of a conceptual phase, with divided attention for exploration and validation, analogue to the work of Banathy (Banathy, 1996), Pahl & Beitz (Pahl & Beitz, 2013), Ulrich & Eppinger (Ulrich & Eppinger, 2004), and Wang (Wang et al., 2002), increases resolving power in the early phase of design. The dynamics of convergence and divergence according to Banathy (Figure 2.7) can be seamlessly mapped to the *constituent roadmap* and is expanded to the rear. The explorative phase will lead to 'the image of the future system, 'conceptual validation' leads to 'the model of the future system', and the phase 'gain robustness' will lead to a new goal: 'a validated system'.

8.5.2 Weaknesses and Limitations of the Constituent Roadmap

The *constituent roadmap* also has shortcomings. Complete underpinning of the *check matrix* can be laborious, especially when descending in the hierarchical tree when it tends to gain in width. Quite a number of design relations have to be scanned before complete understanding of knowledge application can be guaranteed. Moreover, dutiful application of the roadmap does not relieve the designer of the need to collect relevant design knowledge, and thorough understanding of the design remains essential. The *constituent roadmap* may be easy to apply, but cannot unburden the designer of gathering design knowledge.

As said, communication within the company could improve when the *constituent roadmap* is used. However, a limiting factor is that this only works when some instruction about the method is given to its users.

8.5.3 Other Considerations

The *constituent roadmap* combines an implementation of *concurrent gating* and *intelligent gating*. *Concurrent decomposition* was maintained by using the hierarchy of the V-Modell XT and as such supports *concurrent gating*; *zigzagging* is applied by descend through the hierarchy level after level.

On the other hand, knowledge stocktaking forms the basis for the *constituent roadmap*. The method forces knowledge application in the design and its understanding by the designers. The design must be substantiated before the *check matrix* can be upgraded.

8.5.4 Opportunities for Further Improvement

The Axiomatic Maturity Diagram of Chapter 7 was not yet integrated with the Constituent Roadmap. The reason is that the methods to respectively determine the absolute amount of information in the design, and the project status in the Axiomatic Maturity Diagram, are not well-developed yet; so far it was applied as a qualitative tool for analysis of projects in retrospect. However, the Axiomatic Maturity Diagram can be combined in good harmony with the *constituent roadmap* to analyse and recover from eventual errors in the design.

8.6 Conclusions

The *constituent roadmap* may serve as a model to track product development from the earliest stage to market introduction. Two new features are characteristic for the *constituent roadmap*. First, the way of knowledge stocktaking for the product and its domains may be considered novel. Secondly, the way this knowledge is visualised in the *check matrix* focusses on all knowledge related processes. Together, it combines accessibility of the scientifically vigorous method of Axiomatic Design, yet dealing with the plenitude of development challenges in product design in a structured and clear way.

CHAPTER 9
GENERAL DISCUSSION

The chapters in this thesis all ended with a discussion that reflected on the individual chapters. This general discussion is used to reflect on the research questions of the thesis, and issues that go beyond the scope of the individual chapters. The research questions as defined in the introduction were:

RQ1:

How do project uncertainties, during the development process of microsystems and their production means, evolve as the project progresses?

RQ2:

How can this knowledge be applied in a protocol to guide engineers effectively and concurrently through the development process of microsystems and their production means?

The thesis was split into two parts; the first part has addressed RQ1 and the second part focussed on the implementation of that analysis to address RQ2. RQ1 will be discussed in Section 9.1, RQ2 will follow in Section 9.2.

9.1 Discussion of Methods to Monitor How Project Uncertainties Evolve as the Project Progresses

9.1.1 Strengths of the Proposed Methods

Firstly, the *μSD* and *CμSD frameworks* give quick feedback on the product development process for microsystems. As such, it enables agile optimisations of the product design. The *CμSD framework* adds the concurrent approach of microsystem design for production means. Therefore, the agile advantage provides benefits to the development of products and process technology.

Secondly, *intelligent gating*, enabled by the decomposition of information in design, explains that lacking knowledge leads to 'irregularities in design'. This causes a design to have a shortage of structure and organisation and it cannot be guaranteed that such a design operates as required; in terms of AD, there is no certainty that the FRs are satisfied by the DPs (the same applies to DPs and PVs).

Thirdly, the information theory teaches us that the bigger the design challenge, the more irregularities are introduced in the system, and the more likely that something may go wrong in the development process. This uncertainty calls for: (i) a conserved use of immature technology, and (ii) increased appreciation of proven technology. A modular approach in product and manufacturing design may reduce uncertainties when modular parts can be reused, provided that the modular parts are well engineered and documented.

Fourthly, graphical simulation of the development process by application of the Axiomatic Maturity Diagram, increases understanding of typical errors that are made during the design process. The results of this investigation will help to recognise how the errors are caused, what the consequences of the errors are, and how they can be addressed. The Axiomatic Maturity Diagram may be helpful to communicate these causes and consequences within the team of designers, between departments, managers and technicians, or even between companies.

Finally, *unrecognised information* is the most troublesome of problems in the development process. There are two main ways to deal with this kind of information. One way is to eliminate irregularities in the system by organising the design from top to bottom; this is basically what the *zigzagging* process in AD does. The other way is by subjecting the product, or functional elements of the product if it is not completed yet, to proper testing. This is preferably done as early as possible to prevent errors from escalating. Designers, managers, or humans in general, easily let themselves be fooled by their perception of the design and may overlook shortcomings that will lead to unexpected functional behaviour. Physics, or nature in general, cannot be fooled* and by execution of proper tests it will uncover any difference in functionality.

9.1.2 Weaknesses of the Proposed Methods

Application of information in design, as described in this thesis, has also weaknesses. The largest drawback is that there is yet no absolute certainty that *unrecognised information* is found before it may escalate to serious proportions. Even if a design matrix is defined and decoupled, there could be a hidden DP that influences one or more FRs and basically causes coupling of the design. There does not have to be a sign of any kind to warn the designer that there is hidden information in the system. Also, testing does not guarantee that the error is found. Even with a good test-plan, it may happen that a hidden design parameter is not varied. There is a substantial chance that this actually happens; as the DP was not recognised in the first place, probably the test-plan is not targeting this DP either.

---

* *Exclude some novel quantum-mechanical insights from this statement*

Another weakness is that there is a significant chance of ‘over-engineering’ by some designers. The challenge for total understanding of the product design may lead to analysis-paralysis. This response is also known as an ‘anti-pattern’ (Koenig, 1998), a response to a recurring problem that is usually ineffective or even counterproductive. It may lead to a designer being afraid to release the product because he is not sure about many details in the design that might cause unwanted behaviour.

9.1.3 Limitations of the Proposed Methods

Application of information in design as described in this thesis also has its limitations. The method for tracking projects, using the Axiomatic Maturity Diagram in its current state of development, is mainly useful in retrospect. It cannot be applied on problems that are not known yet.

9.1.4 Opportunities for Further Improvement

Time-dependence of a design or system adds substantial uncertainty. It was not yet investigated and remains for future research. However, also for *time-independent information*, there is still room for improvement in determination of the real position in the Axiomatic Maturity Diagram. Chapter 6 has reported a number of measures for quantification of the axes that could bring determination of the real position to a higher level. Note that there will be many drawbacks when trying to apply these measures; due to its nature, *unrecognised information* is not easy to predict and though a quick scan of these methods showed opportunities for investigation, there does not seem to be a quick solution so far. A fairly new opportunity could be to apply an adapted version of the method of inventive problem solving (TRIZ). As is, the method is able to contribute to the synthesis of solution concepts. It could be modified to find potential risks when inventive principles are applied. A quick literature scan learns that some work has been done in this direction e.g. Regazzoni & Russo present an improved risk management model for product and system design to reduce failure occurrence based on TRIZ and FMEA (Regazzoni & Russo, 2011). Teoh & Case published a knowledge modelling procedure based on FMEA that is particularly suitable for automation (Teoh & Case, 2004).

9.2 How to Guide Engineers Through the Development Process?

9.2.1 Strengths of the Proposed Methods

The *constituent roadmap* combines a number of strengths to explore, define, and reduce information in design. First, the principle of *yo-yoing*, as was explained in Chapter 8, implements the iterative feedback loop of the *(C)μSD framework* and as such enables early testing in the development procedure. This is analogue to the probe-sense-respond approach as suggested for complex systems by the Cynefin framework. It takes place in the early stage of product design when invested resources in the project are still low. *Yo-yoing*, increases the chances of finding *unrecognised information* in the design and reveal it as early as possible. The *constituent roadmap* implements the process of *zigzagging* in the conceptual phase, explained in chapter 4, as a final and exhaustive check of the design relations and their decoupling. During the robustness phase, the process of *reversed zigzagging* enforces testing bottom up as the product is composed from its elementary parts. Again, all design relations are tested, however now they are tested for statistical rigour. It is the third and final check that may reveal *unrecognised information* in the design before the product is released. A particular strength of the *constituent roadmap* is that it intrinsically generates product design documentation. This is of advantage for maintenance or further development of the system in a later stadium.

9.2.2 Weaknesses of the Proposed Methods

The weaknesses of the *constituent roadmap* are also related to the nasty characteristics of *unrecognised information* as explained in Section 9.1. Though thorough application of the procedures proposed by the *constituent roadmap* during exploration, conceptualisation, and search for robustness, indeed do contribute to understanding of many project issues, it is still not guaranteed that the actual goal, 'complete understanding of all design relations' will be reached. The reason is that a 'completely regulated system', that includes the work of all suppliers, operations, and materials is exceptionally diffuse and therefore not realistic.

Secondly, the definition and understanding of every detail in a design requires a lot of work; at the lower hierarchical levels, the width of the decomposition trees gets exponentially large. The question is, who is willing to spend substantial effort for something that cannot guarantee full functionality of the design?

Finally, though the method was kept as simple as possible, it requires quite a number of skills to apply the *constituent roadmap*. As the *constituent roadmap* is built on both AD and the V-Model, these methodologies of system engineering also have to be learnt in its periphery.

9.2.3 Limitations of the Proposed Methods

A limitation of the *constituent roadmap* is that the methodology does not expand knowledge. It even does not indicate where or how the missing knowledge can be acquired. It basically guides the designer in his search for *unrecognised information* and, since *unrecognised information* instantly changes to *recognised information* after it is spotted, it leaves fresh *recognised information* flagged to be addressed by the designer. This is only sensible if the designer has the capability and time to acquire the essential knowledge to bring this to successful conclusion.

Another limitation is that the *constituent roadmap* does not bring the design to a higher level of absolute maturity. All actions are based on the current design as is: it does not renew the concept of the design nor does it stimulate the designer to explore new areas. The reality is that it does rather detract from innovation, since most changes come with a lot of uncertainty. It could make the approach rather conservative.

9.2.4 Opportunities for Further Improvement

Within the scope for improvement it would be useful to investigate how innovation can be safely realised within the *constituent roadmap*. Though the *constituent roadmap* does not stimulate innovation, it as well does not hinder it. The same applies for a modular approach; reusing existing modular parts may not be good for innovation but, with good definitions, new modules can be developed which increases innovation. The modular approach supports the *constituent roadmap* since documented modules can be easily reused and assessed within the methodology. So, when combining the *constituent roadmap* with a modular approach, as was done in Chapter 7 for RMS, focus may be maximised on the innovative new modules, while leaving the other modules in the safe and known design space. This could be supported with a software like Acclaro Design™ that has the ability to incorporate existing and known modules of the design as commodities or commercial off the shelf parts.

9.3 Other Considerations

The information theory of Shannon & Weaver, as was explained in Chapter 5, shows that imperfect knowledge of a situation leads to random behaviour. Since it is not possible to exhaustively understand all phenomena in our environment, and it is also not possible to control all these phenomena, it means that the future behaves randomly up to some extent. This is the case for the phenomena that have not been regulated. Things, of which we can guarantee that they have no irregularities, will behave in a structured way. This means that in principle, the future behaves randomly, however things can be conditioned by the application of our knowledge. In this thesis, this is applied for product design and manufacturing of microsystems. The application of this insight goes beyond product design and manufacturing. In this research, a lot of work has been dedicated to the combination or development of suitable and practical models to understand the product design and its manufacturing means; parts of it have proved successful. For phenomena that are not understood, a safety net has been applied in the form of applied testing. The reason that testing is a powerful method to find hidden problems in the design is because ‘people get fooled, but not physics’ (Puik, this thesis). Therefore, testing appears a good procedure to investigate complex systems or parts of it. This is not surprising since it matches the Cynefin method of probe-sense-respond, where probing is an examination of quality, analogue to testing, and sensing is the awareness that follows from the applied test. The examination of quality may be quite broad, since the effect is not known in advance, and this makes it a diverging process. The awareness that follows from the test is converging because new knowledge is acquired and that knowledge confirms or negates the prevailing hypotheses. Probe-sense-respond as investigative cycle is a diverging and converging process. Altogether, this diverging-converging process is applied many times at all levels of the design. As such, the three stages of the *constituent roadmap* can be visualised in Banathy's model of Divergence and Convergence by expanding it to the back. In Figure 9.1 the three investigated models for systems engineering are plotted synchronously in time. Though, the exploration stage only has been distinguished in Banathy's model, its activities are also executed within AD. The image of the future system is characterised by: hierarchical break down, preliminary FRs, proposed DPs for every FR, a number of alternatives in design, and customer accordance with the image of the future system.

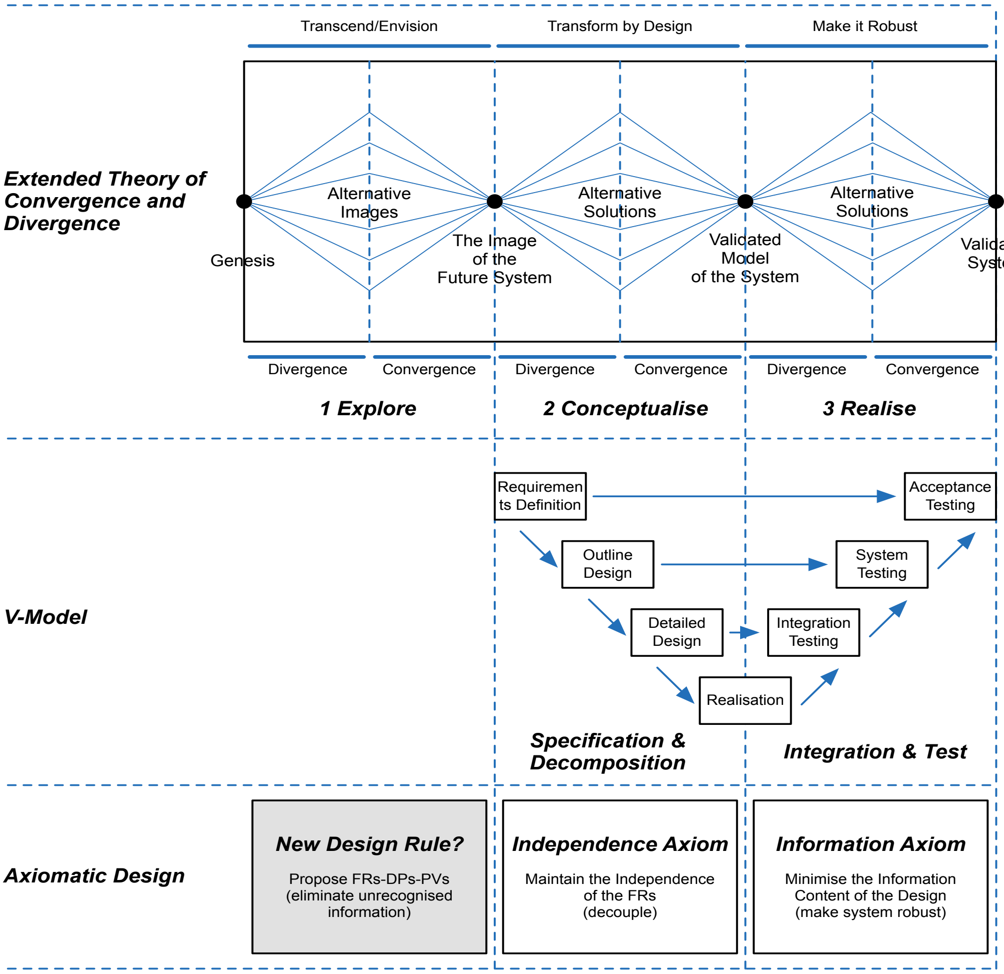


Figure 9.1 *The three stages of the constituent roadmap by extending Banathy's model of divergence and convergence*

9.4 Opportunities for Further Improvement

The exploration stage within the framework of AD, in Figure 9.1 marked with the grey block ‘New Design Rule?’ could need further investigation. Errors made during the explorative phase are disruptive for decoupled designs, comparable to the way in which the Independence Axiom is disruptive to the Information Axiom (as was explained in Subsection 6.2.1). The goal of a designer should be to reduce all *unrecognised information* before starting the process of decoupling. The solution for this problem is to pursue a complete set of design relations during the explorative phase. A complete set of design relations requires: (i) a complete set of FRs, (ii) awareness of all DPs that affect each FR, and (iii) awareness of all PVs that affect each DP. Even if the design cannot be decoupled yet, there is no *unrecognised information* left in the system.

As explained in paragraph 9.2.2, it can be quite an amount of work to understand all of these design relations. Fortunately, there are ways to structure the design effectively, e.g. by applying: (i) commercial off the shelf parts from third parties that are specified by the supplier, (ii) reused solutions from the past that were specified by the design team itself, (iii) experience of the designer (similar to reuse but more flexible since it is based on knowledge instead of knowledge application), and (iv) standards that describe interfaces and the use of systems.

The question is what kind of precept this new design rule is. The new design rule does not appear to be an axiom itself. Even the Information and Independence Axioms are no true axioms in the sense that they cannot be derived from a higher truth; Chapter 5 has shown that they can be derived from *useful information* and/or the Complexity Axiom, the latter still holding the status of a true axiom. Note that this new design rule may be a corollary of the Complexity Axiom. Another perspective is that it is a corollary of the Independence Axiom, because the presence of a complete set of design rules is a binding condition to decouple the design. This perspective would relate the Independence Axiom completely to *unorganised information* (Figure 5.3). This last viewpoint does not harmonise well with the disruptive character of the new design rule, which suggests that finding a complete set of design relations is a separate design stage to be completed before decoupling the system. The exact definition of these options remains open for future investigations.

9.5 Conclusions

Successful development of microsystems may be a great enabler for the product innovation of companies, but the development of these products and their industrialisation comes with many undesirable risks. In the context of risk mitigation, the following outcomes are the result of the research described in this thesis:

i. The μSD, CμSD, and CμSD with *intelligent gating* combine the strengths of iterative and sequential project monitoring methods. As such, they combine the strengths of Scrum, as well as the V-Model, which leads to agile feedback of the performance of the design and rigour in project execution;

ii. Decomposition of possible development problems, when developing (micro)systems, leads to three kinds of information in design to be addressed; *unrecognised*, *recognised*, and *axiomatic information*. The first two kinds are newly defined and relate to the ‘unknown unknowns’ and the ‘known unknowns’ in a system;

iii. These kinds of information in design are strongly related to the Theory of Complexity in Axiomatic Design and may be addressed by a ‘Probe-Sense-Respond’ sequence of actions. *Axiomatic information* is the kind of information that was originally defined in Axiomatic Design, it addresses the robustness of the design;

iv. A number of systems engineering models were proposed to monitor the project risks as they develop during project execution at three levels of abstraction to: (i) to monitor project progression, (ii) to compare *reconfiguration schemes* for RMS, and (iii) a combined framework of the different models in the thesis;

v. This combined framework is called 'Constituent Roadmap' The *constituent roadmap* uses many elements of Axiomatic Design. It consists of three phases: (i) an explorative phase that leads to a number of potential products and defines the first FRs and DPs. A *yo-yoing* motion in this stage, through the product's hierarchical structure, explores the concept of the design by forcing quick tests of the many ambiguities in the design to bring them at an equal level, (ii) a conceptual phase decouples the FRs by a *zigzagging* motion through the hierarchical structure of the system, and (iii) the system is made robust by testing in a *reversed zigzagging* motion that is oriented bottom up through the hierarchy of the system.

Models for systems engineering generally try to bring project risks, latent in the product development process, towards the present. As such, these risks can be addressed in the early stage of development when investments are still low. The models in this thesis have this same characteristic; they may be combined or expanded with new or other existing ways for systems engineering. It is not possible to guarantee that project execution during development of microsystems has no disturbances, even with proper application of the models, but chances of errors remaining in the design will be substantially reduced.

REFERENCES

Abdel-Malek, L. & Wolf, C. (1991). Evaluating flexibility of alternative FMS designs; A comparative measure. *International Journal of Production Economics*. 23 (1-3). pp. 3–10.

Abdel-Malek, L. & Wolf, C. (1994). Measuring the impact of lifecycle costs, technological obsolescence, and flexibility in the selection of FMS design. *Journal of Manufacturing Systems*. 13 (1). pp. 37–47.

Abdi, M.R. & Labib, A.W. (2003). A design strategy for reconfigurable manufacturing systems (RMSs) using analytical hierarchical process (AHP): A case study. *International journal of production research*. 41 (10). pp. 2273–2299.

Abdi, M.R. & Labib, A.W. (2007). Feasibility study of the tactical design justification for reconfigurable manufacturing systems using the fuzzy analytical hierarchical process. *International journal of production research*. 42 (15). pp. 3055–3076.

Akao, Y. (2004). *Quality Function Deployment*. London: Taylor & Francis.

Albano, L. & Suh, N.P. (1994). Axiomatic design and concurrent engineering. *Computer-Aided Design*. 26 (7). pp. 499–504.

Amico, M., Asl, F., Pasek, Z. & Perrone, G. (2006). Real Options: An Application to RMS Investment Evaluation. In: *General RMS Characteristics. Comparison with Dedicated and Flexible Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 675–693.

Andreasen, M.M., Howard, T.J. & Bruun, H.P.L. (2014). Domain Theory, Its Models and Concepts. In: A. Chakrabarti & L. T. M. Blessing (eds.). *An Anthology of Theories and Models of Design*. London: Springer London, pp. 173–195.

Anitha, P.C., Savio, D. & Mani, V.S. (2013). Managing requirements volatility while 'Scrumming' within the V-Model. *2013 IEEE Third International Workshop onEmpirical Requirements Engineering (EmpiRE)*. pp. 17–23.

Asimow, M. (1962). *Introduction to design*. Upper Saddle River, New Jersey: Prentice Hall.

Ayag, Z. (2007). An integrated approach to evaluating conceptual design alternatives in a new product development environment. *International journal of production research*. 43 (4). pp. 687–713.

Bacon, F. (1620). *Novum organum*. London: Forgotten Books.

Bajaj, P. & Narang, P. (2012). Perking Up V-Model to VM-Model. *International Journal of Engineering and Innovative Technology IJEIT*. 1 (5). pp. 236–238.

Banathy, B.H. (1996). *Designing Social Systems in a Changing World*. Boston, MA: Springer Science & Business Media.

Benkamoun, N., ElMaraghy, W., Huyet, A.-L. & Kouiss, K. (2014). Architecture Framework for Manufacturing System Design. *Procedia CIRP*. 17. pp. 88–93.

Boehm, B.W. (1988). A spiral model of software development and enhancement. *Computer*. 21 (5). pp. 61–72.

Boehm, B.W. (1979). Guidelines for Verifying and Validating Software Requirements and Design Specifications S. A. Samet (ed.). *IEEE Software*. pp. 1–20.

Boehm, B.W., Brown, J.R. & Lipow, M. (1976). *Quantitative evaluation of software quality*. pp. 592–605.

Boothroyd, G. (2005). *Assembly automation and product design*. Boca Raton, Florida: CRC Press.

Booton, R.C. & Ramo, S. (1984). The Development of Systems Engineering. *IEEE Transactions on Aerospace and Electronic Systems*. AES-20 (4). pp. 306–310.

Brands, D. & Weert, P. (2000). *Beter op weg naar BEST*. Eindhoven: Technische Universiteit Eindhoven, Faculteit Technologie Management Technische Bedrijfskunde.

Brecker, K. (2005). Micro- and Nanotechnologies for Advanced Packaging. In: *Micro- and Nanotechnologies for Advanced Packaging*. MST News, January 2005, pp. 1–3.

Brillouin, L. & Gottschalk, C.M. (1962). Science and Information Theory. *Physics Today*. 15 (11). pp. 68–68.

Brown, C.A. (2005). Teaching axiomatic design to engineers—Theory, applications, and software. *Journal of Manufacturing Systems*. 24 (3). pp. 186–195.

Broy, M. & Rausch, A. (2005). Das neue V-Modell XT. *Informatik-Spektrum*. 28 (3). pp. 220–229.

Bullema, J.E., Stollman, G., Nederland, B. & Bekkers, P. (1999). *Default Design for Processability aanpak*. P. C. G. CFT (ed.). Eindhoven: Philips Centre of Fabrication Technology.

Bullinger, H.J. & Warschat, J. (2012). *Concurrent Simultaneous Engineering Systems*. Berlin/Heidelberg, Germany: Springer Science & Business Media.

Bundesministerium des Innern (2009). *Das V-Modell XT*. 1st Ed. [Online]. 27 January 2009. Bundesministerium des Innern (BMI). Available from: http://www.cio.bund.de/Web/DE/Architekturen-und-Standards/V-Modell-XT/vmodell_xt_node.html. [Accessed: 10 February 2014].

Burke, R. (2013). *Project management: planning and control techniques*. Hoboken, N.J.: Wiley.

Buseif, I. & Elfeituri, F. (2006). Using IDEF0/SADT Model for Design Flexible Manufacturing Systems (FMS's) Prototype. *Asia Pacific Industrial Engineering and Management Systems Conference*. (7th). pp. 80–91.

Ceglarek, D. (2014). *Eco and Resilient Factories*. In: I. P. O. T. C. O. T. E. Union (ed.). [Online]. 1 October 2014, Bologna. Available from: http://www.lets2014.eu/about-lets/what-is-lets/, Accessed 11/5th 2014.

Chen, X., Gao, S., Yang, Y. & Zhang, S. (2012). Multi-level assembly model for top-down design of mechanical products. *Computer-Aided Design*. 44 (10). pp. 1033–1048.

Chokshi, N.N., Duncan, C. & Mcfarlane, D.C. (2008). *A distributed coordination approach to reconfigurable process control*. B. Grabot, A. Mayère, & I. Bazet (eds.). Berlin/Heidelberg, Germany: Springer Verlag.

Christie, J. (2008). The Seductive and Dangerous V-Model J. M. Diaz Delgado (ed.). *Testing Experience*. (4). pp. 73–77.

Clark, J.O. (2009). System of Systems Engineering and Family of Systems Engineering from a standards, V-Model, and Dual-V Model perspective. *Systems Conference, 2009 3rd Annual IEEE*. pp. 381–387.

Daft, R.L. & Lengel, R.H. (1986). Organizational Information Requirements, Media Richness and Structural Design. *Management science*. 32 (5). pp. 554–571.

Dashchenko, A.I. (2006). *Reconfigurable manufacturing systems and transformable factories*. A. I. Dashchenko (ed.). Berlin, Heidelberg: Springer Verlag.

Deming, W.E. (2000). *Out of the Crisis*. Cambridge, Massachusetts: MIT Press.

Deng, Y.M., Deng, Y.M., Britton, G.A., Tor, S.B. & Tor, S.B. (2000). Constraint-based functional design verification for conceptual design. *Computer-Aided Design*. 32 (14). pp. 889–899.

Deuter, A. (2013). Slicing the V-Model -- Reduced Effort, Higher Flexibility. *2013 IEEE 8th International Conference on Global Software Engineering (ICGSE)*. pp. 1–10.

Diem, B., Rey, P., Renard, S., Viollet Bosson, S., Bono, H., Michel, F., Delaye, M.T. & Delapierre, G. (1995). SOI 'SIMOX'; from bulk to surface micromachining, a new age for silicon sensors and actuators. *Sensors and Actuators A: Physical*. 46 (1-3). pp. 8–16.

Dimov, S., Brousseau, E., Minev, R. & Bigot, S. (2012). Micro- and nano-manufacturing: Challenges and opportunities. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*. 226 (1). pp. 3–15.

Do, S.-H. (2015). *Independence Measure (IM) Using Independence Priority Number (IPN) in Axiomatic Design Framework*. In: 13 November 2015, Houston: ASME, pp. 241–247.

Do, S.-H. & Suh, N.P. (2000). *Object-oriented software design with axiomatic design*. In: 21 June 2000, pp. 278–284.

Do, S.-H. & Suh, N.P. (1999). Systematic OO programming with axiomatic design. *Computer*. 32 (10). pp. 121–124.

ECSFP (2009). *Towards Zero-Power ICT, European Committee Seventh Framework Programme; Wp 2009-2010 -Call 5 Objective ICT-2009.8.6*. Brussels: European Committee.

Efthymiou, K., Pagoropoulos, A., Papakostas, N., Mourtzis, D. & Chryssolouris, G. (2012). Manufacturing Systems Complexity Review: Challenges and Outlook. *Procedia CIRP*. 3 (3-2012). pp. 644–649.

EI-Haik, B. (2005). *Axiomatic Quality*. Hoboken, N.J.: Wiley-Blackwell.

El-Haik, B. (2011). *The Employment of Axiomatic Design in Software Engineering: A Software Development Conceptual Framework*. In: 11 December 2011, pp. 96–107.

El-Haik, B. & Yang, K. (1999). The components of complexity in engineering design. *IIE Transactions*. 31 (10). pp. 925–934.

ElMaraghy, H.A. (2006). Flexible and reconfigurable manufacturing systems paradigms. *International journal of flexible manufacturing systems*. 17 (4). pp. 261–276.

ElMaraghy, W.H., ElMaraghy, H.A. & Tomiyama, T. (2012). Complexity in engineering design and manufacturing. *CIRP Annals - Manufacturing Technology*. pp. 793–814.

Ertas, A. & Jones, J.C. (1996). *The engineering design process*. 1st Ed. Hoboken, N.J.: Wiley.

Fagas, G. (2008). *Semiconducting Nanowire Platform for Autonomous Sensors*. Brussels: European Union, Framework 7.

Fagas, G., Nolan, M., Georgiev, Y.M., Yu, R., Lotty, O., Petkov, N., Holmes, J.D., Jia, G., Eisenhawer, B., Gawlik, A., Falk, F., Khosropour, N., Buitrago, E., Badia, M.F.-B., Krummenacher, F., Ionescu, A.M., Kayal, M., Nightingale, A.M., de Mello, J.C., Puik, E.C.N., van der Bent, F., Lafeber, R., Ramaneti, R., Tong, H.D. & van Rijn, C. (2014). Component design and testing for a miniaturised autonomous sensor based on a nanowire materials platform. *Microsystem Technologies*. 20 (4-5). pp. 971–988.

Farid, A.M. (2008). Facilitating ease of system reconfiguration through measures of manufacturing modularity. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*. 222 (10). pp. 1275–1288.

Farid, A.M. (2014). Measures of reconfigurability and its key characteristics in intelligent manufacturing systems. *Journal of Intelligent Manufacturing*. 28 (2). pp. 353–369.

Farid, A.M. & McFarlane, D.C. (2006). An Approach to the Application of the Design Structure Matrix for Assessing Reconfigurability of Distributed Manufacturing Systems. *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06)*. pp. 121–126.

Farid, A.M. & McFarlane, D.C. (2008). Production degrees of freedom as manufacturing system reconfiguration potential measures. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*. 222 (10). pp. 1301–1314.

Farid, A.M. & Mcfarlane, D.C. (2006). A tool for assessing reconfigurability of distributed manufacturing systems. In: *Information Control Problems in Manufacturing 2006*. Elsevier, pp. 499–504.

Fiege, R. & Stelzer, D. (2007). *Modellierung Serviceorientierter Architekturen mit Axiomatic Design – Analyse des Beitrages zur Verbesserung der Entwurfsqualität*. H. U. Bankhofer, V. Nissen, D. Stelzer, & S. Straßburger (eds.). Ilmenau: Technische Universität Ilmenau. pp. 1-54.

Foley, J.T., Puik, E.C.N. & Cochran, D.S. (2016). *Desirable Complexity*. In: A. Liu (ed.). 21 September 2016, Xi'an, p. 6.

Foley, J.T., Puik, E.C.N. & Cochran, D.S. (2017). The Faucet Reloaded: Improving Axiomatic Design by Example L. Slatineanu (ed.). *MATEC Web of Conferences*. 127. p. 01009.

Friedrich, J., Hammerschall, U., Kuhrmann, M. & Sihling, M. (2009). *Das V-Modell XT*. Berlin/Heidelberg, Germany: Springer.

Garbie, I.H. (2014). A methodology for the reconfiguration process in manufacturing systems. *Journal of Manufacturing Technology Management*. 25 (6). pp. 891–915.

Garvey, P. & Landsdowne, Z.F. (1998). Risk Matrix: An Approach for Identifying, Assessing, and Ranking Program Risks. *Air Force Journal of Logistics*. pp. 16–19 & 29.

Gell-Mann, M. & Lloyd, S. (1996). Information measures, effective complexity, and total information. *Complexity*. 2 (1). pp. 44–52.

Gielen, P., Sillen, R. & Puik, E.C.N. (2012). *Low cost environmentally friendly ultrasonic embossed electronic circuit board*. In: 2012, IEEE, pp. 1–7.

Gonçalves-Coelho, A.M., Mourão, A.J.F. & Pereira, Z.L. (2005). Improving the use of QFD with Axiomatic Design. *Concurrent Engineering*. 13 (3). pp. 233–239.

Gould, J.D. & Lewis, C. (1985). Designing for usability: key principles and what designers think. *Communications of the ACM*. 28 (3). pp. 300–311.

Grosser, V., Reichl, H., Kergel, H. & Schuenemann, M. (2000). A Fabrication Framework for Modular Microsystems. *MST News, January 2005*. 01. pp. 4–8.

Gunasekaran, A. (2001). *Agile manufacturing*. Amsterdam, the Netherlands: Elsevier Science.

Gutierrez, A. (1999). MEMS/MST fabrication technology based on microbricks P. Salomon (ed.). *MSTNEWS; International Newsletter on Microsystems & MEMS*. 99 (1). pp. 4–8.

Haitjema, H. & van Seggelen, J.K. (2003). Metrology for MEMS; MEMS for Metrology. *MEMSTAND: Standardisation for Microsystems: the Way Forward*. pp. 218–227.

Hansen, F. (1966). *Konstruktionssystematik*. Berlin: Verl. Technik.

Hartley, R.V.L. (1928). Transmission of information. *Bell System Technical Journal, The*. 7 (3). pp. 535–563.

Hasan, F., Jain, P.K. & Kumar, D. (2013). Machine Reconfigurability Models Using Multi-Attribute Utility Theory and Power Function Approximation. *Procedia Engineering*. 64. pp. 1354–1363.

Haskins, C. (2006). *Incose Systems Engineering Handbook V3*. C. Haskins (ed.). San Diego, California, U.S.: INCOSE.

Heeren, H., Salomon, P. & El-Fatratry, A. (2004). Worldwide Services and Infrastructure for Microproduction. *MSTNEWS; International Newsletter on Microsystems & MEMS*. 2. pp. 8–9.

Heinig, A., Dietrich, M., Herkersdorf, A., Miller, F., Wild, T., Hahn, K., Grünewald, A., Brück, R., Kröhnert, S. & Reisinger, J. (2014). *System integration - the bridge between More than Moore and More Moore*. In: 24 March 2014, Dresden, Germany: European Design and Automation Association, pp. 132–140.

Herzlich, P. (1993). *The Politics of Testing*. In: 25 October 1993, London: First Eurostar Conference (European Software Testing Analysis & Review), pp. PH1–PH6.

Highsmith, J. & Cockburn, A. (2001). Agile software development: the business of innovation. *Computer*. 34 (9). pp. 120–127.

Hollstein, P., Lasi, H. & Kemper, H.-G. (2012). A survey on changeability of machine tools. *4th International Conference on Changeable, Agile, Reconfigurable and Virtual Production CARV, Montreal, Canada*. (Chapter 15). pp. 92–98.

Höhn, R., Höppner, S., Broy, M., Rausch, A., Petrasch, R., Biffl, S., Wagner, R., Hesse, W. & Bergner, K. (2008). *Das V-Modell XT*. 1st Ed. Berlin/Heidelberg: Springer.

Hsu, T.-R. (2004). *MEMS packaging*. London: Peter Peregrinus Ltd.

Hsu, W. & Liu, B. (2000). Conceptual design: issues and challenges. *Computer-Aided Design*. 32 (14). pp. 849–850.

Hua, Z., Zhu, X., Wang, H. & Koren, Y. (2006). Integration TRIZ with problem solving tools: A literature review from 1996 to 2006. *International Journal of Business Innovation Research*. 2006 (1). pp. 111–128.

Hubka, V. (1973). *Theorie der Maschinensysteme*. Berlin/Heidelberg: Springer-Verlag.

IMTR-Project-Team (2000). *Integrated Manufacturing Technology Roadmapping Project*. The IMTR M&S Workshop GroupThe IMTR Roadmapping Project Team (eds.). Oak Ridge, Tennessee: Integrated Manufacturing Technology Initiative, Inc.

Kear (1992). *Hybrid Assemblies and Multichip Modules*. Boca Raton, Florida: CRC Press.

Khosro Pour, N., Krummenacher, F. & Kayal, M. (2012). A miniaturized autonomous microsystem for hydrogen gas sensing applications. *2012 IEEE 10th International New Circuits and Systems Conference (NEWCAS)*. pp. 201–204.

Khosro Pour, N., Krummenacher, F. & Kayal, M. (2013). Fully Integrated Solar Energy Harvester and Sensor Interface Circuits for Energy-Efficient Wireless Sensing Applications. *ournal of Low Power Electronics and Applications*. 3 (1). pp. 9–24.

Kim, S.-J. & Suh, N.P. (1991). Design of software system based on axiomatic design. *CIRP Annals - Manufacturing Technology*. 40 (1). pp. 165–170.

Koenig, A. (1998). Koenig: Patterns and antipatterns. In: P. Kruchten (ed.). *The view model of architecture*. The patterns handbook: techniques, pp. 42–50.

Komoto, H. & Tomiyama, T. (2012). A framework for computer-aided conceptual design and its application to system architecting of mechatronics products. *Computer-Aided Design*. 44 (10). pp. 931–946.

Koren, Y. (2006). General RMS Characteristics. Comparison with Dedicated and Flexible Systems. In: A. I. Dashchenko (ed.). *Reconfigurable Manufacturing Systems and Transformable Factories*. Berlin/Heidelberg, Germany: Springer Berlin Heidelberg, pp. 27–45.

Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G. & Van Brussel, H. (1999). Reconfigurable Manufacturing Systems. *CIRP Annals - Manufacturing Technology*. 48 (2). pp. 527–540.

Kroll, E. (2012). Design theory and conceptual design: contrasting functional decomposition and morphology with parameter analysis. *Research in Engineering Design*. 24 (2). pp. 165–183.

Kroll, P. & Kruchten, P. (2003). *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*. Reading, MA: Addison Wesley.

Kulak, O., Durmusoglu, M.B. & Tufekci, S. (2005). *A complete cellular manufacturing system design methodology based on axiomatic design principles*. 48 (4). pp. 765–787.

Kumar, V. (2003). *Innovation Planning*. In: 17 October 2003, Chicago, pp. 1–41.

Kurtz, C.F., Snowden, D., Kurtz, C.F. & Snowden, D.J. (2003). The new dynamics of strategy: Sense-making in a complex and complicated world. *IBM Systems Journal*. 42 (3). pp. 462–483.

Kuzgunkaya, O. & ElMaraghy, H.A. (2009). Economic and Strategic Justification of Changeable, Reconfigurable and Flexible Manufacturing. In: W. H. ElMaraghy (ed.). *Changeable and Reconfigurable Manufacturing Systems*. London: Springer London, pp. 303–320.

Lafeber, R., Puik, E.C.N. & Sillen, R. (2014). Production of Smart Dust, HU University of Applied Sciences. *Graduation Report, Master of Engineering - Integrated Product Development*. pp. 1–68.

Lang, W. (1999). Reflexions on the future of microsystems. *Sensors and Actuators A: Physical*. 72 (1). pp. 1–15.

Lau, K.-K., Taweel, F.M. & Tran, C.M. (2011). *The W Model for Component-Based Software Development*. In: 2011, IEEE, pp. 47–50.

Li, W., Li, Y., Wang, J. & Liu, X. (2010). The process model to aid innovation of products conceptual design. *Expert Systems with Applications*. 37 (5). pp. 3574–3587.

Liversidge, E. (2005). *The Death of the V-Model; Harmonic Software Systems*. Billingshurst, West Sussex, UK: Harmonic Software Systems.

Lotfi, V. (1995). Implementing flexible automation: A multiple criteria decision making approach. *International Journal of Production Economics*. 38 (2-3). pp. 255–268.

Madou, M.J. (2002). *Fundamentals of Microfabrication*. Boca Raton, Florida: CRC Press.

Madou, M.J. (2011). *Manufacturing Techniques for Microfabrication and Nanotechnology*. Boca Raton, Florida: CRC Press.

Maropoulos, P.G. & Ceglarek, D. (2010). Design verification and validation in product lifecycle. *CIRP Annals - Manufacturing Technology*. 59 (2). pp. 740–759.

Mathur, S. & Malik, S. (2010). Advancements in the V-Model. *International Journal of Computer Applications*. 1 (12). pp. 29–34.

McHugh, M., Cawley, O., McCaffcry, F., Richardson, I. & Wang, X. (2013). An agile V-Model for medical device software development to overcome the challenges with plan-driven software development lifecycles. *Software Engineering in Health Care (SEHC), 2013 5th International Workshop on*. pp. 12–19.

Mesarovic, M.D. (1964). *Foundations for a general systems theory*. Cleveland, Ohio, U.S.: Systems Research Center, Case Institute of Technology.

Michaelis, M.T. & Johannesson, H. (2012). From Dedicated to Platform-Based Co-Development of Products and Manufacturing Systems. *4th International Conference on Changeable, Agile, Reconfigurable and Virtual Production CARV, Montreal, Canada*. (Chapter 32). pp. 196–202.

Moergestel, L., van, Puik, E.C.N., Telgen, D. & Meyer, J.-J. (2011). Decentralized Autonomous-Agent-Based Infrastructure for Agile Multiparallel Manufacturing. *2011 Tenth International Symposium on Autonomous Decentralized Systems (ISADS)*. pp. 281–288.

Moergestel, L., van, Telgen, D., Puik, E.C.N. & Meyer, J.-J. (2013). A Versatile Agile Agent-Based Infrastructure for Hybrid Production Environments B. Natalia (ed.). *Manufacturing Modelling, Management, and Control*. 7 (1). pp. 210–215.

Moore, G.A. (2009). *Inside the Tornado*. New York City, U.S.: Harper Collins.

Moore, G.E. (2005). Cramming more components onto integrated circuits, Electronics, 38: 8 (1965). *Proceedings of the IEEE*. pp. 82–85.

Mooz, H. & Forsberg, K. (2001). *Visual Explanation of Development Methods and Strategies including the Waterfall, Spiral, Vee, Vee+, and Vee++ Models*. In: 1 July 2001, pp. 610–617.

Morton, S. (2001). The Butterfly Model for Test Development. *Applied Dynamics International*. pp. 1–14.

Nassehi, A., Newman, S., Dhokia, V., Zhu, Z. & Asrai, R.I. (2012). Using formal methods to model hybrid manufacturing processes. *4th International Conference on Changeable, Agile, Reconfigurable and Virtual Production CARV, Montreal, Canada*. (Chapter 8). pp. 52–56.

Offermans, P., Tong, H.D., van Rijn, C.J.M., Merken, P., Brongersma, S.H. & Crego-Calama, M. (2009). Ultralow-power hydrogen sensing with single palladium nanowires. *Applied Physics Letters*. 94 (22). pp. 223110–1–223110–3.

Office of Government Commerce, O.O.G. (2009). *Managing successful projects with PRINCE2*. Norwich, U.K.: The Stationery Office, TSO.

Onori, M. (2009). *Outlook Report on the Future of European Assembly Automation*. Brussels: European Committee.

Pahl, G. & Beitz, W. (2013). *Engineering Design*. K. Wallace (ed.). London: Springer Science & Business Media.

Porter, M.E. (1985). *Competitive advantage: creating and sustaining superior performance*. New York: Free Press.

Pries, K. & Quigley, J. (2010). Scrum and the Waterfall Method. In: *Scrum Project Management*. CRC Press, pp. 95–106.

Puik, E.C.N. (2009). *Keynote: Flexibele Productie in het Micro Domein*. In: E. C. N. Puik (ed.). 11 June 2009, Eindhoven, pp. kn1–kn39.

Puik, E.C.N. & Ceglarek, D. (2016). *A Different Consideration on Information and Complexity in Axiomatic Design*. In: N. P. Suh & A. M. Farid (eds.). pp. 105–129.

Puik, E.C.N. & Ceglarek, D. (2014a). *A Review on Information in Design*. In: M. K. Thompson (ed.). 24 September 2014a, Lisbon, pp. 59–64.

Puik, E.C.N. & Ceglarek, D. (2014b). *A Theory of Maturity*. In: M. K. Thompson (ed.). 24 September 2014b, Lisbon, pp. 115–120.

Puik, E.C.N. & Ceglarek, D. (2015a). *Axiomatic Product Design in Three Stages: A Constituent Roadmap That Visualises the Status of the Design Process by Tracking the Knowledge of the Designer*. In: 13 November 2015a, Houston: ASME, pp. 249–257.

Puik, E.C.N. & Ceglarek, D. (2015b). The Quality of a Design Will Not Exceed the Knowledge of its Designer; an Analysis Based on Axiomatic Information and the Cynefin Framework. *9th International Conference on Axiomatic Design (ICAD)*. 34. pp. 19–24.

Puik, E.C.N. & Moergestel, L., van (2010). *Agile Multi-Parallel Micro Manufacturing using a Grid of Equiplets*. In: S. Ratchev (ed.). 2010, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 271–282.

Puik, E.C.N., Foley, J.T. & Ceglarek, D. (2016). *Ignorance is Bliss: sudden appearance of previously unrecognized information and its effect*. In: A. Liu (ed.). 21 September 2016, Xi'an, p. 8.

Puik, E.C.N., Gielen, P., Telgen, D., Moergestel, L., van & Ceglarek, D. (2014a). A Generic Systems Engineering Method for Concurrent Development of Products and Manufacturing Equipment. *International Precision Assembly Seminar IPAS*. 435 (Chapter 18). pp. 139–146.

Puik, E.C.N., Rijfers, A. & Tillie, L. (2002). A Lower Threshold to the World by Combining a Modular Design Approach and a Matching Production Framework Mancef (ed.). *Commercialization of Micro Systems COMS*. 2002. pp. 1–4.

Puik, E.C.N., Smulders, E., Gerritsen, J., Huijgevoort, B.V. & Ceglarek, D. (2013a). A Method for Indexing Axiomatic Independence Applied to Reconfigurable Manufacturing Systems M. K. Thompson (ed.). *7th International Conference on Axiomatic Design*. 7. pp. 186–194.

Puik, E.C.N., Telgen, D., Ceglarek, D. & Moergestel, L., van (2013b). Qualitative product/process modelling for reconfigurable manufacturing systems. *2013 IEEE International Symposium on Assembly and Manufacturing (ISAM)*. pp. 214–218.

Puik, E.C.N., Telgen, D., Moergestel, L., van & Ceglarek, D. (2017a). Assessment of reconfiguration schemes for Reconfigurable Manufacturing Systems based on resources and lead time. *Robotics and Computer-Integrated Manufacturing*. 43 (C). pp. 30–38.

Puik, E.C.N., Telgen, D., Moergestel, L., van & Ceglarek, D. (2014b). *Classification of reconfiguration resources and lead time for reconfigurable manufacturing systems*. In: 2014b, San Antonio: DEStech Publications, Inc., pp. 857–862.

Puik, E.C.N., Telgen, D., Moergestel, L., van & Ceglarek, D. (2013c). *Structured Analysis of Reconfigurable Manufacturing Systems*. (Chapter 12). pp. 147–157.

Puik, E.C.N., van Duijn, J. & Ceglarek, D. (2017b). Guidelines for Application of the Constituent Roadmap of Product Design Based on Axiomatic Design L. Slatineanu (ed.). *MATEC Web of Conferences*. 127. p. 01013.

Regazzoni, D. & Russo, D. (2011). TRIZ tools to enhance risk management. *Procedia Engineering*. 9. pp. 40–51.

Rook, P. (1986). Controlling software projects. *Software Engineering Journal*. 1986 (1). pp. 7–16.

Roozenburg, N. & Eekels, J. (1995). *Product design: fundamentals and methods*. Hoboken, N.J.: Wiley.

Ross, D.T. (1977). Structured Analysis (SA): A Language for Communicating Ideas. *Software Engineering, IEEE Transactions on Software Engineering*. (1). pp. 16–34.

Royce, W.W. (1970). Managing the development of large software systems. *proceedings of IEEE WESCON*. pp. 328–338.

Saldaña, J. (2012). *The Coding Manual for Qualitative Researchers*. Thousand Oaks, California: SAGE.

Sanders, E.B.N. & Stappers, P.J. (2008). Co-creation and the new landscapes of design. *Co-Design*. 4 (1). pp. 5–18.

Schwaber, K. (1997). *SCRUM Development Process*. (Chapter 11). pp. 117–134.

Scott, G. (2001). Strategic Planning for High-Tech Product Development. *Technology Analysis & Strategic Management*. 13 (3). pp. 343–364.

Semere, D., Onori, M., Maffei, A. & Adamietz, R. (2013). Evolvable assembly systems: coping with variations through evolution M. Onori (ed.). *Assembly Automation*. 28 (2). pp. 126–133.

Shannon, C.E. (1948). A Mathematical Theory of Communication. *The Bell System Technical Journal*. 27. pp. 397–423.

Shannon, C.E. & Weaver, W. (1949). *The Mathematical Theory of Communication*. Chicago: The University of Illinois Press.

Sheffield, J. (2005). Systemic Knowledge and the V-Model. *Int. J. Business Information Systems*. 1 (2). pp. 83–101.

Sheffield, J., Sankaran, S. & Haslett, T. (2013). Systems thinking: taming complexity in project management. *On the Horizon*. 20 (2). pp. 126–136.

Shewhart, W.A. (1939). *Statistical Method from the Viewpoint of Quality Control*. Mineola, New York: Dover Publications.

Simon, H.A. (1996). *The Sciences of the Artificial*. Cambridge, Massachusetts: MIT Press.

Snowden, D.J. & Boone, M.E. (2007). Snowden: A leader's framework for decision making. *Harvard Business Review*. 85 (11). pp. 68–76.

Sohlenius, G. (1992). Concurrent Engineering. *CIRP Annals - Manufacturing Technology*. 41 (2). pp. 645–655.

Spicer, P., Yip-Hoi, D. & Koren, Y. (2007). Scalable reconfigurable equipment design principles. *International journal of production research*. 43 (22). pp. 4839–4852.

Stock, A. & Schünemann, M. (2002). *Kunststofftechnologie - eine weitere MatchX - Bausteintechnologie*. 16 Ed. F. Gesellschaft (ed.). Stuttgart, Germany: Fraunhofer Institut für Produktionstechnologie und Automatisierung.

Strasser, T., Fessl, K. & Hämmerle, A. (2005). *Rapid reconfiguration of machine-tools for holonic manufacturing systems*. In: 2005, pp. 1–6.

Suh, N.P. (1999). A Theory of Complexity, Periodicity and the Design Axioms. *Research in Engineering Design*. 11 (2). pp. 116–132.

Suh, N.P. (2001). *Axiomatic Design - Advances and Applications*. In: Cambridge, MA: Oxford University Press, New York.Oxford, 2001, Chapter 5, pp. 239-298.

Suh, N.P. (2004). *Axiomatic Design Framework for Large-Scale Complex Systems Design*. Boston: MIT.

Suh, N.P. (2005a). Complexity in Engineering. *CIRP Annals - Manufacturing Technology*. 54 (2). pp. 46–63.

Suh, N.P. (2005b). *Complexity: Theory and Applications*. Oxford, England, U.K.: Oxford University Press.

Suh, N.P. (1990). *The Principles of Design (Oxford Series on Advanced Manufacturing)*. Oxford, England, U.K.: Oxford University Press on Demand.

Suh, N.P., Bell, A.C. & Gossard, D. (1978). On an axiomatic approach to manufacturing and manufacturing systems. *Journal of Engineering for Industry*. 100. pp. 127–130.

Taguchi, G., Chowdhury, S. & Wu, Y. (2005). *Taguchi's Quality Engineering Handbook*. Hoboken, N.J.: Wiley-Interscience.

Tay, F.E.H. & Gu, J. (2002). Product modeling for conceptual design support. *Computers in Industry*. 48 (2). pp. 143–155.

Telgen, D., Puik, E.C.N., Moergestel, L., van, Bakker, T. & Meyer, J.-J. (2015). Reconfigurable Equiplets Operating System; A Hybrid Architecture to Combine Flexibility and Performance for Manufacturing. *International Journal on Advances in Software*. 8 (3 & 4). pp. 309–326.

Teoh, P.C. & Case, K. (2004). Failure modes and effects analysis through knowledge modelling. *Journal of Materials Processing Technology*. 153-154. pp. 253–260.

Tichem, M. & Tanase, D. (2008). A DFA Framework for Hybrid Microsystems. In: *Micro-Assembly Technologies and Applications*. IFIP — International Federation for Information Processing. Boston, MA: Springer US, pp. 13–22.

Tong, D.H., Tran, P.D., Pham, X.T.T., Van Binh Pham, Le, T.T.T., Dang, M.C. & Van Rijn, C.J.M. (2010). The nanofabrication of Pt nanowire arrays at the wafer-scale and its application in glucose detection. *Advances in Natural Sciences: Nanoscience and Nanotechnology*. 1 (1). pp. 015011–1–015011–4.

Toonder, M. (1972). *Het Verdwijnpunt*. Bezige Bij.

Triki, A.H., Hachicha, W. & Masmoudi, F. (2011). *A conceptual design framework of production line systems based on axiomatic design: case of assemble line*. In: 21 March 2011, Tunis, pp. 1–8.

Ulrich, K.T. & Eppinger, S.D. (2004). *Ulrich: Product design and development*. Boston: McGraw-Hill.

Van Brussel, H., Peirs, J., Reynaerts, D., Delchambre, A., Reinhart, G., Roth, N., Weck, M. & Zussman, E. (2000). Assembly of Microsystems. *CIRP Annals - Manufacturing Technology*. 49 (2). pp. 451–472.

van der Bent, J.F., Puik, E.C.N., Tong, H.D. & van Rijn, C.J.M. (2015). Temperature balanced hydrogen sensor system with coupled palladium nanowires. *Sensors and Actuators A: Physical*. 226. pp. 98–106.

van der Bent, J.F., van Rijn, C.J.M. & Puik, E.C.N. (2010). Ultra low power temperature compensation method for palladium nanowire grid. *Procedia Engineering*. 5. pp. 184–187.

Vigna, B. (2005). *More than Moore: micro-machined products enable new applications and open new markets*. In: 2005, Washington DC, U.S.: IEEE, pp. 1–8.

Voelkel, R., Gong, L., Rieck, J. & Zheng, A. (2012). Wafer-level micro-optics: trends in manufacturing, testing, packaging, and applications Y. Wang, C. Du, H. Hua, K.

Tatsuno, & H. P. Urbach (eds.). *Photonics Asia*. 8557. pp. 855702–855702–12.

Wang, L., Shen, W., Xie, H., Xie, H., Neelamkavil, J., Neelamkavil, J., Pardasani, A. & Pardasani, A. (2002). Collaborative conceptual design—state of the art and future trends. *Computer-Aided Design*. 34 (13). pp. 981–996.

Wanhill, R., Molent, L. & Barter, S. (2016). *Milestone Case Histories in Aircraft Structural Integrity*. Amsterdam, the Netherlands: National Aerospace Laboratory NLR.

Werdich, M. (2011). *FMEA – Einführung und Moderation*. M. Werdich (ed.). Wiesbaden: Vieweg+Teubner.

Wiendahl, H.-P., ElMaraghy, H.A., Nyhuis, P., Zäh, M.F., Wiendahl, H.H., Duffie, N. & Brieke, M. (2007). Changeable Manufacturing - Classification, Design and Operation. *CIRP Annals - Manufacturing Technology*. 56 (2). pp. 783–809.

Wildemann, H. (2013). *Komplexitatsmanagement in Vertrieb, Beschaffung, Produkt, Entwicklung und Produktion: Leitfaden zur Einfuhrung eines durchgangigen Komplexitatsmanagements*. München, Germany: TCW Transfer-Centrum für Produktions-Logistik und Technologie-Management GmbH & Co. KG.

Withey, P.A. (1997). Fatigue failure of the de Havilland comet I. *Engineering failure analysis*. 4 (2). pp. 147–154.

Yan, P., Zhou, M. & Caudill, R. (2000). A life cycle engineering approach to FMS development. *Robotics and Automation*. 1. pp. 395–400.

Yang, G. & Gaines, J. (2001). *A flexible experimental workcell for efficient and reliable wafer-level 3D micro-assembly*. In: 2001, Seoul, pp. 133–138.

Yang, K. & El-Haik, B. (2008). *Design for Six Sigma*. New York City, U.S.: McGraw Hill Professional.

Zhang, G., Liu, R., Gong, L. & Huang, Q. (2006a). An Analytical Comparison on Cost and Performance among DMS, AMS, FMS and RMS. *Reconfigurable manufacturing systems and transformable factories*. pp. 659–673.

Zhang, G.Q., van Roosmalen, F. & Graef, M. (2005). *The paradigm of 'more than Moore'*. In: 2005, IEEE, pp. 17–24.

Zhang, G.Q.K., Graef, M. & Van Roosmalen, F. (2006b). *Strategic Research Agenda of 'More than Moore'*. In: 2006b, IEEE, pp. 1–6.

Zhang, Z. & Chu, X. (2010). A new approach for conceptual design of product and maintenance. *International Journal of Computer Integrated Manufacturing*. 23 (7). pp. 603–618.

APPENDIX A
DESIGN & IMPLEMENTATION OF RECONFIGURABLE PRODUCTION EQUIPMENT FOR AN AUTOMOTIVE PIEZO ACTUATOR

This appendix provides supplementary data to Chapter 3. The case follows the design optimisation of an automotive actuator to enable efficient manufacturing of a reconfigurable production platform. The maturity of the product design was explicitly registered during the phases from early 'Manufacturing Development' via 'Pilot Production' to 'Manufacturing Ramp-Up'. Product development and manufacturing development took place in a concurrent fashion to find an optimal balance between product- and equipment complexity.

A.1 Definition of the Product

The product under investigation is a subsystem that makes part of a pneumatic switch for an automotive comfort system. It consists of a ceramic piezoelectric strip, one double and two single brass contact springs, as shown in Figure A.1.

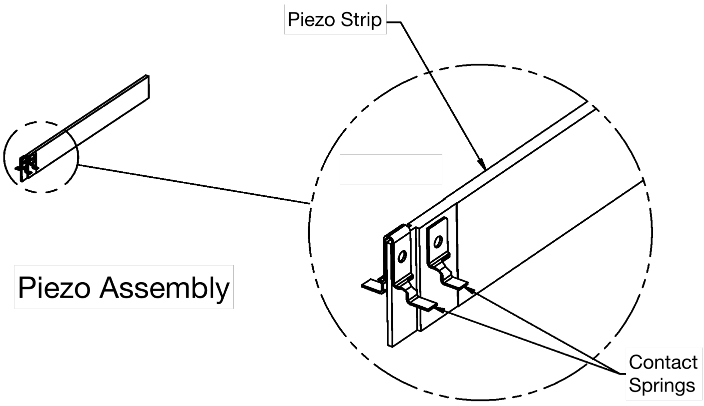


Figure A.1 *Piezo assembly for an automotive pneumatic switch*

The contact springs are to be mounted on the piezo actuator. Due to the complex shape and fragile nature of the product, a sequential production process is preferred. This allows higher system complexity and direct feedback control of the production process (Hsu, 2004). The connection between piezo and contacts should guarantee that the parts are electrically connected. At the position of the bond, the piezos are equipped with a conductive coating. The contacts are stamped from a coated metal strip that is supplied on a reel. The connection will be realised with an electrically conductive adhesive. At the initial phase of the project, the engineers think of applying a ‘Carbon Paste’ for the adhesive connection. Tests have indicated that the carbon paste will ensure an acceptable electrical resistance for this specific application. The carbon paste however needs curing at an increased temperature of 120-140 degrees centigrade; a process that needs several hours to complete. It is not acceptable to occupy the assembly tool that brings the parts together in the right position for hours per product during the curing phase. Therefore, a second adhesive, that can be cured considerably faster, is applied to keep the contact springs into place when the piezo assembly is taken out of the positioning tool. Adhesives that can be cured in time of seconds typically apply an external optical energy source e.g. IR or UV. In this case, an ultraviolet curing adhesive is selected. This will enable the assembly system to continue with the successive product shortly after the previous assembly action is completed. A cycle time of roughly 10 seconds is expected to be feasible, meeting the production requirement of 5000 products per day in a double shift of 16 hours. Curing the carbon paste is performed afterwards as a batch process. Given the small geometrical dimensions of the products, a single furnace is able to hold the daily production. During production hours, parts are buffered by the capacity of the furnace, enabling batch-curing overnight. The next day all parts will be available for testing.

A.2 Initial Setup of the Manufacturing System

The company has deployed a manufacturing strategy comprising reconfigurable production elements in a structure of lines, cells, modules and devices. This means that a number of frequently used processes and their hardware-toolsets have been standardised and documented. These *process modules* available for reuse without the need to develop them from scratch. In this case, a contamination controlled cabinet, the manipulators, and the dispensing modules for the adhesives could be used off-the-shelf. This makes the realisation of the envisioned production platform a matter of 'Configure to Order' rather than 'Engineer to Order'.

Yang and Nelson's approach for determination of system architecture of reconfigurable equipment will be applied (Yang & Gaines, 2001; Hsu, 2004): (i) determination of assembly process flow through the system, (ii) decomposition of system functions, and (iii) determination of the general system architecture.

A.2.1 Determination of Process Flow Through the System

A preliminary layout for the production machine was visualised using the SADT data diagram (Figure A.2). This layout was set as a starting point for further decomposition.

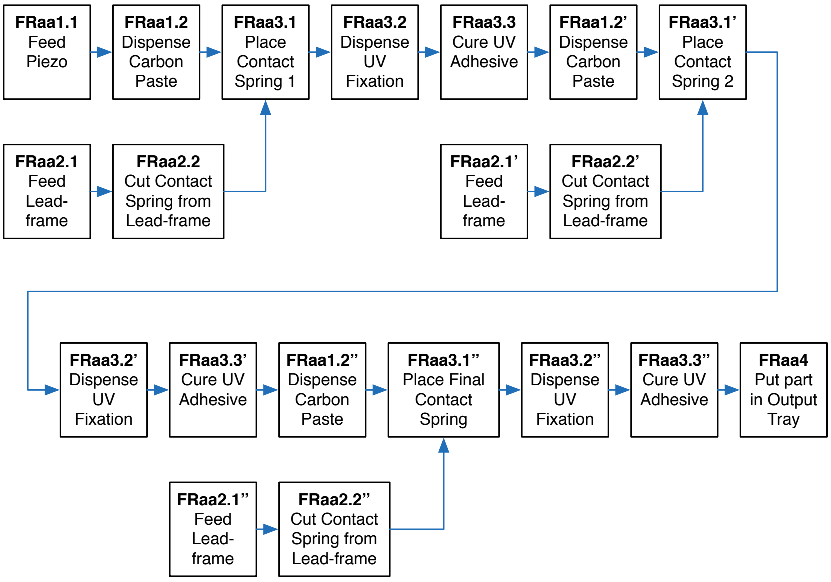


Figure A.2 *Initial assembly process flow*

A.2.2 Decomposition of System Functions

The assembly process flow was divided into five elementary steps: (i) system initialisation, (ii) part pick-up, (iii) move part to assembly position, (iv) align & insert, and (v) prepare for next assembly operation. After the initial inventorying action, a number of logistical starting points were defined:

- Contact springs are fed into the system using tape on reel. This is due to the fact that the manufacturing equipment of the springs defaults to this standard;
- The piezo strips, are supplied in a custom designed tray that was developed for the previous process (coating the piezo strips);
- An output tray, that uses a different number of parts than the input tray, has already been designed. The implicit fact that the different number of parts in input- and output-tray causes up to twice as many machine stops, to feed and remove parts, is accepted (input- and output trays will not be replaced simultaneously);

- The standardised robotic manipulator will be used for the handling of the parts through the machine. For the rest the machine should reuse as much pre-developed technology modules as possible;
- Based on this information, a team of experts was asked to define a most promising layout for the manufacturing of given parts. This layout should meet the five elementary actions for the process flow, based on the chosen logistical limitations.

Note that at this point only prototype parts were produced manually, also no investments in production hardware were made.

A.2.3 Determination of General System Architecture

Based on the assembly process flow and the decomposed system functions, a first analysis of the envisioned manufacturing system was made. It appeared after an initial consideration that a system to match the structure of Figure A.2 calls for large hardware investments. Due to the sequential approach, the system would be equipped with separate assembly stations for placement of respectively Contact Spring 1, Spring 2 and the Final Contact Spring. Each station would need a complete set of *process modules*, each consisting of at least a carbon paste dispenser, a placement tool for each contact spring, a dispenser for UV curing adhesive, the light source for UV curing, and the feeder systems.

A general guideline in machine building is to keep all process components (dispensers, feeders, manipulators) as occupied as possible (Boothroyd, 2005; Dashchenko, 2006). However, this is not the case in the initial manufacturing layout. The assembly stations are used only used once every cycle. This leaves constantly two of the three assembly stations inactive, causing a bad occupational ratio (being defined as the active time of the module divided by the cycle time). To reduce the investments in manufacturing hardware, the occupational ratio should be improved.

A solution was found by grouping the placement actions of all three contact springs. The three assembly sequences, for the separate contact springs, are combined in a single parallel placement action instead of sequential per contact. Additionally, all adhesive dots of the carbon paste may be dispensed with one and the same dispensing system. The product design must be optimised by applying ‘Design for Assembly’ (DfA) to enable parallel assembly. In analogy with proven semiconductor back-end industry, the piezo actuator was optimised by leaving the contacts springs attached to another till the assembly of the contact springs has been completed. The contact springs may still be supplied using a lead-frame on a reel. In this case the springs will be cut from the mother tape in two stages. Initially, an assembly of three springs is cut from the tape in such way that the springs remain attached by the metal of the tape. This ’contact spring assembly’ is positioned and UV-fixated as a whole. This reduces three separate assembly cycles to a single combined process. A final cutting action separates the contacts after curing the UV-adhesive. Batch curing of the carbon paste is still performed afterwards. Its integrity is relying on the UV-fixation until thermal curing has completed. The optimised product design is shown in Figure A.3.

Optimisation of the product design has considerable impact on the process flow through the production system. This is shown in the updated SADT data representation of Figure A.4. The reduction of the number of process steps indicates a lower demand for logistical handling actions though the manufacturing system and thus increases efficiency.

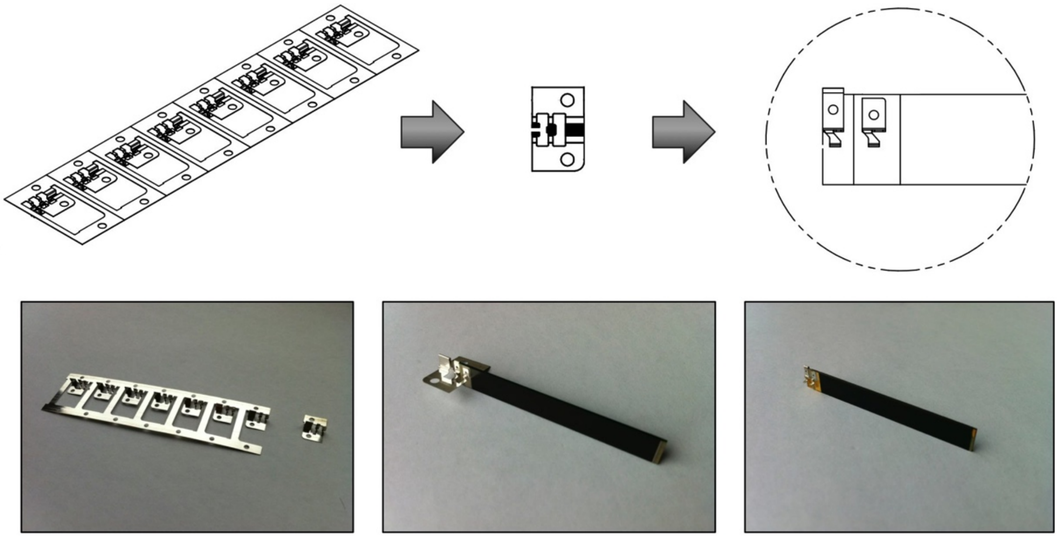


Figure A.3 Optimised assembly sequence leaves the contact springs attached

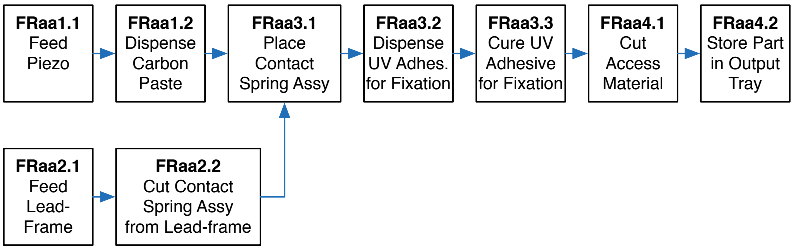


Figure A.4 SADT data diagram for the optimised assembly process flow

A.2.4 Further Decomposition of the Proposed System

The next phase is to bring all modular parts of the machine to a comparable and adequate operational standard. This should provide certainty that little or no problems will occur when used for industrial manufacturing. The first step to achieve this is to further examine the modular parts of the machine using the SADT parameter analysis. Not only new modules to the reconfigurable concept are under investigation; all modules will be examined in their (new) context. A team of experts from both product design and production engineering perform the examination in face-to-face meetings. Out of a total of nine processes, a number of three processes are emphasised in this appendix as an example. In Figure A.5, process FRaa1.2 ‘Dispense Carbon Paste’ is shown.

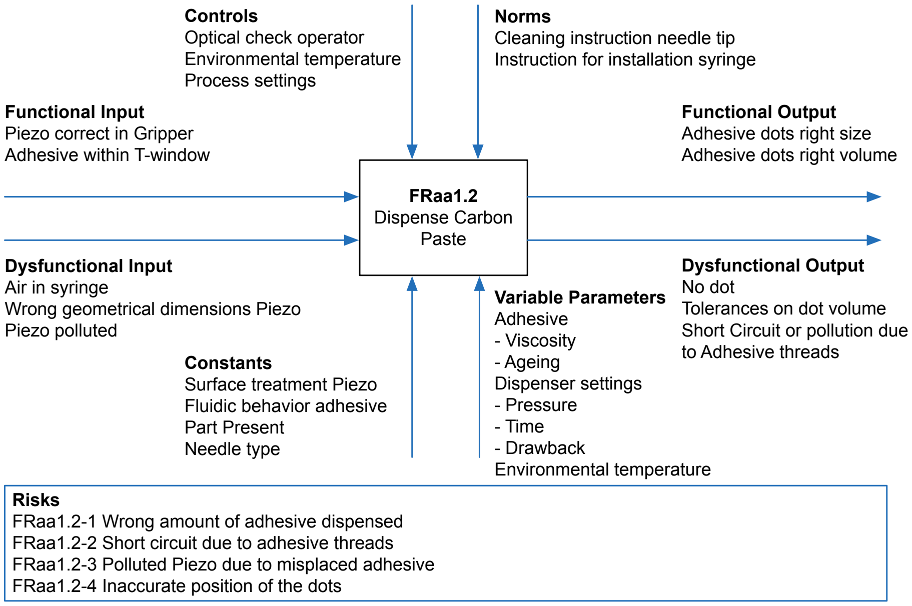


Figure A.5 *SADT parameter analysis of the dispensing process of the carbon paste dots*

The flow is divided in a ‘Functional’ part (what must go right) and a ‘Dysfunctional’ part (what can go wrong). ‘Controls’ and ‘Norms’ can lower the severity of the whims of the system since it enables monitoring of the process. Appointing ‘Constants’ and ‘Parameters’ helps determination of the sensitivities of the process. It contributes to a higher level of objectivity in the expert group process, since the disturbing factors are named and their influences can be estimated. The remaining risks, which are to be considered as basic hazards, are inventoried at the end and are identified with a number for practical follow-up.

For two other processes, placing the contact spring assemblies and cutting the access material, the SADT parameter description is given in Figure A.6 respectively Figure A.7.

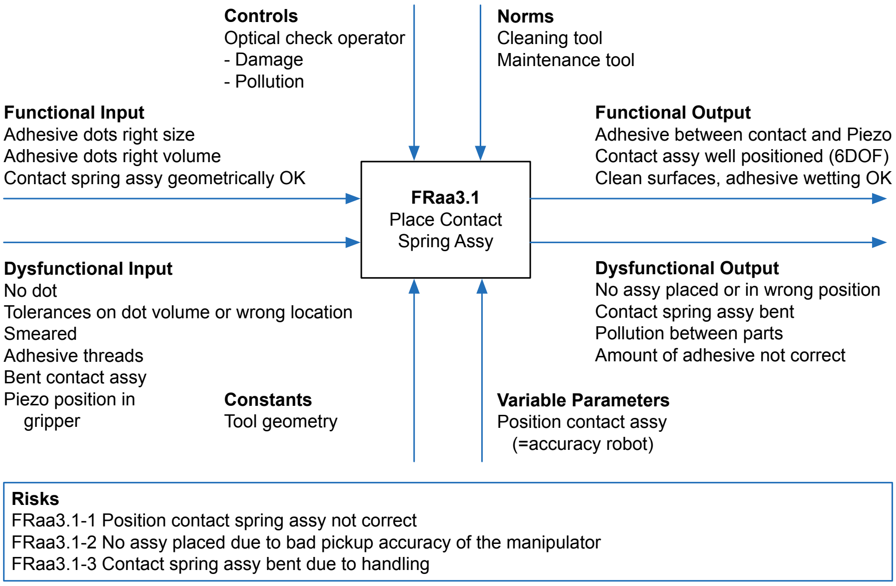


Figure A.6 *SADT parameter analysis of the placement of the contact spring assembly*

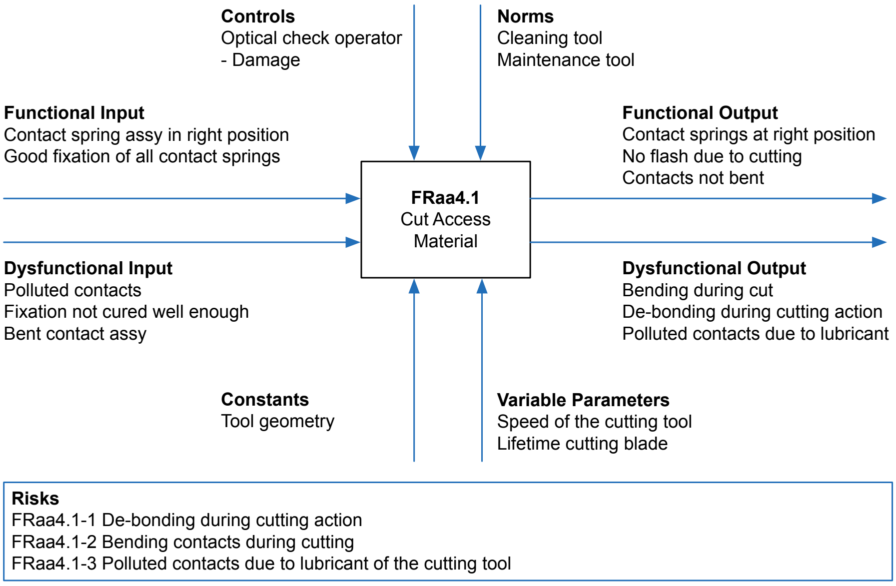
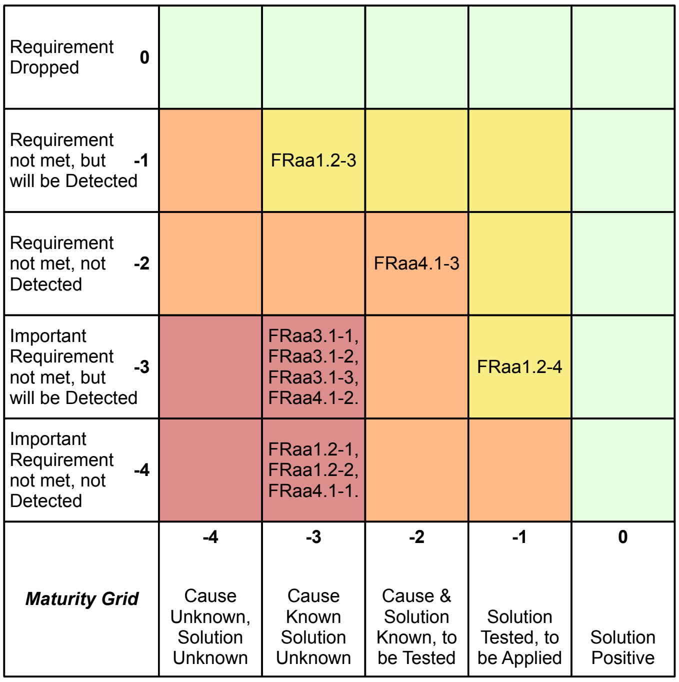


Figure A.7 *SADT parameter analysis of the cutting process after UV fixation of the spring contact assembly*

At this stage, the product flow through the production systems is known and reviewed. All processes need to be made to fit in harmony to prove feasibility of the system. The Maturity Grid is applied to inventory and visualise the remaining risks in the system.

A.2.5 Application of the Maturity Grid

Not all problems in the proposed manufacturing solution are solved at this early stage of production engineering. First, it is still uncertain that all potential problems in the system have been recognised. Secondly, no differentiation has been made between the various risks that have been defined in the SADT parameter analysis. The solution for the first problem is to update the SADT analysis a number of times during the pro-cess of early production engineering. The Maturity Grid will be applied to determine the severity of the risks that have been defined. Risks from the SADT parameter analysis will be plotted in the in the matrix (Figure A.8, Table A.1).



Requirement Dropped	**0**					
Requirement not met, but will be Detected	**-1**		FRaa1.2-3			
Requirement not met, not Detected	**-2**			FRaa4.1-3		
Important Requirement not met, but will be Detected	**-3**		FRaa3.1-1, FRaa3.1-2, FRaa3.1-3, FRaa4.1-2.		FRaa1.2-4	
Important Requirement not met, not Detected	**-4**		FRaa1.2-1, FRaa1.2-2, FRaa4.1-1.			
Maturity Grid		**-4** Cause Unknown, Solution Unknown	**-3** Cause Known Solution Unknown	**-2** Cause & Solution Known, to be Tested	**-1** Solution Tested, to be Applied	**0** Solution Positive

Figure A.8 *The Classic Maturity Grid will give feedback about the severity of uncertainties in the actual production concept*

***Table A.1** Risks in the early process development stage*

FRaa1.2 Dispense Conductive Adhesive	FRaa1.2-1	(-3, -4)	Wrong amount of adhesive
	FRaa1.2-2	(-3, -4)	Short circuit due to adhesive threads
	FRaa1.2-3	(-3, -1)	Polluted Piezo due to misplaced adhesive
	FRaa1.2-4	(-1, -3)	Inaccurate position of the dots
FRaa3.1 Place Contact Spring Assembly	FRaa3.1-1	(-3, -3)	Position Piezo vs contact spring assembly not correct
	FRaa3.1-2	(-3, -3)	Placement accuracy of the contact spring changes while handling
	FRaa3.1-3	(-3, -3)	Contact spring assembly bent due to handling
FRaa4.1 Cut Access Material	FRaa4.1-1	(-3, -4)	De-bonding during cutting action
	FRaa4.1-2	(-3, -3)	Bending contacts during cutting
	FRaa4.1-3	(-2, -2)	Polluted contacts due to lubricant of the cutting tool

A.2.6 Motivation of the Initial Risk Ratings

The risks rating of Table A.1 are motivated as follows:

- FRaa1.2-1 ‘Wrong amount of adhesive’, was set to level (-3, -4), because there was no guarantee that the dispense system would be able to produce a constant dot size without further precautions due to internal and external influential factors of the dispense process. E.g. due to air in the adhesive, when filling the syringe, air bubbles could disrupt the homogeneous character of the adhesive. Since a Time-Pressure dispense system was anticipated on, this effect is not suppressed. Secondly, viscosity changes of the adhesive could lead to disruptions of the amount of adhesive dispensed. In the system, at least two factors that dominantly influence viscosity are present; temperature variation of the environment and ageing of the adhesive in the syringe that causes it to cure in a premature phase;
- FRaa1.2-2 ‘Short circuit due to adhesive treads’ was also set to (-3, -4) due to the risk of short-circuiting the contacts by smearing threads pulled from the adhesive. This problem is not reliably detected with optical inspection;
- FRaa1.2-3 ‘Polluted piezo due to misplaced adhesive’ occurs when adhesive does not reach the right place for the dot e.g. when the needle tip is polluted or vibrates.

Note that this is an aesthetic issue, the product may be expected to function according to specs. Therefore, the status was defined as (-3, -1);

- FRaa1.2-4 ‘Inaccurate position of the dots’ was set to (-1, -3). This process was applied in numerous previous systems and the tolerances were designed vastly within the operational range. The only cause to disrupt the process seemed the situation where the piezo was not gripped accurately. An incorrectly gripped piezo however, is certain to be discovered since the gripper will run short of vacuum, being not able to keep the part firmly enough for placing the contact spring assembly. This would cause the machine to stop. Therefore, the remaining risk was considered minimal;
- FRaa3.1-1 ‘Position piezo vs. contact spring assembly not correct’, FRaa3.1-2 ‘Placement accuracy of the contact spring changes while handling’ & FRaa3.1-3 ‘Contact spring assembly bent due to handling’ were all defined as (-3, -3). At this stage, the needed tolerances required for a correct placement of the contact spring assembly were in range of the maximum performance of the manipulator. This standardised manipulator for this reconfigurable platform was able to perform these accuracies in terms of reproducibility, but not in terms of absolute accuracy. Therefore, the process was considered not mature enough at level 3, meaning that further development was a necessity. Pickup errors are most certainly detected by visual inspection; this justifies category A;
- In case of FRaa4.1-1 ‘De-bonding during cutting action’, a cutting action damages the UV fixation at the risk of failing conductance. The key issue however is that this problem is initially undiscovered. It causes products to fail at a later stage in

the field and in such way that the customer will definitely not accept the problem. Since the cause of the problem is known, the problem qualifies as (-3, -4) risk;

- FRaa4.1-2 ‘Bending contacts during cutting’ could occur when the contact spring assembly was not positioned well on the piezo actuator. In the cutting tool, a support for the contacts during cutting is foreseen, but if the contact has been in placed in a position too high on the piezo, the contacts will bend first till they are supported. At this stage, de-bonding could occur. De-bonding will be noticed; therefore, this risk was set to (-3, -3);
- FRaa4.1-3 ‘Polluted contacts due to lubricant of the cutting tool’. Stamping tools as applied cutting the contact spring assembly from the lead frame are usually lubricated with a mineral lubricant. Access lubricant would, if it were able to reach the bonding surfaces of the piezo, have a negative effect on bonding quality. The ability of the lubricant to actually reach the bonding surface is at least questionable and it would only occur when the tool is lubricated excessively. De-bonding is certainly not accepted. An escape for the problem can be applied by using the cutting tool without lubricant. This reduces the operational life expectations for the tool. The impact of this effect still had to be investigated at this point. For this reason, the risk was qualified as (-2, -2).

A.3 Structured Risk Optimisation using the Maturity Grid; First Improvement Cycle

Now all risks have been charted, the next goal is to structurally reduce them to a safe level ((x, 0) or (0, y)). This may be done by: (i) changing the product design, by (ii) optimisation of the production equipment, or (iii) optimising both concurrently. The risk analysis gives an overview of the magnitude of the risk, but that does only provide an indirect qualification which of the strategies (i, ii or iii) should be applied; small risks may be easily solved by sole optimisation of production equipment while large risks may need changes of product design and equipment. Assuming the product development team has already seriously considered the manufacturing aspects of the product in advance (preferably in consultation with production engineers), a first assumption would be that all processes that are needed for the manufacturing of the product may be expected to be well chosen. Therefore, the first attempt will be to produce the product 'as is' without the need for initial design optimisations. Note that a first evaluation, to determine the process flow through the equipment, has taken place and has led to an early stage design optimisation i.e.; the decision to mount all spring contacts simultaneously. At this stage the focus is on finding solutions to produce the product against acceptable cost with the envisioned layout. Until problems occur, with technical or economic feasibility, this method will be maintained as a starting point.

A.3.1 Dispense Conductive Adhesive (FRaa1.2-1, FRaa4.1-2, FRaa4.1-3 & FRaa4.1-4)

To place the four dots, as the contact at the end face will be connected with adhesive dots on both sides, the manipulator will handle the piezo and move it to the dispensing system. The advantage of this method is that the robotic manipulator can accurately control the movements of the piezo to optimise the dispensing action. A test setup was made to perform early testing of the process. The results of the tests are shown in Figure A.9

Figure A.9 *Sequential stadia of the dispense results as performed by the test setup for process FRaa1.2*

The first test was not successful. As the piezo moved away from the dispense needle, the adhesive dots were torn apart and flipped downwards after 'breaking'. This led to uncontrolled blobs of adhesive on the piezo surface (Figure A.9 left). The cause of the problem is mainly that dispensing in the vertical plane is a procedure that is aberrant from the preferred modality (to perform this task in the horizontal plane). Nevertheless, after slowing down the process, the problem could be reduced as shown in the middle picture. Note that the dot height is still one and a half times the diameter, which is inconvenient for placement of the contact spring assembly. By adding an upward movement at the end of the dispensing action the dots could be flattened providing an excellent solution for this product (right).

After producing a test run of 250 pcs, bonding was tested positive. Smear of the adhesive was low due to the flattened dots and the correct position of the dots. The bonding strength was tested destructively and proved well within margins. The application of de-aired adhesive syringes was selected to prevent problems with air bubbles in the adhesive. Due to these process optimisations, the engineers were confident they had reduced the risks FRaa1.2-1, FRaa1.2-2, FRaa1.2-3 & FRaa1.2-4 ready for application.

A.3.2 Place Contact Spring Assembly (FRaa3.1-1, FRaa3.1-2 & FRaa3.1-3)

To mount the spring contact assembly onto the piezo, a solution was chosen to hold the assembly after cutting it from the lead frame in the cutting tool. This position has accuracy in the micron range determined by the centre pins in the cutting tool. This accuracy suits the strict requirement for the placement process optimally. In this station, the piezo could be inserted in the opening (Figure A.10). Spring action of the contacts from the assembly was designed to supply enough friction to keep the assembly in place. Due to the low weight of the assembly, even the inertial forces due to high acceleration remain acceptably low. The procedure was tested on prototype parts acknowledging the concept of generating enough friction.

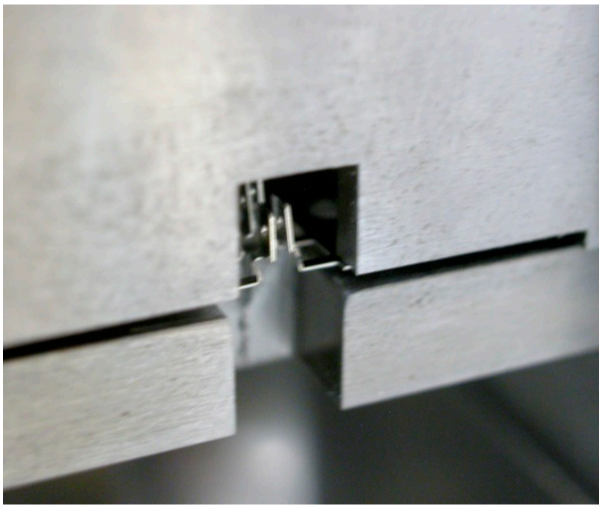


Figure A.10 *Holding the contact spring assembly in the cutting tool after separating it from the lead frame*

The robotic manipulator needed a relatively high accuracy of plus or minus 20 microns in the lateral direction to pick up the spring assembly without damaging it. The available manipulation module, a standard SCARA robot, was able to reproduce this accuracy in a testing environment but when operating 24/7, the thermal growth of the robot-arm was too large to maintain performing in this range of accuracy. Increasing the tolerances in the product design was no option due to the fact that it would reduce the clamping action of the contact spring assembly beyond acceptation, not being able to pick the piezo out of the tool without moving the assembly. A solution was found by increasing the accuracy of the robotic manipulator. At the start of every new tray, roughly 15 minutes, the manipulator would calibrate its absolute position in respect of the cutting tool. This was done by adding a switch with micrometre accuracy on the tip of the robot. Using the switch, it could 'feel' the absolute location of the cutting tool and calibrate the origin for the tool. In this way, the absolute accuracy of the manipulator was made equal to the reproducibility that is specified significantly higher.

Above solutions were tested in a lab setup. Based on the outcome, the engineers decided to reduce the risks FRaa3.1-1 to FRaa3.1-3 to level (-2, -3), 'solution to be tested'.

A.3.3 Cut Access Material (FRaa4.1-1, FRaa4.1-2, FRaa4.1-3)

Cutting the access material was considered as a delicate process step. While cutting, some mechanical load on the contact spring assembly is inevitable. After the cutting procedure, the three contacts of the con-tact spring assembly will be separated. This means that if one of the three contacts will suffer from de-bonding, due to the parasitic forces from cutting action, the product must be rejected.

To get grip on the strength of the bonds, to survive the cutting process, the UV fixation was tested by destructively pulling the contacts from the piezo. The test setup for this is shown in Figure A.11.

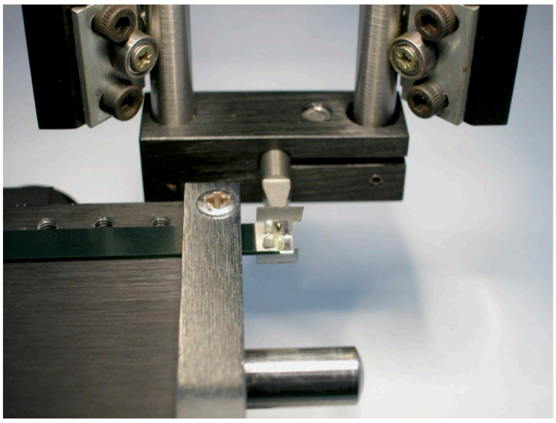


Figure A.11 *Pull testing the strength of the UV adhesive fixation by applying upward force to the contact spring before the cutting process*

The test showed serious bending of the contact assembly, before the actual fixation was destructed. Hence, if the cut-ting process could be performed without noticeable deformation, chances would be fair to assume the bond still being intact. A test sequence of 250 products was cut using a hand tool to evaluate the ratio of defected parts. All parts survived the cutting process without de-bonding. Based on these findings, the engineers were confident to reduce risks FRaa4.1-1 and FRaa4.1-2 to status ‘to be applied’.

The supplier of the cutting tool was consulted about minimising the amount of lubricant for the cutting tool. Based on the lifetime performance of earlier designed tools, calculations made plausible that the tool would last beyond the lifetime stated for this project, even with minimal lubrication. Given this information FRaa4.1-3 was given (0, -2) by the engineers.

A.3.4 Risk Analysis: Reflecting on the First Series of Iterative Loops by a Team of Experts

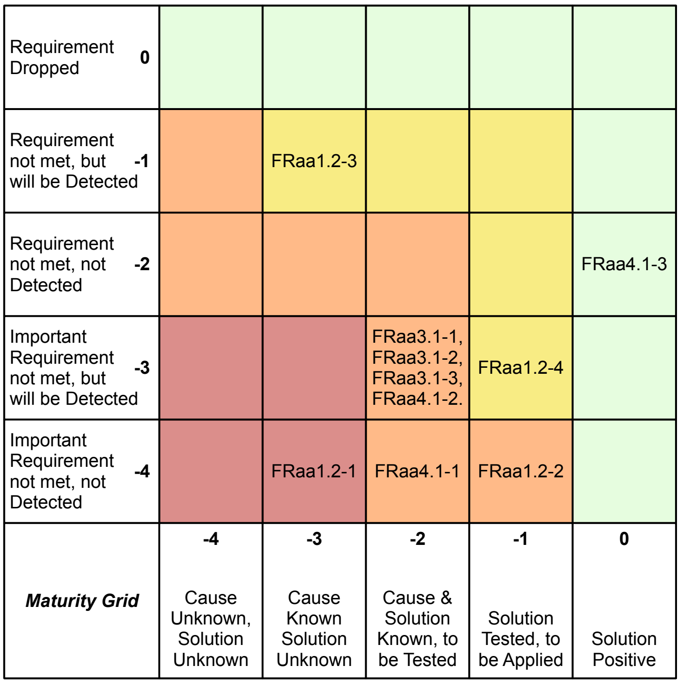
After the risk optimisation phase was completed, the team of experts was again invited to update the Maturity Grid. The matrix was updated based on their impression of the tests that had been done. The analysis by the team of experts came at some points to a different outcome than the engineers'. This was due to a broader scope of the team of experts as it was represented by members of many specialties (product design, system engineering, materials engineering & production engineering). This led to some new insights and different positions for some of the appointed risks (Table A.2):

- FRaa1.2-1 'Wrong amount of adhesive', was set back because the environmental temperature that was not conditioned, would influence the viscosity of the adhesive, causing changes in dot volume. Secondly, there was no guarantee that the position of the syringe-tip was steadily reproducible and could randomly touch the piezo surface. It is not possible to check the dot volume after placement of the contact spring assembly since it will cover the dots. For this reason, the status was set to (-3, -4); it concerns an essential functionality that is not being detected during production;
- FRaa1.2-3 'Polluted piezo due to excess adhesive', which occurs by adhesive wires or excess adhesive caused by a dirty dispense needle, was set back to (-3, -1). It was confirmed by the team of specialists that this is an aesthetic issue and the product may be expected to function according to specs. However, the team did not accept reduction of the risk level until the customer explicitly approved the aesthetic consequences;
- FRaa3.1-1, FRaa3.1-2 & FRaa3.1-3 were accepted by the team as proposed. A new potential problem was identified; if the piezos are not discharged before they are fed into the machine their form will change due to parasitic charge that is collected during earlier processing or temperature changes. This effect is will cause the placement process to fail. The risk status (-2, -3) was subject to be

maintained until the piezos could be sufficiently discharged before the assembly action takes place;

- FRaa4.1-1 & FRaa4.1-2 'De-bonding during cutting action'. Since the cutting tool is still to be designed from scratch it seems inappropriate to assume that the solution is sufficiently tested with a few prototypes and a hand-cutting tool; cutting speeds and geometry will have a comprehensive influence. The risk was redefined to (-2, -4) for FRaa4.1-1 & (-2, -3) for FRaa4.1-2.

The updated results are plotted in the Maturity Grid in Figure A.12.



Requirement Dropped	**0**					
Requirement not met, but will be Detected	**-1**		FRaa1.2-3			
Requirement not met, not Detected	**-2**					FRaa4.1-3
Important Requirement not met, but will be Detected	**-3**			FRaa3.1-1, FRaa3.1-2, FRaa3.1-3, FRaa4.1-2.	FRaa1.2-4	
Important Requirement not met, not Detected	**-4**		FRaa1.2-1	FRaa4.1-1	FRaa1.2-2	
Maturity Grid		**-4** Cause Unknown, Solution Unknown	**-3** Cause Known Solution Unknown	**-2** Cause & Solution Known, to be Tested	**-1** Solution Tested, to be Applied	**0** Solution Positive

Figure A.12 *Updated Maturity Grid after pre-investigations for the production of the automotive actuator*

Table A.2 Redefined positions of the process-risks after completion of the first optimisation cycle

FRaa1.2 Dispense Conductive Adhesive	FRaa1.2-1	(-3, -4)	Wrong amount of adhesive
	FRaa1.2-2	(-3, -4)	Short circuit due to adhesive threads
	FRaa1.2-3	(-3, -1)	Polluted Piezo due to misplaced adhesive
	FRaa1.2-4	(-1, -3)	Inaccurate position of the dots
FRaa3.1 Place Contact Spring Assembly	FRaa3.1-1	(-3, -3)	Position Piezo vs contact spring assembly not correct
	FRaa3.1-2	(-3, -3)	Placement accuracy of the contact spring changes while handling
	FRaa3.1-3	(-3, -3)	Contact spring assembly bent due to handling
FRaa4.1 Cut Access Material	FRaa4.1-1	(-3, -4)	De-bonding during cutting action
	FRaa4.1-2	(-3, -3)	Bending contacts during cutting
	FRaa4.1-3	(-2, -2)	Polluted contacts due to lubricant of the cutting tool

A.4 Structured Risk Optimisation Using the Maturity Grid; Second Improvement Cycle

A.4.1 Addressing Remaining Risks as Set by the Specialist Team

At this point the analysis using the SADT and the Maturity Grid indicates that after the first optimisation cycle, a number of seven processes still have no adequate solution yet. This level of consciousness about the proposed manufacturing system postpones a ‘go’ to continue with the implementation of the proposed solutions. It was decided to add an extra cycle for optimisation of the process technology, to address remaining problems (inventoried in Table A.3):

- In case of FRaa1.2-1 ‘Wrong amount of adhesive’, a solution was found by upgrading process technology with a combination of solutions. First, temperature controlled needle heaters were installed to minimise the influence of the environmental temperature of the production environment. Secondly, a calibration tool was designed to adjust the exact position of the dispense-needle-tips during production, guaranteeing that the syringes would be installed well. After testing the improvements, risk FRaa1.2-1 was reduced back to (-1, -4);

- Problem FRaa1.2-3 'Polluted piezo due to excess adhesive', could be addressed by adding extra inspection intervals at the change of the trays. This however would require time and a number of test samples at every check, lowering the effective output of the production system. Using a test setup, a number of 1000 products were produced. All products were tested electrically and were found to be functioning well. The problem indeed was of an aesthetic- instead of a functional nature. After consulting the final customer of the product, the decision was not to invest in further actions and accept the pollution. The risk of FRaa1.2-3 was set to (-2, 0);
- FRaa3.1-1, FRaa3.1-2 & FRaa3.1-3. The problem of the bending piezos was investigated and appeared to occur in some situations. While cooling down the piezos from the previous heat cycle, the piezos would bend as much as half a millimetre from its neutral axis, causing the accurate placement of the piezo in the contact spring assembly to fail (mounting in the slot shown in Figure A.10. By leaving the piezos to rest after the heat cycle however, the electric charge would flow away through the air depending on the humidity at the works. An intermission of 120 minutes was found to be sufficient to eliminate the problem even with low environmental humidity. By subscribing a waiting time, before parts are allowed to pass, the problem was reduced to position (-1, -4);
- Issue FRaa4.1-1 & FRaa4.1-2 'De-bonding or bending during cutting action', that was set back to (-3, -4) appeared of a serious nature. Though the cutting process has been designed to minimally transfer cutting forces to the bonding plane, tolerances on positioning accuracy and parts would lead to forces on the bond, which may cause de-bonding. The bottom punch will support solder leads of the

contacts during the cut up to some level. Geometry and tolerances of the parts were chosen in such way that the upper punch always applies a downward force to the part. Though this could straighten the parts and reduce tolerances in the final assembly, only a minimal height difference could be eliminated in this way. Finite element calculations of the contact showed that the forces on the bond rose quickly with the height difference due to the fact that the situation was over constrained. To prevent this problem from occurring the contact was modified with a double bend to add extra degrees of freedom to allow straightening (Figure A.13). The width of the contact was reduced to increase contact flexibility further. Based on the earlier performed tests on the bonding strength, the situation was found safe to be implemented; the risk of FRaa4.1-1 was set back to (-1, -4).

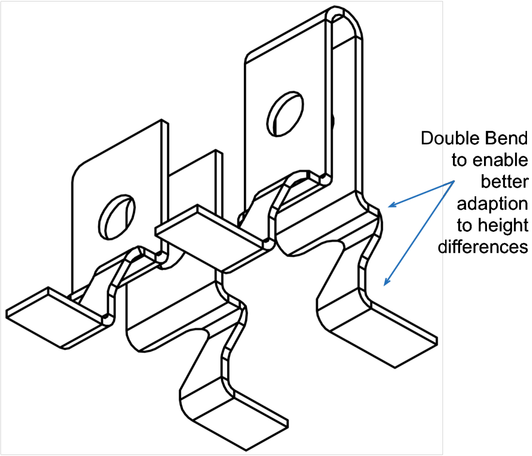
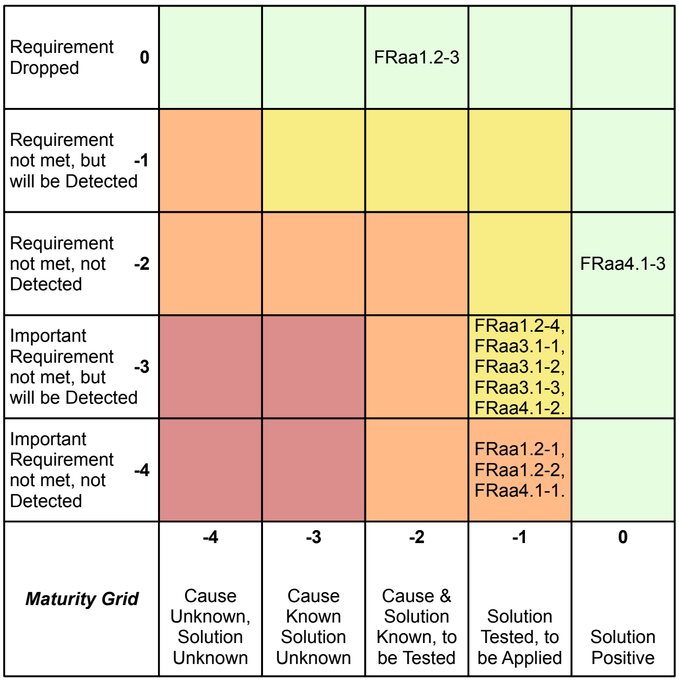


Figure A.13 *Adding a double bend in the contact spring assembly allowed the cutting process to straighten the contacts without having the risk of damaging the fixation on the assembly*

A.4.2 Updating the Maturity Grid

The updated results are plotted in in the Maturity Grid in Figure A.14.



Requirement Dropped	**0**			FRaa1.2-3		
Requirement not met, but will be Detected	**-1**					
Requirement not met, not Detected	**-2**					FRaa4.1-3
Important Requirement not met, but will be Detected	**-3**				FRaa1.2-4, FRaa3.1-1, FRaa3.1-2, FRaa3.1-3, FRaa4.1-2.	
Important Requirement not met, not Detected	**-4**				FRaa1.2-1, FRaa1.2-2, FRaa4.1-1.	
Maturity Grid		**-4** Cause Unknown, Solution Unknown	**-3** Cause Known Solution Unknown	**-2** Cause & Solution Known, to be Tested	**-1** Solution Tested, to be Applied	**0** Solution Positive

Figure A.14 *Updated Maturity Grid after pre-investigations for the production of the automotive actuator*

Table A.3 *Redefined positions of the process-risks after completion of the second optimisation cycle*

FRaa1.2 Dispense Conductive Adhesive	FRaa1.2-1	(-1, -4)	Wrong amount of adhesive
	FRaa1.2-2	(-1, -4)	Short circuit due to adhesive wires
	FRaa1.2-3	(-2, -0)	Polluted Piezo due to misplaced adhesive
	FRaa1.2-4	(-1, -3)	Inaccurate position of the dots
FRaa3.1 Place Contact Spring Assembly	FRaa3.1-1	(-1, -3)	Position Piezo vs. contact spring assembly not correct
	FRaa3.1-2	(-1, -3)	Placement accuracy of the contact spring changes while handling
	FRaa3.1-3	(-1, -3)	Contact spring assembly bent due to handling
FRaa4.1 Cut Access Material	FRaa4.1-1	(-1, -4)	De-bonding during cutting action
	FRaa4.1-2	(-1, -3)	Bending contacts during cutting
	FRaa4.1-3	(-0, -2)	Polluted contacts due to lubricant of the cutting tool

A.5 Integration of the Production System

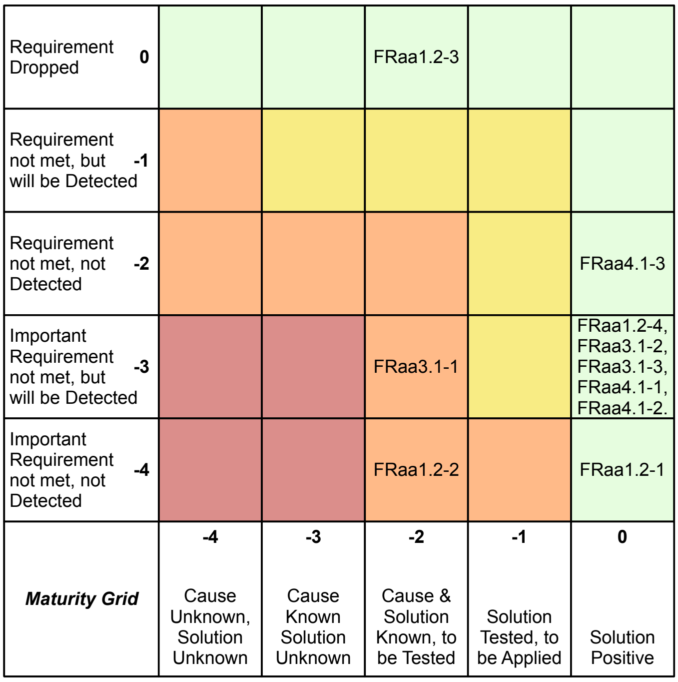
A.5.1 Hardware and Software Integration

With the process flow through the system determined, the system decomposed to elementary functions and the general system architecture known, the production system could be engineered in detail. Since this was a reconfigurable manufacturing platform, a framework of *process modules* was existent and a most modules were reused keeping engineering efforts minimal. The whole process could be integrated on a configurable production cell of 1m^2.

The hardware integration of the system took eight weeks from finalisation of the last Maturity Grid. After that, a number of four weeks were needed for debugging control software. After completion of the total system, hardware and software, the machine was considered ready for pilot production. From this moment on, pilot series were produced on a regular basis. During pilot production, preliminary product series were delivered to the customer and applied in prototype products. During this phase, all parts were inspected both visually and electrically. Statistical information was kept and inventoried. During the first 12 weeks from start production, the development team followed the manufacturing performance accurately, 100% monitoring the production. After twelve weeks, the daily numbers produced grew over 1500 products. From this moment on samples were taken and analysed producing statistical data of the systems performance.

A.5.2 Troubleshooting During Pilot Production

Many small optimisations were made in the equipment control software, to increase consistency, like the manufacturing yield, and to speed up the production process to the desired performance. During week 2 however, problems of a more serious nature occurred. The contact spring assembly did not stay in place, after initially being positioned correctly, before the UV curing adhesive could fixate the assembly. The contact spring assembly was designed to have clamping action but in some situations, the friction seemed too low to keep the assembly in place. Measuring the metal lead frame with the contact spring assembly and the piezo actuators learned that the dimensions of the contacts were within tolerances as were the thicknesses of the piezo itself. In worst-case conditions, however, the clamping force was just too low. Basically, this was a design issue; orienting tests had been performed to choose the dimensions and their tolerances, but apparently, the working range had not been determined well. The result was twofold; the position of the contact spring assembly could not be maintained till UV fixation had taken place and, due to movement of the assembly, smearing adhesive was causing short circuit. This last problem was mainly the case when the amount of conductive dispensed adhesive was on the upper limit. The problem results in roughly 50% non-functioning products and thus destroying the opportunity of a good yield. The problem is visualised in the Maturity Grid in Figure A.15.

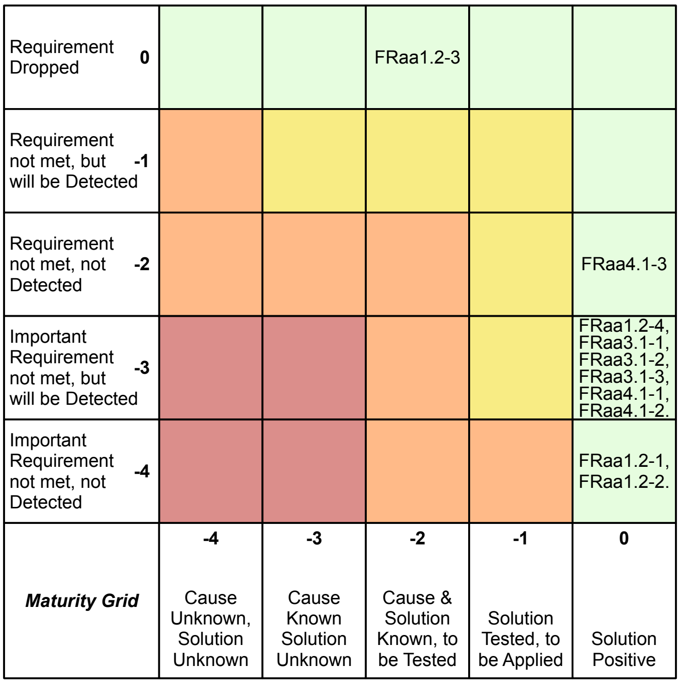


Requirement Dropped	**0**			FRaa1.2-3		
Requirement not met, but will be Detected	**-1**					
Requirement not met, not Detected	**-2**					FRaa4.1-3
Important Requirement not met, but will be Detected	**-3**			FRaa3.1-1		FRaa1.2-4, FRaa3.1-2, FRaa3.1-3, FRaa4.1-1, FRaa4.1-2.
Important Requirement not met, not Detected	**-4**			FRaa1.2-2		FRaa1.2-1
Maturity Grid		**-4** Cause Unknown, Solution Unknown	**-3** Cause Known Solution Unknown	**-2** Cause & Solution Known, to be Tested	**-1** Solution Tested, to be Applied	**0** Solution Positive

Figure A.15 *Moving contact spring assemblies, in combination with misplaced adhesive, may lead to short circuited contacts (FRaa1.2-2) and an incorrectly positioned assembly (FRaa3.1-1)*

To solve the problem, a two-stage solution was chosen. Initially, batches of thin piezos were skipped and the amount of adhesive was reduced. The tool for production of the lead-fame could be slightly adjusted to reduce the opening of the spring contact assembly. These measures took care for a workaround on the short term. In parallel, a more structural solution was prepared by modifying the production tool for the lead-frame so the opening could be reduced further and the clamping action could be guaranteed under all tolerance combinations.

After implementation of the improvements, all risks as defined were in the category ‘Solution Positive’. With all significant obstacles addressed, it was expected that the yield would rise quickly (Figure A.16).



Requirement Dropped	**0**			FRaa1.2-3		
Requirement not met, but will be Detected	**-1**					
Requirement not met, not Detected	**-2**					FRaa4.1-3
Important Requirement not met, but will be Detected	**-3**					FRaa1.2-4, FRaa3.1-1, FRaa3.1-2, FRaa3.1-3, FRaa4.1-1, FRaa4.1-2.
Important Requirement not met, not Detected	**-4**					FRaa1.2-1, FRaa1.2-2.
Maturity Grid		**-4** Cause Unknown, Solution Unknown	**-3** Cause Known Solution Unknown	**-2** Cause & Solution Known, to be Tested	**-1** Solution Tested, to be Applied	**0** Solution Positive

Figure A.16 *After solving the clamping problem and fine-tuning parameters all risks were considered under control in week 4*

A.5.3 Increasing Manufacturing Yield

The yield as determined in this project uses the general definition that “manufacturing yield’ is equal to the division of ‘Correctly produced products’ and ‘Total number of products that initially was started with’.

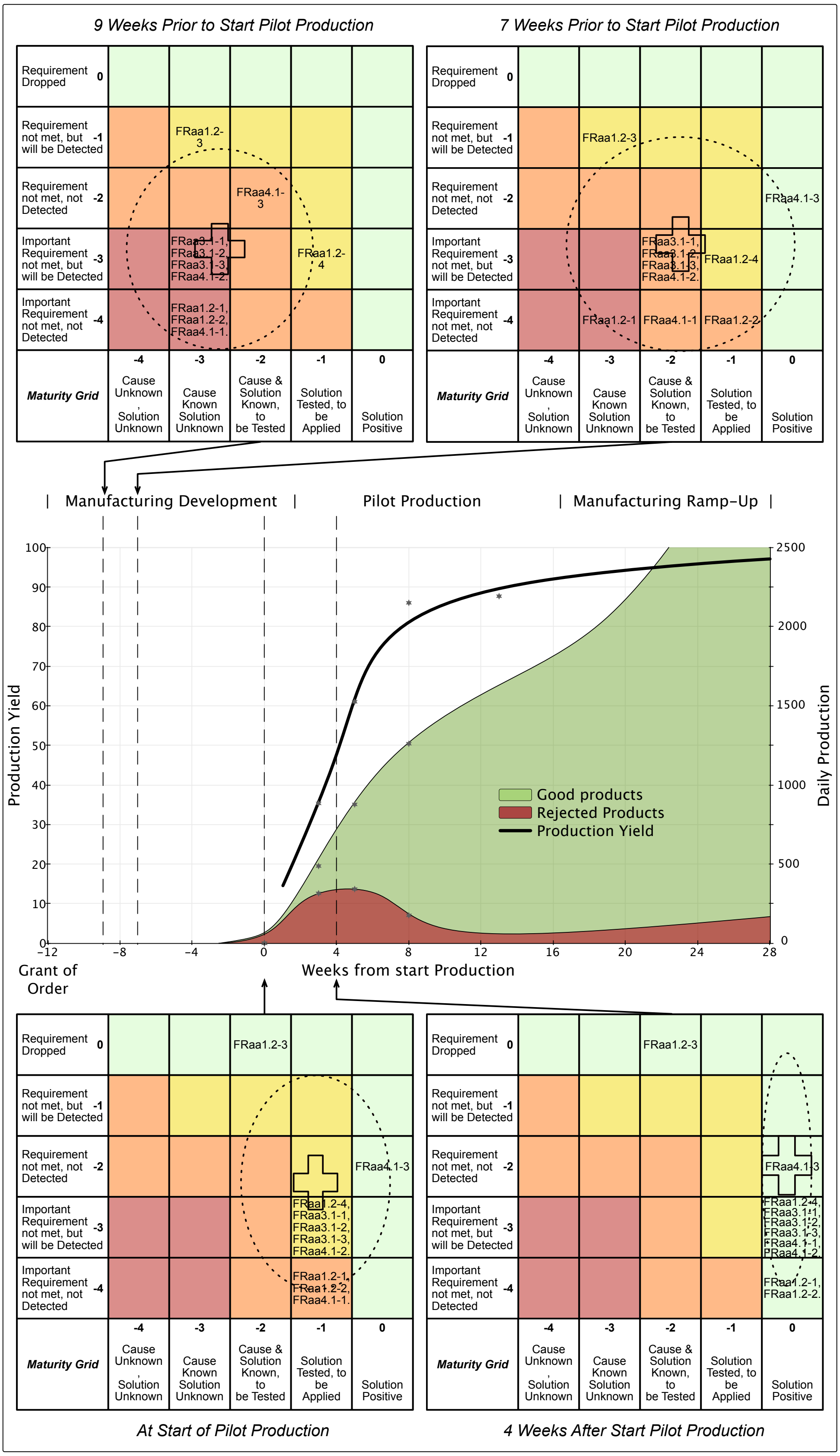


Figure A.17 *Development of production yield as function of time (dark line)*

The production yield as realised by this project is shown in Figure A.17. The following milestones apply:

- Week number -12 is the moment that the order was granted;
- Week number 0 is the official start of the 'pilot production';
- Week number 16 is the start of the regular production. The equipment was transported to the manufacturing location.

The manufacturing yield raised in the first eight weeks to over 80% and after fourteen weeks over 90%. Most rejected parts were produced in the first eight weeks. Due to a rapid growth of production at a time where the manufacturing yield is still moderate, a lot of scrap parts were produced. This however was not different from earlier projects.

After pilot production, when the equipment was moved to the production site, it was also placed in line with the manufacturing equipment of the lead frames. By this change, the capability of the production system was able to improve further. The targeted daily production of 5000 products was reached before week 32 (outside graph). The manufacturing yield at that point exceeded 98,5%.

APPENDIX B
AXIOMATIC DESIGN

B.1 The Axiomatic Domains and Design Equations

AD demands clear formulation of design objectives through the establishment of ‘Domains’ called: (i) ‘Customer Attributes’, (ii) ‘Functional Requirements’, (iii) ‘Design Parameters’, and (iv) ‘Process Variables’. The domains are hierarchically decomposed (Figure B.1).

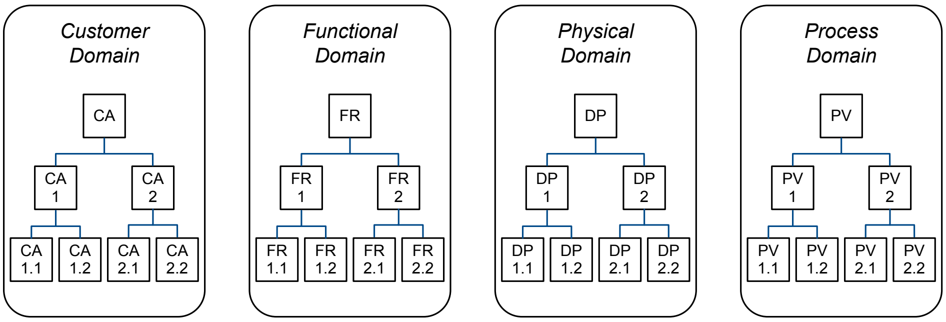


Figure B.1 *Axiomatic Domains and their hierarchical organisation*

To investigate the relations between these domains, AD declares ‘Axioms’ that cannot be proven nor derived from physical phenomena. A number of seven conceptual axioms were defined in 1978 when the first paper about AD was presented (Suh et al., 1978). Two of those seven axioms stood the test of time and form the foundation of AD today, now known as the ‘Independence Axiom’ and the ‘Information Axiom’. The Independence Axiom advises to ‘Maintain the independence of the functional requirements’, the Information Axiom recommends to ‘Minimise the information content of the design’. A product design will be a *good design* if both axioms are satisfied. AD also explains how the axioms may be satisfied as shown in Figure B.2. The right-hand three domains are mathematically related (Equations B.1 and B.2) but this is not possible for the customer domain; the customer domain is therefore disregarded for now.

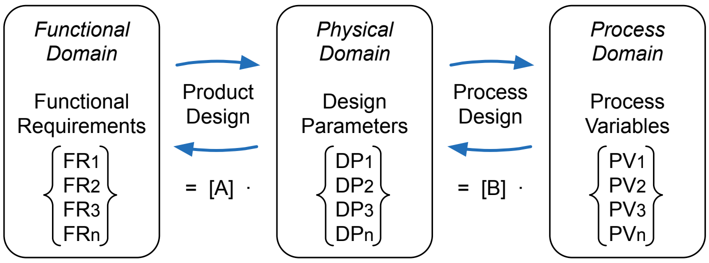


Figure B.2 *Axiomatic Domains and their relations*

The domains in which functional requirements (FR), design parameters (DP) and process variables (PV) are represented as vectors are interrelated with design matrices (B.3), starting with the design Equations according to good-AD-practice (Suh, 1990)

$$\{FR\} = [A] \cdot \{DP\} \tag{B.1}$$

$$\{DP\} = [B] \cdot \{PV\} \tag{B.2}$$

where [A] & [B] are the product and process design matrices. If a product design has three FRs and three DPs, the product design matrix has the following form

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{B.3}$$

and a *good design* would be ‘Uncoupled’ or ‘Decoupled’ if the matrix is diagonal or triangular as respective shown in Equations (B.4) and (B.5),

$$[A] = \begin{bmatrix} \mathrm{X} & 0 & 0 \\ 0 & \mathrm{X} & 0 \\ 0 & 0 & \mathrm{X} \end{bmatrix} \tag{B.4}$$

$$[A] = \begin{bmatrix} \mathrm{X} & 0 & 0 \\ \mathrm{X} & \mathrm{X} & 0 \\ \mathrm{X} & \mathrm{X} & \mathrm{X} \end{bmatrix} \tag{B.5}$$

where the X-es indicate non-zero elements of the matrix and as such indicate a relation between the associated DPs and the FRs. In an uncoupled design, every FR is related to a single DP. In a decoupled design, it may be related to more than one DP, but if the right order is applied to adjust the FRs with the DPs, all FRs can be tuned sequentially.

B.2 The Process of Zigzagging

To check if all FRs are satisfied by their DPs and if subsequently the DPs are satisfied by their PVs, AD uses a procedure called 'Zigzagging'. *Zigzagging* is a top down descend through the hierarchy of the design while all domains are successively addressed. At every level a check is performed if the FRs and DPs are satisfied before going down on level, this is shown in Figure B.3.

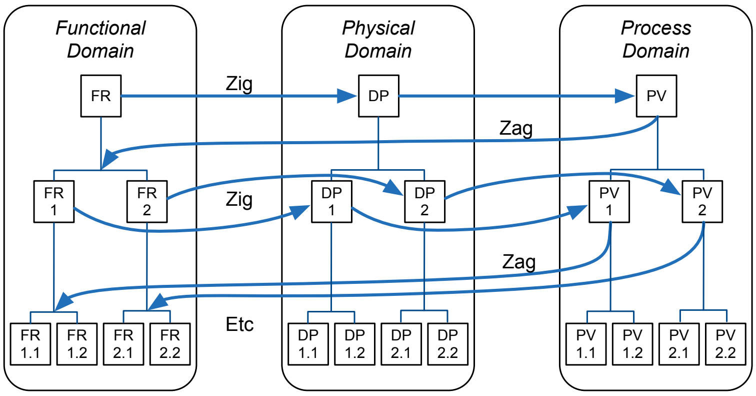


Figure B.3 *The process of hierarchically zigzagging through the domains*

The process of *zigzagging* is always performed from the left- to the right-hand side. *Zigzagging* preferably covers all domains. In practice, it is not always possible to incorporate the customer domain since the functional requirements have been recorded in the project agreement with the customer. Successful completion of the *zigzagging* process will lead to an uncoupled or a decoupled design matrix and satisfies the Independence Axiom which completes the conceptual phase of the project.

B.3 Robustness in Axiomatic Design

The completion of the process of *zigzagging* does not automatically guarantee that the FRs will always be satisfied because tolerances on the DPs and PVs may lead to drift of the FRs and cause them to move outside their tolerance range; this indicates a problem with robustness of the design. Here the Information Axiom is applied to make sure that the FRs will stay within the envisioned tolerances. The concept of 'Information in Design' will be exhaustively explained in Chapter 5. For now, it satisfies to explain the concept of overlap between the 'Design Range' and the 'System Range' of the design. The FR will be satisfied if the actual value as realised in the physical system is within the boundaries of the design range. The realised value behaves according to a system 'Probability Density Function' as shown in Figure B.4. Only in the cases of overlap of the two ranges, the shaded area, the FR is successfully satisfied. The probability of a good outcome is analogue to the area of the overlap as represented by the 'Common Range'.

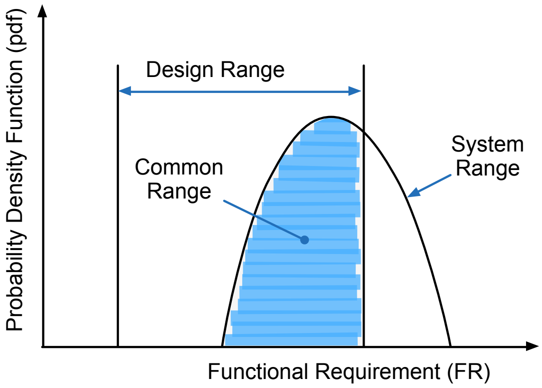


Figure B.4 *Design range, System range and the Common range*

Ideally, the common ranges of the design are (close to) 100%, which guarantees that the FRs will be satisfied under almost all circumstances and the design may be considered to be a robust design. In this case, the 'Information Content' of the design is minimal and the Information Axiom is satisfied.

APPENDIX C
DEPENDENCIES OF THE AXIOMS IN AD

C.1 Investigations into the Dependency of the Axioms

The dependency of the axioms has been investigated a number of times. The first book about AD (Suh, 1990) includes a paragraph about the relationship between axioms 1 and 2. Suh addresses the misunderstanding that the Independence Axiom is a consequence of the Information Axiom, by explaining that a coupled design could have lower information content than an uncoupled design. Without the Independence Axiom, it is not possible to choose the uncoupled design, which, from the design perspective, is more preferred than the coupled design. The second book (Suh, 2001) contains some mathematical proof of the independence, based on the Boltzmann entropy of the FR array as was published by (El-Haik & Yang, 1999). If the design matrix is square and non-singular with constant entries, and DPs are normally distributed random variables, the entropy h of the FRs is given by

$$h(\{FR\}) = h(\{DP\}) + ln|[A]| \tag{C.1}$$

where |[A]| is the determinant of the design matrix [A]. Investigation of the determinant leads to the understanding that a coupled matrix can indeed have lower information content than an uncoupled matrix, which was reflected by the substantiation of corollary 7. In 2005, the book of El-Haik confirms Equation (C.1) (EI-Haik, 2005). Based on these investigations it may be concluded that both axioms serve a particular goal and should be maintained.

C.2 Background on Axiomatic Complexity

Complexity is defined as 'A measure of uncertainty in achieving the specified FRs' (Suh, 2005b). The Complexity Axiom advises to 'Reduce the complexity of a system'. The theory defines two kinds of complexity, 'Time-Independent' and 'Time-Dependent Complexity'. In the case of time-independent complexity, the behaviour is governed by the given set of FR and DP relationships. Time-dependent complexity depends upon the initial condition with FR and DP relationships, but unless the system goes back to the same set of initial conditions periodically, the distant future behaviour is totally unpredictable as the system tends to escalate (Suh, 1999). Time-dependent complexity is not further investigated in this thesis.

Time-independent complexity consists of two components: 'Real' and 'Imaginary' time-independent complexity, further to be referred to as *real complexity* and *imaginary complexity* (C_R and C_{Im}). *Real complexity* is inversely related to the probability of success that the associated FRs are satisfied according to one of the following relations

$$C_R = -\sum_{i=1}^{m} log_b P_i \qquad (C.2)$$

$$C_R = -\sum_{i=1}^{m} log_b P_{i|\{j\}} \qquad for\ \{j\} = \{1,2,..,i-1\} \qquad (C.3)$$

depending if the system is uncoupled (6.2) or decoupled (6.3). Relation (6.2) is under the reservation that the total probability Pi is the 'joint probability of processes that are statistically independent'. Relation (6.3), for decoupled systems, is modified to correct for dependencies in the probabilistic function (Suh, 2005b). 'b' Is in both cases the base of the logarithm, usually in bits or nats depending of the preferred definition. Given (6.2) and (6.3), real complexity can be related to the information content in AD, which was defined in terms of the probability of success of achieving the desired set of FRs (Suh, 1990), as

$$C_R = I \tag{C.4}$$

in which C_R is *real complexity* and I is information according to the definition of Shannon (Shannon, 1948).

Imaginary complexity is defined as complexity that exists due to ‘a lack of understanding about the system design, system architecture or system behaviour’ (Suh, 1999). It is caused by the absence of essential knowledge of the system. The designer cannot solve the problems in a structured manner and therefore is forced to apply trial-and-error. Imaginary complexity exists due to a lack of knowledge of the designer.

C.3 Breakdown of Complexity in the Context of AD

Like most definitions in AD, complexity is also defined in the functional domain. This implies that ‘Total Complexity’ can be decomposed in a functional and a non-functional part analogue to information. The breakdown of total complexity is given in Figure C.1. *Total complexity* is broken down in a functional part ‘Complexity according to the Complexity Axiom’ and a ‘Superfluous’ part. *Superfluous complexity* has no effect on the FRs of the system and therefore is not relevant for AD. It is further ignored. *Real complexity* is by definition equal to the information of the Information Axiom; their direct relation was given by Equation (C.4).

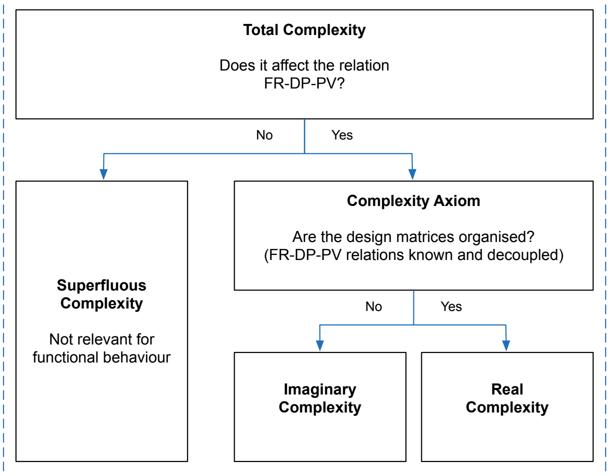


Figure C.1 *Breakdown of total complexity*

Imaginary complexity is harder to understand. Suh defines it as ‘uncertainty that is no real uncertainty’ and ‘it arises because of the designer’s lack of knowledge and understanding’ (Suh, 2005a). The book states further, ‘when a design is uncoupled or decoupled, the imaginary component of complexity is equal to zero’ (Suh, 2005a). For a decoupled design this is only guaranteed if the optimisation order of the design relations is known. Imaginary complexity is inversely related to the satisfaction of the Independence Axiom.

APPENDIX D
THE CYNEFIN FRAMEWORK

Cynefin is a decision-making framework that can be applied on organisations, systems, or even complex social environments. It was applied, evaluated, and refined at the IBM Institute of Knowledge Management (Kurtz et al., 2003), and later expanded to be used as a leadership model (Snowden & Boone, 2007). Cynefin has not yet gained much drag within the AD community or even product development in general, but with the view on information in AD as reported by Puik & Ceglarek (Puik & Ceglarek, 2014a), both methodologies appear to connect and harmonise well together.

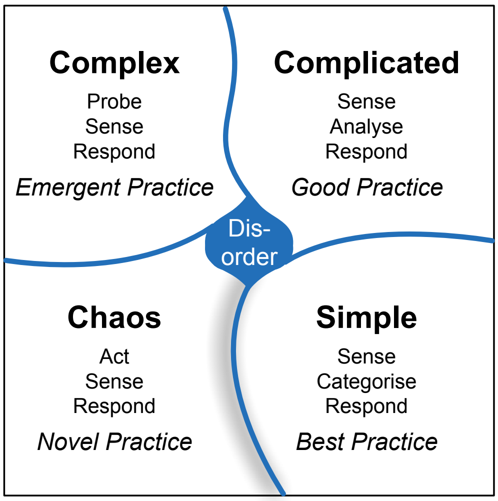


Figure D.1 *The four contexts of the Cynefin framework.*
When in disorder, the actual context is not known

The framework consists of three basic types of systems; ‘Ordered’ systems, ‘Complex’ systems and ‘Chaotic’ systems. Ordered systems are divided in to two types: ‘Simple’ ordered systems and ‘Complicated’ ordered systems. In the centre of the four contexts is a fifth field added: ‘Disorder’. Together this leads to the Cynefin framework as shown in Figure D.1.

- In the Simple context, cause and effect relationships are clear, predictable, repeatable, and generally linear. The systems in this context are self-evident to every reasonable person. The decision model of the Simple context is sense-categorise-respond. Good response in these situations would be to watch what is coming in, match it to previously determined categories and decide what to do. The Simple context is the context of ‘best practice’;
- In the Complicated context, there is a logical relation between cause and effect, but it is not self-evident and therefore requires expertise. An analytical method is needed to solve problems, or an expert could be called in. The decision model therefore is sense-analyse-respond. The Complicated context is the context of ‘good practice’;
- A complex system is a system without causality. Cause and effect are only obvious in hindsight, with unpredictable emergent outcomes. The decision model is probe-sense-respond. Carrying out experiments is a key characteristic; a successful outcome is enhanced; a bad outcome is suppressed. Actions lead to a novel way of doing things. The Complex context is the context of ‘emergent practice’;
- A Chaotic system shows no relation between cause and effect. The goal should be to restore order. The decision model therefore is to act-sense-respond. Actions will be new and unconventional. This is the context of ‘novel practice’;
- Disorder is the space when it is not clear to which context a situation should be appointed.

The boundaries between the contexts are transitions that can be taken without specific effects, except for the boundary between the Simple context and the Chaotic context. This boundary is referred to as the ‘Complacent Zone’ or the ‘Cliff’. The danger is that once a system is in the Simple context, people start to believe that things are simple by nature. It may lead to the belief that things are always ordered and that success from the past is proof that systems cannot fail. The result is that the actual position moves to the border and at a given moment falls over the cliff into a crisis (Kurtz et al., 2003).

APPENDIX E
APPLICATIONS OF THE AXIOMATIC MATURITY DIAGRAM

E.1 An On-Line Payment Application

The first case concerns a Dutch company that delivers solutions for on-line payments. This store management system combines online payments, store payments, and integrates stock keeping of stores and warehouses in a single solution. It is a complex multi-mainframe system with many interfaces. The system may be seen as the core around which the operations of many stores are organised. If the system goes down, no payments can be done in both physical and on-line stores.

Because of the importance for the operations to the customers, the company gives an up-time guarantee with penalty clause. Maintenance of the system is done at certain nights of the week when all stores are closed. Regular updates take place to add features and to correct malfunctions. Backups are made to secure data. All this is done over the internet from a single location in the Netherlands over several thousands of cash registers in Europe. Security is a significant issue; many attempts to hack the system take place. The company also hires professional hackers to test the system for vulnerabilities. Since all systems are connected to the internet there are many interfaces and even more ports to approach the system.

E.1.1 The Problem

At a given day, the ICT manager of the company is hinted by the one of the professional hackers that a certain interface gives access to the system because a port is opened. The manager has this problem examined by the team and they confirm that the port is opened. This is a necessity to ad certain functionality to the system. However, the team is convinced that this vulnerability is not of a worrisome nature. Some days later, the manager is not quite comfortable with the situation and in the evening, he tries to get access to the system from his home location. To his surprise, he is able to gain access without any password and he is also able to start and stop processes on the mainframes and even worse, he is able to execute ghost payments.

E.1.2 The Consequence

The same evening, the manager reports the problem to the general manager. That same night they try to reconstruct the origin of the problem. They conclude that the vulnerability has been there for over six months. Next morning, a crash team is composed. A risk analysis indicates a severe problem. A few thousand systems are in the field with the same vulnerability.

E.1.3 The Solution

The problem can be fixed; the team has to reroute a number of communication channels to restore the vulnerability. After some long days, a fix is completed. It is implemented on a test system and tested for a week. After this it is rolled out to a limited number of systems before it is rolled out completely.

E.1.4 Elaboration from the Perspective of Information

In the beginning of this evaluation the vulnerability already is in the system. There is peace in the company because no one is aware of the problem. But this calm is unfounded; The system may be terribly hacked any moment with the result that the system can be halted or fake bank transfers take place. All this is caused by the presence of *unrecognised information* in the system.

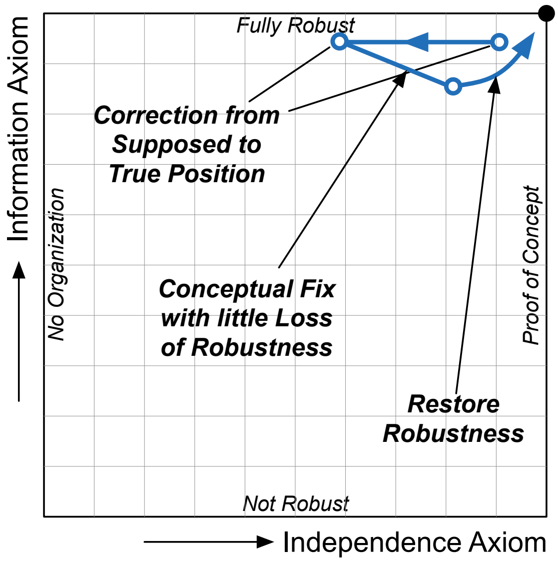


Figure E.1 *The designer is confronted with a correction in the Axiomatic Maturity Diagram when unrecognised information reveals itself*

Once discovered, the calmness in the company gives way to the 'Restrained Panic' of the knowledge that anything may go wrong any moment. This is visualised in Figure E.1 with a drop in the Axiomatic Maturity Diagram. After this, the engineers concurrently develop a conceptual fix. The fix needs changes in the software design which may reduce the robustness of the system. Robustness is regained by testing the system again. The end position in the Axiomatic Maturity Diagram is comparable to the supposed end position before learning about the *unrecognised information* but it is more mature than the true starting point.

In the Cynefin framework, the situation moves from the Simple context directly to the Complex context. There is no state of chaos in the company, but all engineers feel the pressure to understand the situation and come up with the solution. Since it is a complex system, they need time to find that solution. Once rolling out the system starts, the company comes at ease and moves via the Complicated context back to Simple.

E.2 Case 2 De Havilland Comet

The second case is a case from the history books. It concerns the De Havilland Comet. Extensive research has been done to find the cause and effect of this case (Withey, 1997; Wanhill et al., 2016). The Comet was the world’s first production commercial jetliner developed and manufactured by de Havilland. It featured an aerodynamically clean design with four turbojet engines buried in the wings, a pressurised fuselage, and large square windows. The plane was a gigantic step forward in avionics, with cruising speeds up to 800 km/h and cruising altitudes of over 13.000 metres.

E.2.1 The Problem

In 1954 two de Havilland Comets broke up in flight with no apparent reason known at that time. Because of this, the plane was grounded.

E.2.2 The Consequence

Investigations were needed to find the problem that caused the two crashes in 1954 and this appeared a complex problem. The planes were put in a water basin to test the integrity of the fuselage by pressurising it. After a relatively low number of load changes, it ruptured. Further investigation learned that fatigue cracks starting at the pivots of the square windows and hatches led to accelerated growth of cracks. When the cracks became too large, the fuselage ruptured, starting at the forward escape hatch and the top hatches (Figure E.2).

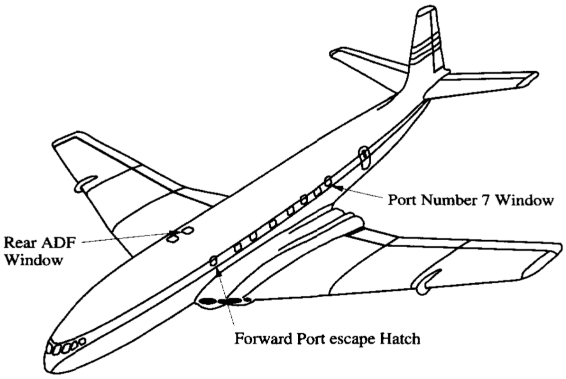


Figure E.2 *Cracks started at the escape hatches and the windows that both had a square shape and were pivoted (Withey, 1997)*

E.2.3 The Solution

The problem was solved by an increase of the hull thickness, and by improving the pivots and the shape of hatches and windows, but it took four years before commercial flights resumed.

E.2.4 Elaboration from the Perspective of Information

At the beginning of the design, this plane already suffered from the weakness that the shape of the square hatches led to tension concentrations in the metal. Pivots weakened the fuselage further at the locations with high tension. The wall of the fuselage was relatively thin to save weight and the combination of these factors led to the presence of *unrecognised information*. When this information came to the surface the problems were difficult to oversee; not only many lives were lost but also the confidence in the safety of the plane disappeared; enthusiasm about a great plane made place for total chaos.

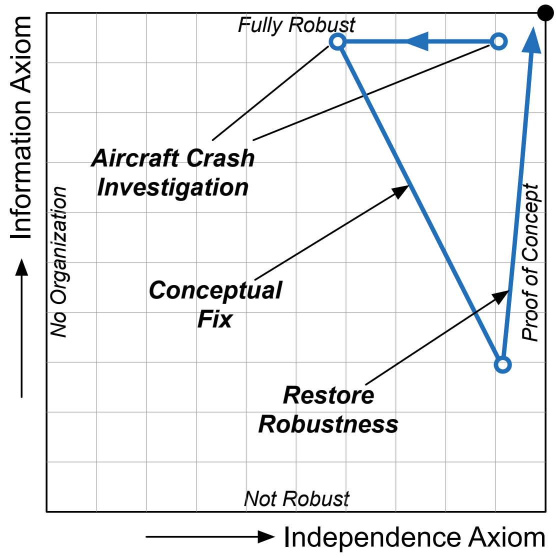


Figure E.3 In this situation a safe development path through the Axiomatic Maturity Diagram is followed that is characterised by a steep incline during the restoration of robustness

The aircraft crash investigation that followed revealed the true position in the Axiomatic Maturity Diagram (Figure E.3). The fatigue problem shortened the lifespan of the plane, and that FR was no longer satisfied. Substantial conceptual improvements were needed which resulted in a significant drop in robustness. The improvements restored the independence of the system up to a high extent. The repair cycle ended with restoring the robustness by repeating the many tests that are required to get the necessary permissions to resume service.

In the Cynefin framework, the situation moves from the Simple context to the complacent zone and falls over the cliff, straight into chaos. When the fuselage tests were completed the relation between cause and effect was restored. Based on that understanding a new start could be made by the De Havilland Comet. That restart was successful from the technological perspective as the fuselages remained intact from that moment.

APPENDIX F

DESIGN & IMPLEMENTATION OF A MICRO HYDROGEN SENSOR

	Constituent Roadmap Scores at start project						
	Customer Attributes		***Functional Requirements***		***Design Parameters***		***Process Variables***
Project	- Foundations of Energy Harvesting at the nanoscale: Demonstration of radically new strategies for energy harvesting and local storage below the micrometre scale. Exploration and harnessing of potential energy sources at that scale including kinetic energy present in the form of random fluctuations, ambient electromagnetic radiation, chemical energy and others; - Self-powered autonomous Nanoscale electronic devices: Autonomous Nanoscale electronic devices that harvest energy from the environment, possibly combining multiple sources, and store it locally. These systems would coordinate low-power sensing, processing, actuation, communication and energy provision into autonomous wireless Nanosystems.	2	FR Develop a self-powered ICT system that is beyond state of the art.	2	DP Develop smallest PV powered sensor in the world.	2	PV Apply CMOS compatible production processes for all silicon dies; PV' Apply engineering methods systematically; PV'' Apply a project team with proven credentials for the field of research.
Product	- Possibility of building autonomous Nanoscale devices (from sensor to actuators), extending the miniaturisation of autonomous devices beyond the level of the 'smart dust'; - New applications in a vast number of ICT fields such as intelligent distributed sensing, for health, safety-critical systems or environment monitoring.	2	*Constraint: Develop the smallest PV powered sensor in the world.* FR1 Harvest energy for operation from the environment; FR2 Manage the energy in the device; FR3 Quantify environmental chemical substance; FR4 Provide power and data to the RF frontend.	2	*Constraint: Apply engineering methods systematically.* DP1 Apply nanowire PV Cells; DP2 Include Energy Management circuitry; DP3 Develop sensor die to measure chemical substance; DP4 Include external interface.	2	*Constraint: Apply CMOS compatible production processes for all silicon dies.* PV1-3 Dies will have their specific CMOS manufacturing recipe and will be made in batches per kind; PV4 Interface with substrate and standardised wirebond process.
System			Limited to Measuring System (Energy Harvester, Energy Manager, and Integration not further elaborated): - Enables bio sensing; - Detection of specific molecules at the surface of nanowires.	0	Limited to Measuring System (PV Cells, Energy Management Circuitry, and Integration not further elaborated): - FET device; - Linking receptor groups to enable surface functionalised Nanowires.	1	Limited to Measuring System (PV Cells, Energy Management Circuitry, and Integration not further elaborated): - PMMS Microfluidic system; - Flow structure made with lithography process.
Part			Limited to Measuring System: - Contain functionalised surface; - Direct medium along functionalised surface.	0	Limited to Measuring System: - Channels in PMMS; - Some Functionalisation for probe/target reactions.	0	Limited to Measuring System: - PMMS Channel structuring; - PMMS bonding; - Buried resist removal.

Table F.1 *Constituent roadmap at start project*

	Constituent Roadmap Scores at start project (alternative sensor)					
	Customer Attributes	***Functional Requirements***		***Design Parameters***		***Process Variables***
System		Limited to Measuring System & ADC (Energy Harvester, Energy Manager, and Integration not further elaborated): - Idea is to apply two Palladium nanowires on a single chip. One of the wires will be passivated to suppress Hydrogen diffusion and allows for temperature corrections. Electronic circuitry enables compensation and calibration.	1	Limited to Measuring System & ADC (PV Cells, Energy Management Circuitry, and Integration not further elaborated): - Measuring current will be applied on the Pd Nanowires; - ADC is applied to convert current to digital value; - State machine from ADC will be applied for numerical calculation.	1	Limited to Measuring System & ADC (PV Cells, Energy Management Circuitry, and Integration not further elaborated): - Adjust manufacturing of Nanowires to meet compliance with low-power character and realistic input currents of ADC.
Part		Limited to Measuring System & ADC: - One Nanowire temperature sensitive; - Second Nanowire temperature and Hydrogen sensitive; - Dual ADC allows for subtraction.	1	Limited to Measuring System & ADC: - One Nanowire isolated with passivation coating; - Dual ADC allows for subtraction.	1	Limited to Measuring System & ADC: - Passivation with glass coating or fluid acrylate dispensing; - Standard electronic circuitry for ADC, processing, and interfacing.

Table F.2 *Alternative sensor system at start project*

	Final Constituent Roadmap Scores for Conceptual Stage						
	Customer Attributes		***Functional Requirements***		***Design Parameters***		***Process Variables***
Project	- Foundations of Energy Harvesting at the nanoscale: Demonstration of radically new strategies for energy harvesting and local storage below the micrometre scale. Exploration and harnessing of potential energy sources at that scale including kinetic energy present in the form of random fluctuations, ambient electromagnetic radiation, chemical energy and others; - Self-powered autonomous Nanoscale electronic devices: Autonomous Nanoscale electronic devices that harvest energy from the environment, possibly combining multiple sources, and store it locally. These systems would coordinate low-power sensing, processing, actuation, communication and energy provision into autonomous wireless Nanosystems.	2	FR Develop a self-powered ICT system that is beyond state of the art.	2	DP Develop smallest PV powered sensor in the world.	2	PV Apply engineering methods systematically; PV' Apply a project team with proven credentials for the field of research.
Product	- Possibility of building autonomous Nanoscale devices (from sensor to actuators), extending the miniaturisation of autonomous devices beyond the level of the 'smart dust'; - New applications in a vast number of ICT fields such as intelligent distributed sensing, for health, safety-critical systems or environment monitoring.	2	*Constraint: Develop the smallest PV powered sensor in the world.* FR1 Harvest energy for operation from the environment; FR2 Manage the energy in the device; FR3 Quantify environmental chemical substance; FR4 Provide power and data to the RF frontend.	2	*Constraint: Apply engineering methods systematically.* DP1 Apply nanowire PV Cells; DP2 Include Energy Management circuitry; DP3 Apply sensor die to measure chemical substance; DP4 Include external interface.	2	*Constraint: Apply CMOS compatible production processes for all silicon dies.* PV1-3 Dies will have their specific CMOS manufacturing recipe and will be made in batches per kind; PV4 Interface with substrate and standardised wirebond process.
System			Limited to Measuring System & ADC (Energy Harvester, Energy Manager, and Integration not further elaborated): FR3.1 Determine temperature FR3.2 Determine function of temp and hydrogen concentration; FR3.3 Compensate concentration with temperature.	2	Limited to Measuring System & ADC (PV Cells, Energy Management Circuitry, and Integration not further elaborated): DP3.1 Apply setup with passivated Nanowire & ADC; DP3.2 Apply setup with Palladium Nanowire & ADC; DP3.3 Integrate subtraction in ADC.	2	Limited to Measuring System & ADC (PV Cells, Energy Management Circuitry, and Integration not further elaborated): PV3.1 Calibrate ADC-bias current to resistance of passivated Nanowire; PV3.2 Calibrate ADC-bias current to resistance of Pd Nanowire; PV3.3 Integrate state machine.
Part			Limited to Measuring System & ADC: FR3.1.1 Measure accurately; FR3.1.2 Measure with quick response; FR3.2.1 Measure Hydrogen in air selectively; FR3.2.2 Measure sensitively; FR3.2.3 Measure with quick response.	2	Limited to Measuring System & ADC: DP3.1.1 Characterise resistance; DP3.1.2 Low mass of passivated Nanowire; DP3.2.1 Intrinsic selectivity for H_2 by application of Palladium; DP3.2.2 Tune Nanowire thickness for sensitivity and response.	2	Limited to Measuring System & ADC: PV3.1.1 Software compensation; PV3.1.2 Thin passivation layer; PV3.2.1 Manage purity of Pd;; PV3.2.2 Sufficiently high resolution process.

Table F.3 *Constituent roadmap at finalisation of conceptual phase*

	Final Constituent Roadmap Scores for <u>Project</u>							
	Customer Attributes		***Functional Requirements***		***Design Parameters***		***Process Variables***
Project	- Foundations of Energy Harvesting at the nanoscale: Demonstration of radically new strategies for energy harvesting and local storage below the micrometre scale. Exploration and harnessing of potential energy sources at that scale including kinetic energy present in the form of random fluctuations, ambient electromagnetic radiation, chemical energy and others; - Self-powered autonomous Nanoscale electronic devices: Autonomous Nanoscale electronic devices that harvest energy from the environment, possibly combining multiple sources, and store it locally. These systems would coordinate low-power sensing, processing, actuation, communication and energy provision into autonomous wireless Nanosystems.	3	FR Develop a self-powered ICT system that is beyond state of the art.	3	DP Develop smallest PV powered sensor in the world.	3	PV Apply engineering methods systematically; PV' Apply a project team with proven credentials for the field of research.
Product	- Possibility of building autonomous Nanoscale devices (from sensor to actuators), extending the miniaturisation of autonomous devices beyond the level of the 'smart dust'; - New applications in a vast number of ICT fields such as intelligent distributed sensing, for health, safety-critical systems or environment monitoring.	3	*Constraint: Develop the smallest PV powered sensor in the world.* FR1 Harvest energy for operation from the environment; FR2 Manage the energy in the device; FR3 Quantify environmental chemical substance; FR4 Provide power and data to the RF frontend.	3	*Constraint: Apply engineering methods systematically.* DP1 Apply nanowire PV Cells; DP2 Include Energy Management circuitry; DP3 Apply sensor die to measure chemical substance; DP4 Include external interface.	3	*Constraint: Apply CMOS compatible production processes for all silicon dies.* PV1-3 Dies will have their specific CMOS manufacturing recipe and will be made in batches per kind; PV4 Interface with substrate and standardised wirebond process.
System			Limited to Measuring System & ADC (Energy Harvester, Energy Manager, and Integration not further elaborated): FR3.1 Determine temperature FR3.2 Determine function of temp and hydrogen concentration; FR3.3 Compensate concentration with temperature.	3	Limited to Measuring System & ADC (PV Cells, Energy Management Circuitry, and Integration not further elaborated): DP3.1 Apply setup with passivated Nanowire & ADC; DP3.2 Apply setup with Palladium Nanowire & ADC; DP3.3 Integrate subtraction in ADC.	3	Limited to Measuring System & ADC (PV Cells, Energy Management Circuitry, and Integration not further elaborated): PV3.1 Calibrate ADC-bias current to resistance of passivated Nanowire; PV3.2 Calibrate ADC-bias current to resistance of Pd Nanowire; PV3.3 Integrate state machine.
Part			Limited to Measuring System & ADC: FR3.1.1 Measure accurately; FR3.1.2 Measure with quick response; FR3.2.1 Measure Hydrogen in air selectively; FR3.2.2 Measure sensitively; FR3.2.3 Measure with quick response.	3	Limited to Measuring System & ADC: DP3.1.1 Characterise resistance; DP3.1.2 Low mass of passivated Nanowire; DP3.2.1 Intrinsic selectivity for H_2 by application of Palladium; DP3.2.2 Low mass of Nanowire; DP3.2.3 Low mass of Nanowire.	3	Limited to Measuring System & ADC: PV3.1.1 Software compensation; PV3.1.2 Thin passivation layer; PV3.2.1 Manage purity of Pd;; PV3.2.2 'Flying' Nanowires for better interfacing of medium; PV3.2.3 Nanowires in the 50 Nanometre range.

Table F.4 *Constituent roadmap at finalisation of robustness phase*