DTU Library

Designing Mechatronic Products - Achieving Integration by Means of Modelling Dependencies

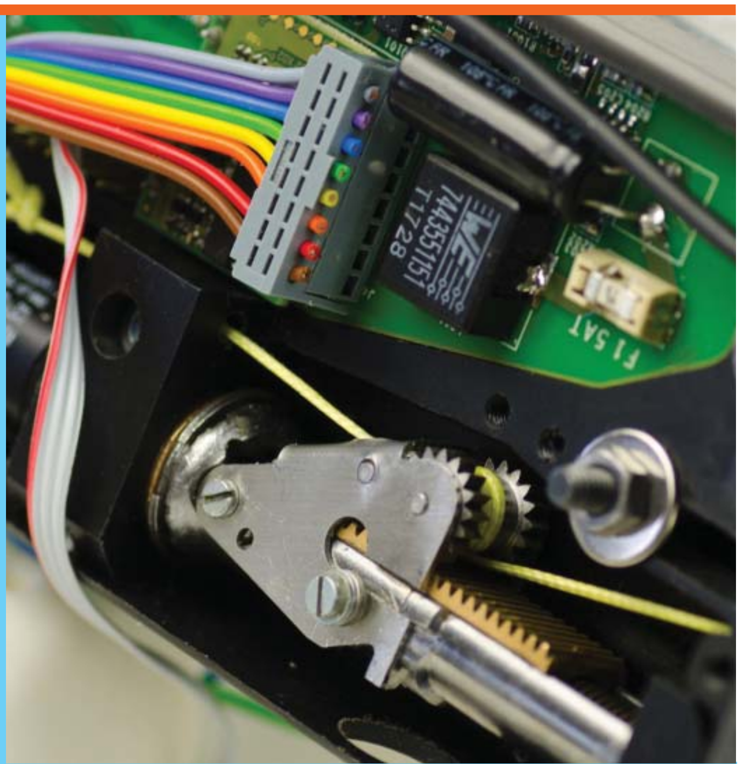
Torry-Smith, Jonas

Publication date:
2013

Document Version
Publisher's PDF, also known as Version of record

Designing Mechatronic Products - Achieving Integration by Means of Modelling Dependencies

PhD Thesis



Jonas Mørkeberg Torry-Smith
DCAMM Special Report No. S158
Februar 2013

DTU Mechanical Engineering
Department of Mechanical Engineering

Designing Mechatronic Products – Achieving Integration by Means of Modelling Dependencies

PhD thesis

Jonas Mørkeberg Torry-Smith

2013

DTU Mechanical Engineering
Department of Mechanical Engineering

Designing Mechatronic Products – Achieving Integration by Means of Modelling Dependencies
PhD thesis
Jonas Mørkeberg Torry-Smith
2013

ISBN: 978-87-7475-366-7

DTU Mechanical Engineering
Section of Engineering Design and Product Development
Technical University of Denmark (DTU)
Produktionstorvet, Bld 426
DK-2800 Kgs. Lyngby
Denmark

Phone (+45) 4525 6263
www.mek.dtu.dk

ABSTRACT

The research carried out in this PhD thesis focuses on the integration phenomenon in the design of mechatronic products. It contributes to the understanding of the phenomenon and to conceptual modelling of mechatronic products. These products are perceived to have emerged as a consequence of a joint effort between mechanical, electronics and software development.

Companies engaged in multi-disciplinary product development face the challenge of coordinating different engineering disciplines to pursue a synergistic effect and to avoid failures when solutions from the different engineering disciplines have to be combined and function in an integrated fashion. The holistic view of the product concept is sometimes missing in projects and engineering disciplines are reluctant to interact.

Investigations of literature, cases from industry and practical experience are used for dissecting the integration phenomenon. This reveals a vast number of challenges, which companies experience in relation to the phenomenon, ranging from product-level challenges to organization-level challenges.

A comprehensive literature study is carried out with the aims of finding proposed solutions for addressing the challenges related to the integration phenomenon and evaluating how well the challenges are addressed by using the solutions in a design setting. The study shows that the available solutions only partly cover the stated challenges and that a large part of the identified solutions appear to support analysis activities rather than synthesis activities. The study further shows that functional modelling can be used to create a common model in the early phases of design, but later phases have to rely on informal and formal transformation of information about the product concept between the domains due to the absence of a common modelling language.

The further research is aimed at one particular challenge related to the integration phenomenon, which is the difficulty in modelling and controlling multiple relations in the product concept as a consequence of the multi-disciplinary development. 'A product-related dependency' is defined to describe relations in the product concept between *functions*, *properties* and *means* (solutions) appearing as a consequence of the design process. Based on three cases from industry a classification of dependencies is established, highlighting thirteen types of generic dependencies to be aware of when designing mechatronic products. The literature study shows that such a classification has not been attempted before. Without a classification to provide a basis for a structured search it is left to chance for the development team to discover potentially critical dependencies in due time.

A 'Mechatronic Integration Concept' is proposed aiming at operationalizing the use of the classification. The purpose is to facilitate discussions between engineers from different disciplines on matters that are important to clarify between the designers working on contributing with solutions to the product concept. Hence, it is not meant to be a conceptual description of every important aspect of the product concept. The classification and the Mechatronic Integration Concept are evaluated by deploying them in a mechatronic project from industry. The results from the evaluation show that the thirteen groups of dependencies can be identified, clarified and modelled by use of the classification and the Mechatronic Integration Concept. The results also indicate that rework can be avoided, lead-time can be shortened and the performance of the product increased by utilizing them in industrial projects.

Resumé (In Danish)

Forskningen som ligger til grund for denne phd-afhandling omhandler integrationsfænomenet ved udvikling af mekatroniske produkter. Den bidrager til fænomenforståelsen og til konceptmodellering af mekatroniske produkter. Disse produkter opfattes som værende fremkommet ved en koordineret indsats mellem mekanik-, elektronik- og softwareudvikling.

Virksomheder som er involveret i multidisciplinær produktudvikling står over for udfordringen om at skulle koordinere forskellige ingeniørdiscipliner med det formål at opnå en synergetisk effekt og samtidigt undgå faldgrubber når løsninger fra forskellige ingeniørdiscipliner skal kombineres og fungere som en integreret helhed. Den holistiske synsvinkel på et produktkoncept er til tider ikke eksisterende i projekter og ingeniører fra forskellige discipliner er tilsyneladende tilbageholdende med at interagere med hinanden.

Litteraturundersøgelser, industrieksempler og praktisk erfaring benyttes til at granske integrationsfænomenet. Dette fører til en afdækning af en række udfordringer som virksomheder oplever i relation til fænomenet, som opleves fra produktniveau til det virksomhedsorganisatoriske niveau.

En omfattende litteraturundersøgelse foretages med det formål at afdække eksisterende løsninger som adresserer udfordringerne relateret til integrationsfænomenet, og med det formål at evaluere hvilken tilstrækkelighed hvormed udfordringerne er adresseret af de anførte løsninger i den konkrete udviklingssituation. Undersøgelsen påpeger at de eksisterende løsninger kun delvist adresserer de anførte udfordringer og at løsningerne i vid udstrækning understøtter analyseaktiviteter frem for synteseaktiviteter. Undersøgelsen viser endvidere at funktionsmodellering kan benyttes som fælles modelleringssprog i de tidlige faser i udviklingsprocessen, men at senere faser må bero på formelle og uformelle transformationer af informationer om produktkonceptet mellem domænerne forårsaget af mangel på et fælles modelleringssprog.

Den videre forskning er målrettet mod en af de tidligere afdækkede udfordringer som er relateret til integrationsfænomenet. Denne udfordring dækker problematikken ved at modellere og kontrollere multiple relationer som opstår i produktkonceptet som følge af den multidisciplinære udviklingsform. En 'produkt-relateret afhængighed' defineres dækkende over relationer i et produktkoncept mellem *funktioner*, *egenskaber* og *midler* (løsninger) som en konsekvens af udviklingsprocessen. Baseret på tre cases fra industrien etableres en klassifikation som indeholder 13 generiske typer af afhængigheder. Litteraturundersøgelsen viser at en sådan klassifikation ikke er forsøgt opnået tidligere. Uden en klassifikation som kan danne grundlæget for en struktureret afdækning af afhængigheder må udviklingsteamet forlade sig på at disse potentielt kritiske afhængigheder opdages i tide på anden vis.

Et 'Mekatronisk Integrationskoncept' opstilles med det formål at operationalisere brugen af klassifikationen af afhængigheder. Endvidere er formålet at iscenesætte en diskussion blandt ingeniørerne fra de forskellige discipliner på områder som er vigtige at afklare mellem de udviklingsingeniører som bidrager med løsninger til det samlede produktkoncept. Således har det ikke været intentionen af skabe en holistisk konceptbeskrivelse, hvori alle relevante aspekter er medtaget. Klassifikationen og det Mekatroniske Integrationskoncept bliver evalueret via en case fra industrien. Resultatet af evalueringen viser at de 13 typer af afhængigheder kan identificeres, afklares og modelleres ved brug af klassificeringen og det Mekatroniske Integrationskoncept. Resultatet indikerer endvidere at oprettende arbejde som følge af integrationsproblemer kan reduceres, projektgennemløbstiden forkortes og produktperformance kan øges i projekter i industrien.

PREFACE

This PhD thesis documents the outcome of three years of research carried out at the Department of Mechanical Engineering at the Technical University of Denmark with Professor Niels Henrik Mortensen as supervisor. The PhD project has partly been sponsored by the consultancy company IPU-Product Development and partly by DTU. The project was initiated in January 2009 and ended in February 2013. The project was interrupted by leaves of absence due to consultancy work and due to parental leave.

Looking back I can now see that the seed for pursuing a PhD was planted around eleven years ago. At that time I did my masters within product development at DTU with Professor Mogens Myrup Andreasen as supervisor. His deep and profound knowledge and his razor-sharp ability to see through complex problems did and has always struck me with awe. After my masters I pursued a career within the consultancy business, which led to many invaluable insights into design aspects of product development. Yet, whenever I got the chance to talk with Professor Andreasen, I got the feeling that there were some hidden layers describing the nature of product development, which I did not have access to. This feeling grew in me and after six years, a PhD project seemed very much appealing to me. At that time I was fortunate that the consultancy company I worked for, IPU-Product Development, offered to support such a research project financially. For that I am deeply grateful. DTU also supported the PhD financially for which I am equally grateful.

First of all I would like to thank Professor Niels Henrik Mortensen for being my supervisor and for showing confidence in my research. Thanks for the many inspiring, constructive and insightful discussions. I have very much appreciated the collaboration and the guidance provided.

Additionally, I would like to express my gratitude to Professor Sofiane Achiche, who enthusiastically and altruistically took part in my research. I am truly thankful for your energetic involvement and support throughout the project. You have given me invaluable insights to research and your moral support has been invaluable during the hard times in the project. It has been a privilege to work with you.

I wish to thank the colleagues at IPU for interesting and valuable discussions and for providing opportunities to test the research results in industrial projects. Especially, I would like to thank CEO, Lars Hein and Division Manager, Thorkild Ahm for acting as an advisory board and for pushing the limits to ensure the topicality of my research. Thanks, Jesper Windum, for encouraging me to take on the PhD project!

I wish to thank Ahsan Qamar, my fellow researcher at KTH, Sweden, for many fruitful discussions and expanding my horizon towards SysML and Model-Based Systems Engineering. The hard work on the papers paid off and it has been fun working with you. I would also like to thank the other co-authors for contributing to the various papers: Professor Niels Henrik Mortensen, Professor Sofiane Achiche, Professor Jan Wikander, Director Carl During and R&D Manager Ole Plough.

I express my gratitude to my good colleagues at the section for sharing their insights and for their willingness to discuss my research. Professor Mogens Myrup Andreasen, thanks for your 'critical enthusiasm' pinpointing the weaknesses in my research thus pushing me further.

I would like to express my gratitude to R&D Manager Ole Plough for always being willing to engage in discussions and for sharing his expertise and experience within the industrial application of the research I was engaged in.

I wish to thank my fellow PhD students at the K&P Section at MEK/DTU for sharing the ups and downs in the course of the research. I wish to thank each and every one of you for the many constructive discussions on my research topic and for making it fun and joyful to show up for another day of research.

I wish to thank all the companies and their representatives who willingly shared their experience and development material for me to conduct the research. Thanks to Flowsion, Invencon, Linde Werdelin, Rose Technology, IPU-Product Development, I/O technologies, Oticon.

To those who in any way have contributed to the PhD but have not been mentioned above; consider this as my sincere thanks for your contribution, help and support.

Finally I would like to express my deepest thankfulness to my wife Hanne for her love and moral support. I deeply admire you for your endless patience with your husband. And to you Liva and Aksel, our children, for the joy and daily smiles turning the darkest moments of a PhD project into sunny days.

Jonas Mørkeberg Torry-Smith

Lyngby, February 2013

TABLE OF CONTENTS

1 INTRODUCTION

1.1 BACKGROUND/PROBLEM AREA

As the cost of electronics and microchips has been decreasing over the last decades the introduction of electronics and software in mechanical products have increased. Today, there are a continuum of products ranging from mechanical products with added functionality realised by electronics and software to electronic products with embedded software with the need of mechanical encapsulation. The multi-technological products pose a fundamental challenge for companies joining development processes, different in nature, into a synergistic process to produce successful products. While some companies see new market potentials in the joining of solutions from several engineering disciplines, other companies sees it as a necessity to stay competitive on already captured market segments. Prior to this research project I worked as a product development consultant for Danish and international companies for 6 years. My personal experience is that both successful companies and those struggling to break-even on the development investments, all struggle with multi-disciplinary issues in the development. Due to the globalization and thus greater possibility for outsourcing and off-shoring, both the development and the production and the need for staying competitive in terms of driving costs down and performance up with even shorter lead-times is immense. The added complexity caused by going from single-disciplinary development to multi-disciplinary development can have various consequences, bringing potential danger to the project. Some of them include a significant increase of the development costs compared to those originally budgeted and much longer lead-times than scheduled. The problems caused by the multi-disciplinary aspect of product development are many and the *means* for addressing the problems are equally diverse and not always effective or easy to implement. A case study in Finnish industry revealed the following problems faced by companies, describing the diversity of the problem area caused by multi-disciplinarity (Salminen and Verho 1989):

- Lack of common language between expert groups.
- Risk of clique formation among expert groups.
- Understanding the totality over and beyond disciplinary border lines, lack of total view.
- Definition of responsibilities not clear enough, risk of unreliable and poor quality interface between the various techniques.

A more recent report on some of the challenges related to the phenomenon of integration in mechatronic development shows that the integration is still a source for problems in mechatronic development (Cabrera et al. 2010). The stated challenges are specifically directed at modelling the product concept.

Difficulties in developing mechatronic products relate to:

- Exchange of design models and data.
- Cooperative work and communication among the design engineers.
- Multi-disciplinary modelling.
- Simultaneous consideration of designs from different disciplines.
- Early testing and verification.
- Persistence of a sequential design process.
- Lack of tools and methods supporting multi-disciplinary design.
- Support of the design of control software.

While some of the described problems can be related to organizational issues and generic collaborative issues between engineering disciplines, other problems relate to describing the product concept in a multi-disciplinary development setting. Andreasen et al. (1989) refer to 7 dimensions of the ‘development function’ in companies, which are: the organisation structure, the decision structure, the social system, the methods and tools, the knowledge structure, the measuring structure and the physical surroundings. Within each of these dimensions a company may strive to reach a desired level of integration between the engineering disciplines. In the first part of the thesis a wide scope is applied to shed light on which type of integration problems companies encounter and the measures used to address them. When the engineering disciplines interact, relations appear which can be of benefit in terms of synergistic possibilities as well as constrictions to agree upon and resolve in the course of the development. This area is the focus of the last part of the thesis in which the specific problem of modelling and controlling multiple relations in the product concept is investigated. The risk of failure for a design team increases if these relations are not carefully considered. If integration tests fail, the team has to spend time fixing the problems, instead of spending time on the value-adding activities. Advantages of controlling the relations include the potential to bring down the lead-time caused by avoidance of re-work. From a broader perspective, the controlling of the relations is a prerequisite for obtaining a concurrent process of activities within the different engineering disciplines harvesting effects such as even shorter lead-times and better utilization of synergies in solution finding.

In section 1.2 the aim and scope of the research are described and in section 1.3 the research questions are stated. Thus, sections 1.2 and 1.3 will proceed with further descriptions of the delimitation of the research allocated to this PhD project.

1.2 AIM AND SCOPE FOR THE THESIS

Research into development of mechatronic systems continuously states a demand for fruitful and seamless integration between the involved engineering disciplines (Gausemeier et al. 2008; Tomiyama et al. 2007; Andreasen and McAloone 2001) to create success in the development of mechatronic products. Many research angles can be applied to the phenomenon of integration. One example could be to apply general collaboration theories between non-specific disciplines. However, the attention in this research is directed towards the product being developed and the synthesis process in which the product is being created. The focus is on the synthesis aspects of designing, which delimits the research by de-emphasising areas such as requirements management, documentation management as well as business strategy management considerations. It is assumed that these competences, not directly related to the integration phenomenon, are established to a satisfactory level, thus not negatively influencing the integration phenomenon. In the first part of the research the scope is on the integration phenomenon in the design process, which can be considered as a rather wide scope. In the last part of the research the focus is narrowed down to relations, which appear in the product concept which cause integration problems in the design process. When viewing the integration phenomenon it is assumed that the competence of performing development within the separate engineering disciplines is acquired to a satisfactory level.

The following engineering disciplines are considered to comprise mechatronic design: mechanical engineering, electronics engineering and software engineering. For some researchers, control engineering is perceived to be at the very centre of research in mechatronics (e.g. Gausemeier et al. (2009a); Welp and Jansen (2004); Isermann (2005)). Thus, control engineering might be considered as an engineering discipline on the same level as mechanical, electronics and software engineering. This viewpoint is typically observed for researchers within control engineering. However, in this thesis control engineering is regarded as one of many competences needed when developing mechatronic products. Thus, control engineering should not be regarded at the same level as mechanical engineering, electronics engineering and software engineering. The competence can be located in any of the three engineering disciplines. Researchers working within the field of control engineering also work on the aspect of how the mechatronic designs can be described and manipulated across engineering disciplines. Thus, this research field can be used as a source for inspiration and comparison. Often the objective within this research is to ensure the best performance in relation to analysing and manipulating control parameters such as accuracy and speed of moving parts or embedding artificial intelligence into the control strategy. Thus the research is aimed at mechatronic products including but not limited to products that include control engineering considerations.

My background is within mechanical engineering and the research has been performed in a department at the Technical University of Denmark concerned with mechanical engineering. Although the areas of mechanical, electronics and software engineering have been treated with equal interest, the research is influenced by the mechanical engineering discipline with respect to aspects listed in the following paragraph. Firstly, the research methodologies applied have been adopted from the mechanical research community. Still, the methodologies have a high degree of universality (Blessing and Chakrabarti 2009) and thus embrace research within product development comprising other technical disciplines; hereunder electronics and software. Secondly, a requirement has been made that the mechanical engineering effort in the investigated projects must at least be 'comparable' to the effort within electronics engineering and software engineering. This serves as a sound delimitation of the research scope, since projects primarily driven by software and electronics will be different in nature and further away from my competences or those expected to be acquired during the PhD project. Thirdly, the theoretical framework used to describe technical systems has been adopted from the mechanical engineering domain. However, research on mechatronics has concluded that the theoretical framework can be applied to mechatronic products (Buur 1990).

Summing up, the aim of the research is to investigate and contribute to the integration phenomenon in designing mechatronics. The research scope focuses on mechanical engineering, electronics engineering and software engineering, while requiring the mechanical design effort to be considerable compared to electronics and software. In this context the first two research questions are formulated. By applying further delimitations two more research questions are directed at relations, which appear in the product concept which cause integration problems in the design process.

1.3 Research questions

The first research question is aimed at investigating the reasons for the integration phenomenon problems. For example, the nature of the design process is different within the domains causing problems in coordinating activities and deliverables between the domains. A vast number of challenges related to the integration phenomenon have been stated in the literature by various researchers; but the question is, are some of the challenges more significant or important than others? The first research question relates to this problem and is formulated as follows:

> ***RQ1: What are the central integration phenomena posing a challenge to companies when developing mechatronic products?***

In order to answer this question a structured process is formed around a literature review to claim its generality and a case study is performed to support the validity of the results. It is then natural to subsequently ask about the available solutions capable of addressing the integration phenomena. The second research question is hence formed around this problem.

> ***RQ2: What solutions in terms of methods, tools and mind-sets exist which can facilitate integration between the involved engineering disciplines in the development of mechatronic products?***

It is expected that answering the first two research questions will provide knowledge about the integration phenomena and that conclusions can be drawn about the prospect of providing solutions to the identified phenomena.

When we turn to mechatronic development instead of single-disciplinary development we face a trade-off. The multi-disciplinary development should provide new opportunities while we commit to fit the solution in each domain to obtain a required performance of the product. These types of constrictions will be labelled *dependencies* in this thesis. A clear definition of 'a dependency' has not been established within the mechatronics literature or within affiliated literature on topics such as Dependency Structure Matrix, Complexity Management etc. Thus, a definition of 'a dependency' has been created to be used for the research presented in this thesis. The definition builds on concepts defined in the ToD framework (Hansen and Andreasen 2002) and is presented in the following.

> ***A dependency is a relation appearing in the product as a consequence of the design process between the following product attributes: functions, properties and means.***

Functions and *properties* constitute the behaviour of the product and the *means* are the solutions which realises the behaviour. When the relation is viewed in a design context it represents a dependency. For example: solutions created by the different engineering disciplines will provide the functionality in the product. If a solution within one domain is changed the solutions within the other domains might have to be changed accordingly to realise the desired functionality. Dependencies can also appear between e.g. activities in a development project, as stated by Danilovic and Browning (2007). However, other types of dependencies other than the dependencies described by the above definition are not the object of study in this research.

To create a product the designers have to establish *functions*, create solutions and realise desired *properties*. *Functions* will be allocated to the different engineering disciplines and as a consequence the solutions are created within each of the disciplines, which have to come together to form the final product. Because there are a vast number of relations between *functions, properties and means* in a product, dependencies in the product concept will appear as a consequence of the design process where the involved engineering disciplines collaborate. If the dependencies are ignored then successful integration will be less likely. On the other side, if we are able to be systematic in handling the dependencies it will enable us to facilitate the integration between the engineering disciplines (Haskins and Forsberg 2011) [p14 and p18]. Aiming at providing a systematic approach in viewing dependencies in the product concepts the third research question is formulated as follows:

> ***RQ3: What classification can be identified for dependencies appearing in a mechatronic product concept?***

The fourth research question is formulated with the incentive of creating support for controlling and manipulating the dependencies in a mechatronic development project. The assumption is that a higher level of integration will be reached if we are able to model the dependencies in a way that engineers can come together at integration meetings and reveal, discuss and decide on these dependencies. Thus the fourth research question is formulated as follows:

> ***RQ4: How can a classification of dependencies be used as a basis for modelling and describing dependencies in a mechatronic product concept?***

Due to the formulation of the research questions and the definition of *a dependency,* two *research objects* can be defined. Due to the strong focus on the mechatronic product the *core product* is one of the research objects. The second research object is the design process. In section 2.1 the chosen research methodology is described for supporting the research process with the aim of answering the stated research questions.

1.4 Definition of key terms used

Certain terms will be used for describing the product and the design process. For clarity, these are defined in the following:

Domain: A domain is equivalent to an engineering discipline. A multi-domain product would therefore refer to a product which has been created in a joint effort between several engineering disciplines. A domain-specific component will refer to a component that one engineering discipline is responsible for developing. The terms domain and engineering discipline will be used interchangeably.

Multi-disciplinary: The term 'multi-disciplinary' relates to a 'design effort'. A design effort is considered as multi-disciplinary when more than one engineering discipline is involved in the effort of developing a product.

Multi-technological: The term 'multi-technological' relates to a 'product'. A multi-technological product is a product comprising components developed by different engineering disciplines.

Mechatronic product: A product that comprises components developed by mechanical, electronics and software engineers.

Dependency: In one of the papers presented in this thesis (Paper D) the term 'product-related dependency' is used which is equivalent to the definition on page 4. Even though product-related dependency might be more descriptive, the use of the short version 'dependency' was chosen for ease of reading.

A design: A design is equivalent to a product concept, which is a representation in a development project of the intended product.

Integration: Integration in the design process refers to the required coordinated collaboration between engineers from different engineering disciplines in order to create a design which fulfils the requirements in the specification.

1.5 Outline of the thesis

The introduction (section 1) contains the background for the research along with the aim, scope and research questions. Then the research approach is discussed (section 2). It contains the chosen research methodologies and the plan for the research. Section 3 provides an overview of design theories used to form the basis for this research. In the subsequent section (section 4), experience from industry in developing mechatronic products and related problem areas are described. Section 5 contains the results from the research. The results are presented via five scientific papers labelled A, B, C, D and E. The conclusion is found in section 6, in which the research questions are answered. In the same section the research is evaluated and implications on industry are discussed. The thesis is ended by proposing directions for future research (section 7) and then rounded-up by concluding remarks (section 8). References are listed in section 9. Appendices are found in section 10 and section 11 contains the scientific papers produced in this PhD project.

2 RESEARCH APPROACH

2.1 RESEARCH METHODOLOGY

The type of research performed in this PhD can be categorised as design science. Design science differs from natural science on the aim of improving design practice, and thus does not have a purpose without its applicability in industry (Andreasen 2009). Several models have been proposed describing design research methodology, e.g. Jørgensen (1992), Blessing and Chakrabarti (2009), Andreasen (2009). Yet, no one model has been accepted as the de facto standard to be used within design research. This leads to the question of choosing and applying an appropriate research methodology to guide the research.

The outline of the design research methodology (DRM) proposed by Blessing and Chakrabarti (2009) is illustrated in **Fig. 1**. The main concept of the DRM framework is to view the research as progressing through distinct research stages each with assigned objectives. The Research Clarification is used for finding an indication or evidence for the initial assumption, which most research projects start out with. The aim of the stage is to define the goals of the research and to determine the criteria against which the research can be evaluated. In the following stage, the Descriptive study I, the researcher, am elaborating the understanding of the current situation and the influencing factors which can positively affect the problem area of interest. Having decided what influencing factors to target, the Prescriptive Study is used for synthesising *support* which will improve the situation. In the Descriptive Study II *the support* is tested to validate to what extent *the support* has had the intended effect. The DRM framework is used as guidance for the research in this PhD project because it provides a sound pattern for the sequence of clarifications in terms of goals, understanding, support and evaluation (see **Fig. 1**).

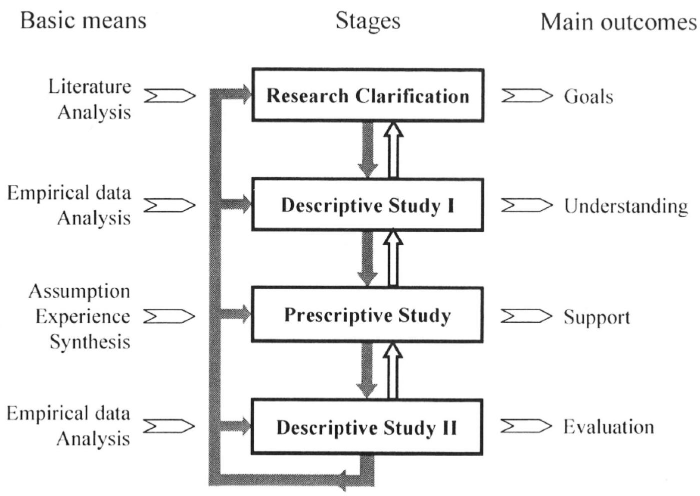


Fig. 1 DRM framework by Blessing and Chakrabarti (2009)

Another design research methodology highly applicable to this research is presented by Jørgensen (1992). It suggests that design research will either be *Problem-based,* addressing problems conceived in the design practice or *Theory-based* addressing a set of existing theories (**Fig. 2**). Therefore, research may be approached with two different methods, i.e. starting by either analysing or synthesising as shown in **Fig. 2**. However, in practice, most research projects will involve both paradigms to various degrees (Jensen 1999). Although the model by Jørgensen (see **Fig. 2**) has not been followed step-by-step, it highlights the two types of entries to design research, which is also used in this PhD project. The *Problem-based* entry to the research in mechatronics is composed of case studies and observations from industry. The *Theory-based* entry is based on using systems theory and theories on design processes.

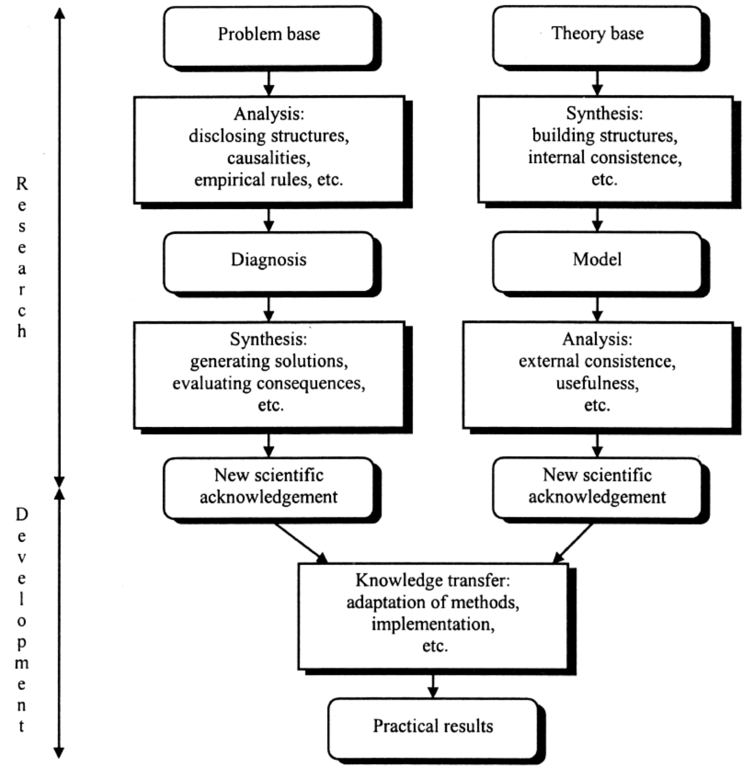


Fig. 2 Model of design research, in which two research approaches can be followed, a Problem-based and a Theory-based approach, after Jørgensen (1992)

The presented research frameworks by Blessing, Chakrabarti, and Jørgensen belong to design science, with roots going back to research communities formed around product development of mechanical products. Thus, it could be anticipated that the research frameworks will have a stronger affiliation with mechanical engineering than electronics or software engineering, leading to the questioning of the applicability of the frameworks for research into design of mechatronic products. However, the DRM framework is considered universal and domain independent (Blessing and Chakrabarti 2009). In addition, the framework by Jørgensen is also considered to be universal and domain-independent; since nothing indicates that it can only be applied within mechanical engineering research. Blessing and Chakrabarti (2009) describe DRM as focusing on "supporting engineering and industrial design research…". Therefore, the described research frameworks are considered to be adequate and valuable for the scope of this research within the design of mechatronic products. A description of the research plan is presented in section 2.3 relating the research performed in this PhD project to the stages of the DRM model (**Fig. 1**).

2.2 Verification of results

Verification of scientific results in design science poses a general challenge to researchers within the design science research field (Buur 1990). Due to the human involvement in the design process the verification of results is basically different from natural sciences, where a prediction based on the scientific contribution typically can be falsified or verified. The research contribution in this PhD project is evaluated by applying and testing it in a design context from industry. The research performed to answer Research Questions 1 and 2 is evaluated based on: *generality*, *validity* and *completeness*. Research performed in connection with answering Research Questions 3 and 4 will be evaluated based on the DRM methodology by Blessing and Chakrabarti (2009). They recommend evaluating research contributions based on: *usability, applicability* and *usefulness*.

- Usability: the ease with which the method can be used for the intended task;
- Applicability: whether the method has the direct intended effect on the design process;
- Usefulness: Whether the introduction of the method leads to an overall success of the project measured by a number of parameters taking into account possible uncontrollable influencing factors.

Note: In the description above 'a method' covers any type of support developed and tested in the research project.

The *usability, applicability* and *usefulness* aspects cannot be applied to the research formed around Research Questions 1 and 2 since the result of this research is not a method. Hence, the aspects *generality*, *validity* and *completeness* are used for evaluating this research. The evaluation of the research performed is described in section 6.3.

2.3 RESEARCH PLAN

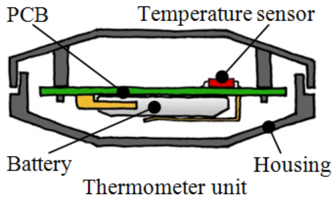
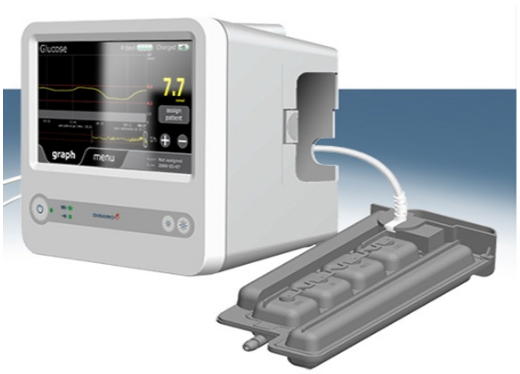
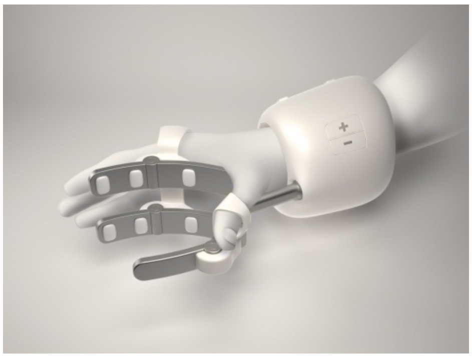
This section presents the research plan for the PhD project and how it was executed.

The master plan for the research was to initiate a literature study and a case study to gain an in-depth understanding of existing challenges and solutions related to the integration phenomenon within mechatronics development. Then, by utilizing the gained insight, a specific area would be targeted for proposing support to the design process. To ensure that the research would cover relevant industrial needs, the aim was to perform tests of the proposed support in industrial settings. Case studies have been applied to a large extent in the research to infuse 'real life' examples, issues and dilemmas appearing in design practice. A more detailed description of the research relating it to the DRM stages is stated below as well as illustrated in **Table 1**. **Fig. 3** illustrates which DRM research stages were used for answering the four research questions.

Table 1 Overview of activities performed in the various research stages of the DRM framework. The industrial projects mentioned in the table are briefly described in **Table 2**.

Stages according to DRM	Activities in the PhD project			
	Literature	Case studies of industrial projects	Verification of results	Reporting
RC	Review	'Linde Werdelin'		Paper A (conference)
DS-I	Comprehensive review		Simulation by use of previously performed project: Linde Werdelin	Paper B (conference) Paper C (journal)
PS-I	Review	'Diramo' 'Cooling module' 'Linde Werdelin'		Paper D (journal)
DS-II (initial)			Test in industrial project: 'DSH' (3 months)	Paper D (journal)
PS-II	Review	'AV products' 'Diramo' 'Linde Werdelin'		Paper E (journal)
DS-III (initial)			Test in industrial project: 'DSH' (2 months)	Paper E (journal)

Table 2 Overview of projects used for analysis and verification of results in the PhD project

Name of project	Picture of product	Short description of project
Linde Werdelin	 	External temperature unit, which is a part of a watch system for out-door sports
Diramo		Continuous blood sugar measurement device for intensive care units at hospitals
Cooling module		Cooling module for vending machines
AV products	No picture due to respect of the confidentiality of the projects	Audio and audio/visual products
DSH		Actuated hand for arthritis patients

Research Clarification and Descriptive Study I:

Literature was investigated in terms of what mechatronics phenomena had been identified and reported on to gain an initial understanding of prior research within the area. Then a case study was performed to reveal the design practice in a mechatronics project and understand the need for corporation between the involved engineering disciplines. This research was reported on in Paper A (see section 11.1). In Descriptive Study I a more thorough literature search was performed which included data processing in software to reveal the most cited researchers within the field of mechatronics research. The purpose was to compile a list of challenges as well as the proposed solutions originating from the most prominent researchers within the field. This research was reported on in Paper B and C (see sections 11.2 and 11.3). Paper C is an extension of the conference Paper B into a journal paper for the Journal of Mechanical Design. The first two stages in the research provided the basis for answering Research Questions 1 and 2. As a consequence of the performed research a need as well as a gap in the literature was identified regarding management of dependencies in mechatronic products. Thus, Research Questions 3 and 4 could be formulated.

Prescriptive Study-I and Descriptive Study-II:

A prescriptive study was performed with the aim of classifying dependencies appearing in mechatronic products when designing. The aim of this part is to make the management of dependencies more tangible and thereby facilitate better integration between the domains. Three case studies comprising mechatronic products were used to establish the classification and Descriptive Study II was undertaken to validate the found classes of dependencies. In Descriptive Study II an industrial project was monitored for approximately three months. The findings were reported on in a paper (Paper D, see section 11.4) addressing Research Question 3.

Prescriptive Study-II and Descriptive Study-III:

Prescriptive Study II was initiated with the aim of further operationalizing the use of the classification of dependencies. Observations of the need for conceptual descriptions in industrial projects were combined with a synthesis on how to represent the dependencies within this framework obtained from the observations. The proposal was then tested in Descriptive Study III in an industrial development project, where the dependencies were modelled over a period of approximately two months. The results formed the basis for the last paper (Paper E, see section 11.5).

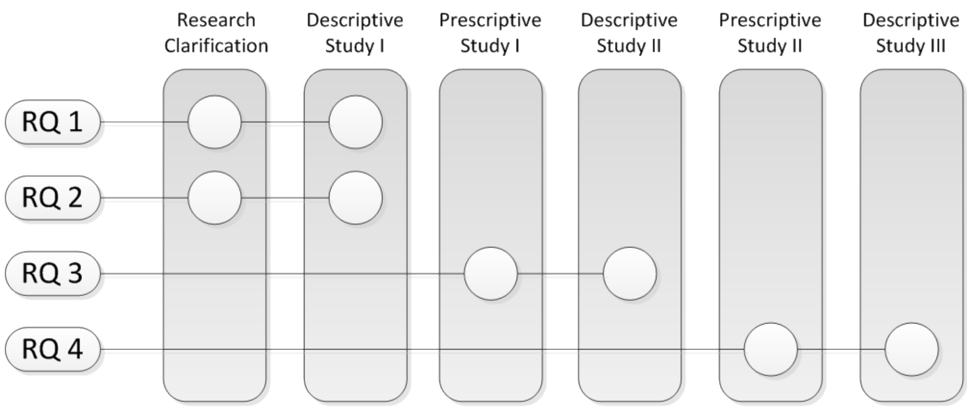


Fig. 3 The research stages used for answering the four research questions

3 THEORETICAL BASIS

The purpose of this section is to describe design theories, which constitute or contribute to the theoretical foundation used for the research presented in this PhD project. Although many other alternative design theories may exist, they will not be described. The mechatronics angle to product development poses a special challenge in choosing the theoretical basis which provides the necessary framework for describing a technical system and the design process. Should design theories from one domain be extended to cover the other two domains assuming adequate coverage? Doing so may limit what is possible and what might be considered adequate to describe e.g. a 'product model' seen from the designers' point of view. Aspects important to the other domains might simply be omitted using a design theory originating from one domain only. Joining design theories from three domains into a common design theory presents a great challenge due to e.g. obstacles in terminology and concepts and thus does not fall within the scope of this research. The solution used for this research in terms of adequate design theories has an analogy to Systems Engineering, though Systems Engineering is not claimed to be a design theory (Haskins and Forsberg 2011). In Systems Engineering as presented in Haskins and Forsberg (2011) only concepts familiar to the involved domains (including mechanical, electronics and software engineering) are used. This strategy will be applied to the design theories selected which will constitute the conceptual framework used in this research to describe the product and the design process. Thus, in this section argumentation will be provided for terms and concepts to be used suited for describing the technical system and the design process across the domains. First, design theories originating from mechanical engineering will be presented. Then this framework will be compared to frameworks from electronics and software to define the terms and concepts applicable across domains. This is the approach chosen which is the alternative to choosing one design theory from one of the domains causing the terms and concepts used to be highly influenced by that one domain. The following theories and design procedures will be presented:

Rooted in the mechanical engineering discipline

- Theory of Technical Systems (TTS) (Hubka and Eder 1988)
- Theory of Domains (ToD) (Andreasen 1980)
- Function/Means Law (Hubka 1967)
- Theory of Dispositions (TD) (Andreasen and Olesen 1990)
- Integrated Product Development (IPD) (Andreasen and Hein 1987)

Rooted in the software engineering discipline (short descriptions used for comparison)

- Systems Engineering, including Waterfall Model, V Model and Spiral Model (Haskins and Forsberg 2011; Blanchard and Fabrycky 1998; Sage and Rouse 2009)
- Agile development, including SCRUM and RAD (CMS 2007)

Rooted in the electronics engineering discipline (short descriptions used for comparison)

- Procedure suggested by Williams (1991)
- Procedure suggested by Jones (2004)

3.1 DESIGN THEORIES

3.1.1 THEORY OF TECHNICAL SYSTEMS

Theory of Technical Systems (TTS) is renowned as a fundamental and important contribution to the design research community. It builds on elements from systems theory (Klir and Valach 1967) and on cybernetics (Ashby 1956). A system is regarded as a 'mental construct' and thus an abstraction used to describe and model a specific area of interest. In **Fig. 4** Hubka and Eder (1988) illustrate a hierarchy of systems, relating technical systems to other types of systems.

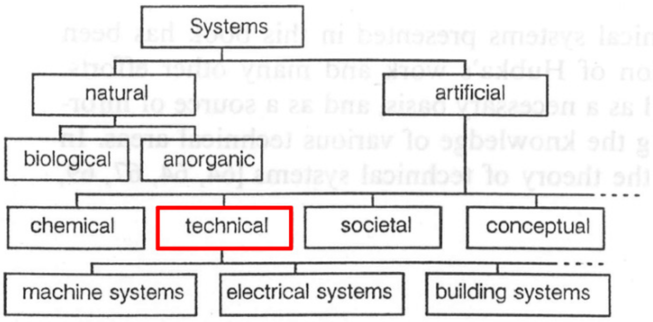


Fig. 4 Hierarchy of systems, adapted from Hubka and Eder (1988)

A system consists of elements. A boundary (the System Borderline) delimits the elements, which are a part of the system. Outside the System borderline elements are located, which are adjacent to the system (see Fig. 5). Relations exist between the elements, which will define the structure of the system. A system is considered recursive, thus enabling a system to be an element of a larger system. This inherently implies that a system might be decomposed into sub-systems.

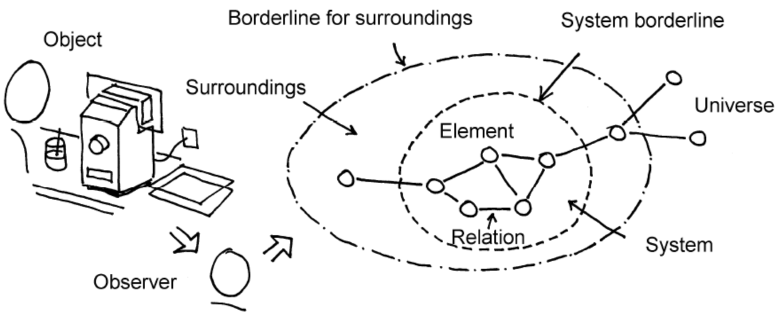


Fig. 5 Illustration of a technical system with elements and relations, after Andreasen (2005)

Central to TTS is the transformation system in which operands are transformed in a transformation process. The operands may be a biological object, material, energy or information. In the transformation process the technical system delivers an effect. In addition the transformation process may be aided by a human system as well as an information system and a management & goal system as illustrated in Fig. 6.

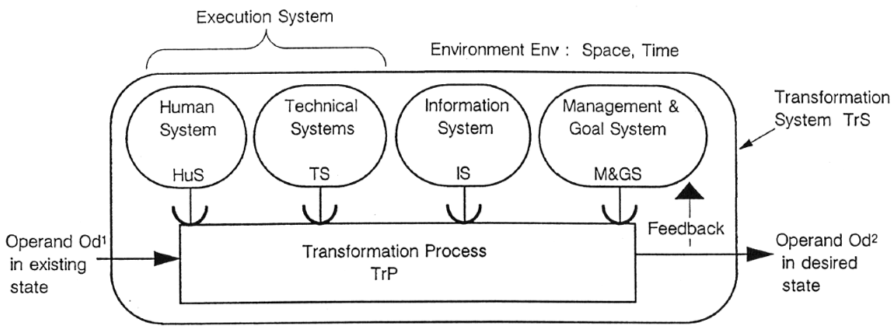


Fig. 6 Model of the transformation system. Adapted from Hubka and Eder (1988)

The process performed by the technical system creating the desired effect is called a technical process. A technical process consists of a sequence of processes each bringing the operand through a number of intermediate states until the end of the technical process. A sequence of technical sub-processes is illustrated in Fig. 7 for an electronic relay.

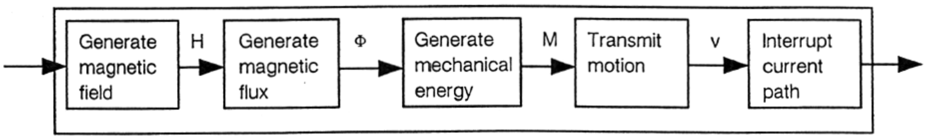


Fig. 7 Sequence of technical sub-processes constituting the structure of the technical process, adapted from Hubka and Eder (1988)

Two kinds of processes are found in a transformation system: 1) a technical process which transforms the operand, and 2) an action process delivering an effect to the technical process (see Fig. **8**). The concept of *functions* appears in the context of the action process. Hubka and Eder (1988) state that "a *function* is a *property* of the technical system, and describes its ability to fulfil a purpose", thereby providing the necessary effect to the technical process. *Functions* are nested in organs, which can be considered as functional carriers. Organs are an abstraction of the technical systems made to address functionality in the product. An organ consists of individual parts, which a product is composed of.

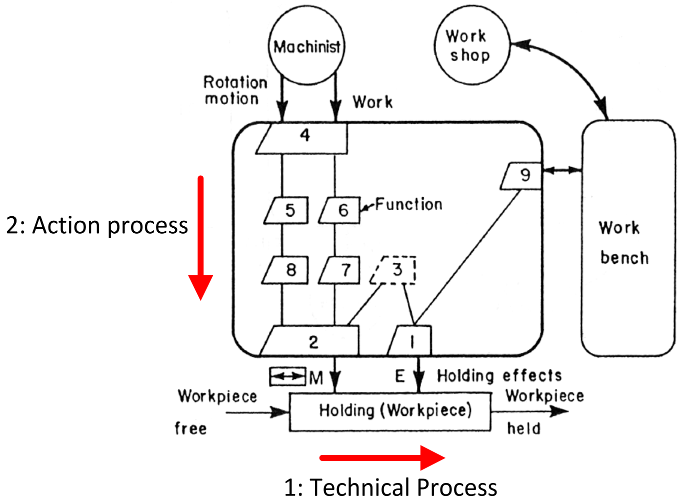


Fig. 8 The technical process (2) and the process delivering an effect (1), adapted from Hubka and Eder (1988)

The TTS is important to this research because it provides an explanation pattern to understand and describe a technical system.

3.1.2 Theory of Domains

The Theory of Domains (Andreasen 1980) is a refinement of the TSS. It was first published in 1980 by Mogens Myrup Andreasen and has since been subjected to some modifications with the purpose of continuously improving the design theory, e.g. Andreasen (1998). The ToD will be presented in the current state advocated by the K&P section at the Technical University of Denmark (Hansen and Andreasen 2002). When describing the Theory of Domains the term 'domain' refers to a specific viewpoint and not to an engineering discipline such as mechanics, electronics and software.

The Theory of Domains states that an artefact may be seen in three different domains (see **Fig. 9**):

- *A transformation domain*, where the transformation of the operands (biological object, material, energy or information) in the technical system is considered.
- *An organ domain*, where the *functions* and the 'functional carriers' (the organs) are considered.

- *A part domain*, where individual parts of the design are considered; parts that are decomposed from the organ structure.

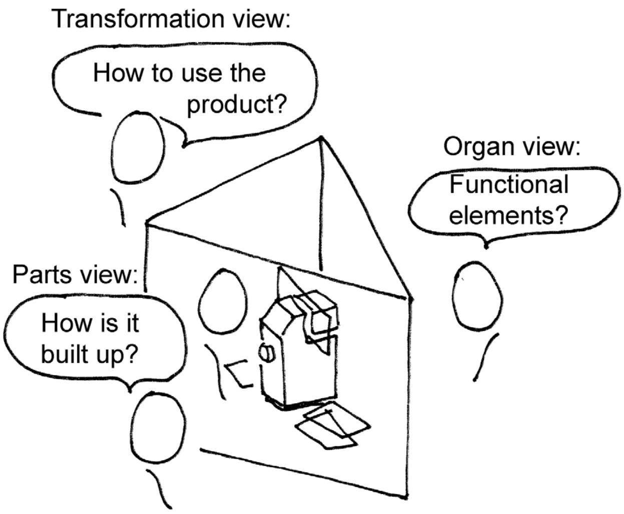


Fig. 9 Three domains in which a product can be viewed, adapted from Andreasen (1980)

Mortensen (2000) suggests that within each domain the designer can reason about structure and behaviour. Within the organ domain structure is related to the organs and how they are connected to create the necessary effect to the technical process via the *functions*. Within the part domain the structure relates to the parts assembly created in a production process. A basic assumption for ToD (as well as other design theories based on systems theory, e.g. TTS) is that the structure of a system is determined by its characteristics whereas the behaviour of the system is how it reacts to stimuli (inputs) as well as how its *properties* are perceived by humans (Hansen and Andreasen 2002). In other words: *behaviour* is what 'it does' and *structure* is what 'it is'. Behaviour seen in the ToD consists of the product's functionality and its *properties*. What differentiates *functions* from *properties* is their ability to create an effect. *Properties* do not have this ability. A *function* could be a 'release cable' whereas a *property* could be 'weight' or 'manufacturability'.

The ToD is used for the research because it is a refinement of the TTS and thus presents the most current status on how to describe a technical system. Furthermore, the definitions of a product's functionality and behaviour are useful for describing relations in a mechatronic design.

3.1.3 Function/Means Law

The Function/Means Law expresses the relation between the behaviour of a technical system and its structure. A causality between *functions* and *means*, capable of realising the *functions*, was first pointed out by Hubka in 1967 (Hubka 1967). Later this causality was formalised by Andreasen Andreasen (1980) and was named Hubka's Law. The law describes the causality as :

> *"In the hierarchy of effects (the functions), which contribute to realisation of the mechanical artefact's overall purpose function, there exist causal relations, determined by the organs (the means), which realise the effects." (Andreasen 1980)*

The causality can be illustrated in a Function/Means tree (see **Fig. 10**) where a *function* is realised by a *means* (indicated by a line). *Means* will again require supporting *functions* such as energy, control, support and auxiliary *functions* (Hubka and Eder 1988), which will be realised by *means*, and so forth. Thereby a product or a design can be decomposed in a hierarchal structure determined by the causalities between *functions* and *means*. The understanding of the decomposing pattern of technical systems is considered to be fundamental in describing a design and thus important to the research.

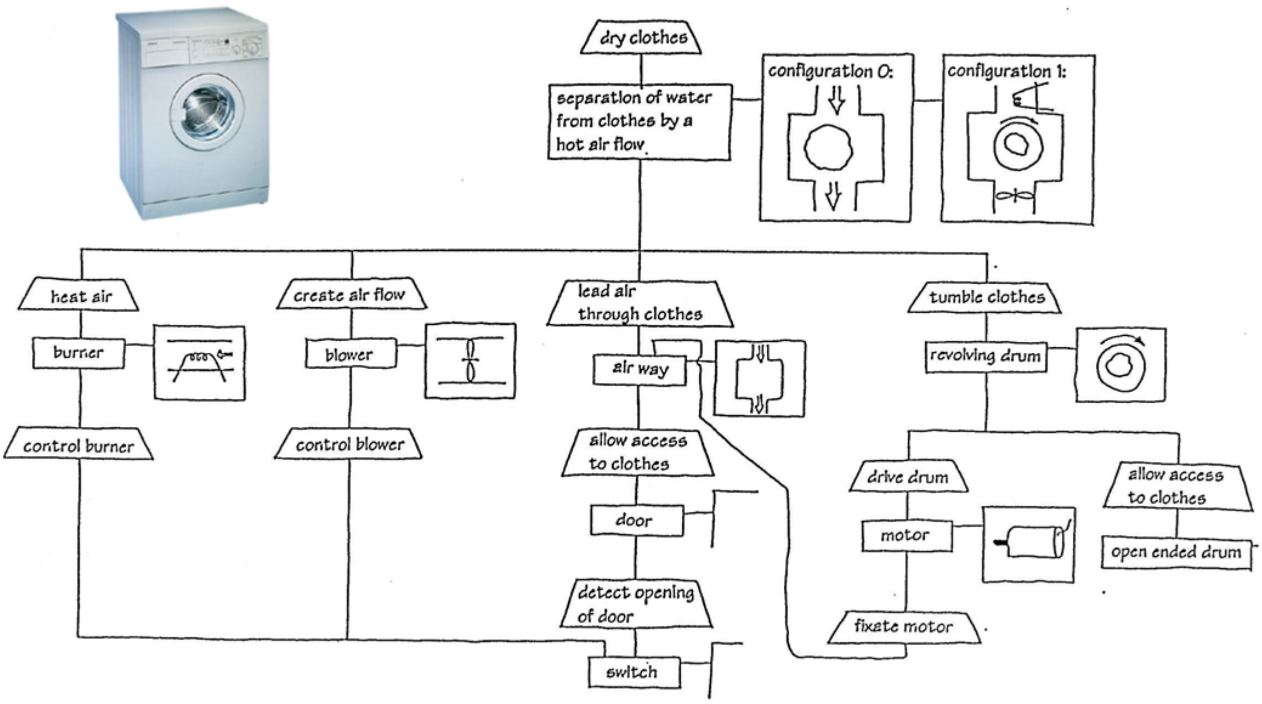


Fig. 10 Functions/Means tree, adapted from Hansen (1997)

3.1.4 Theory of Dispositions

In the Theory of Dispositions (TD) proposed by Andreasen and Olesen (1990), it is suggested to view the designers' decisions as dispositions. Olesen (1992) defines a disposition as:

> *"By a disposition we understand the part of a decision taken within one functional area that affects the type, content, efficiency or progress of activities within other functional units"*

A functional area can be considered as a 'function unit' in a company or as an activity in the life of the product as proposed by Andreasen (2007). The impact of the dispositions on activities will be effectuated in the so-called 'meetings' between the product and its encountered life phases. Knowing about or anticipating what will happen in a 'meeting' will make it possible to establish rules which can be utilized in the design process to increase the product's expected performance. Such rules are known from e.g. DFM (Design for Manufacture). One of the rules could be: "Create parts that can be stacked for faster automation assembly. **Fig. 11** illustrates the dispositional effects of the decisions taken in the development of the product and the affected life phases. The contribution is important to this research because the definition declares that there will be an effect on our decisions as designers whether we are aware of it or not. In development of mechatronics it might not be clear how dispositions in one domain may affect the other domains, because the designers taking the decisions may not be proficient within the other domains.

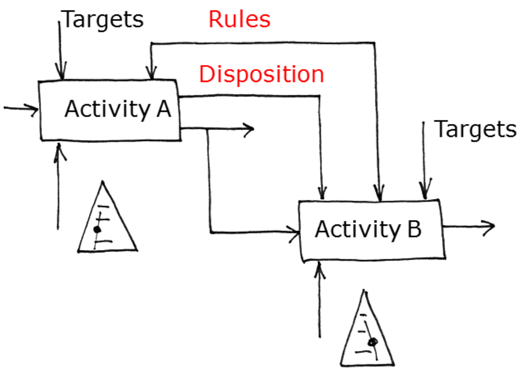


Fig. 11 The dispositions made in activity A affect activity B. Rules can be established guiding the dispositions made in activity A to increase the performance of the product in activity B. Adapted from Andreasen (2007)

3.1.5 Model of the design processes – Integrated Product Development

Many suggestions for models describing the design process exist. A vast majority of the models have the common trait of reflecting a gradual determination of the design when moving through a number of phases from initial idea to finished design (see **Fig. 12** for an illustration of the gradual determination of the design seen as abstraction levels and **Table 3** for illustrations of phases in the design process). Typically, the models are aimed at covering a certain aspect of the design process whereby the models become different in what they describe. Integrated Product Development (Andreasen and Hein 1987) is aimed at describing the interaction between marketing, development and production (see **Fig. 13**). It was launched by Andreasen and Hein in 1987 and has gained international recognition since then. It is labelled 'an idealised model of product development' by Andreasen and Hein (1987) and it describes some fundamental characteristics of the design process. Among these characteristics are: the advantage of dividing the project into phases, key point decisions, planning and collaboration between functional units in the company as well as exemplifying the importance of concurrent engineering. The IPD as presented by Andreasen and Hein (1987) is used in this PhD research project as the explanation pattern for viewing the design process. The gradual determination of the design and the need for concurrency are two of the most important aspects related to this research.

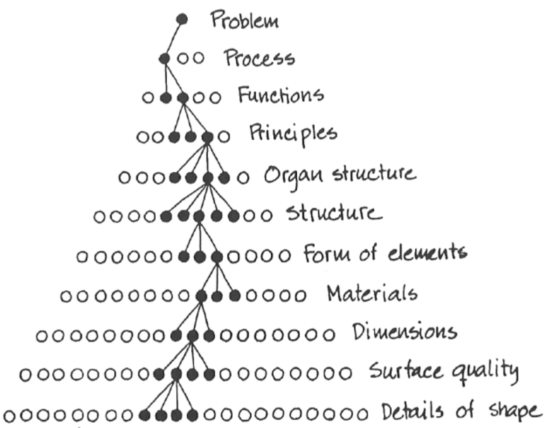


Fig. 12 Levels of abstractions in the design process, adapted from Tjalve in Andreasen and Hein (1987)

Models	Establishing a need phase	Analysis of task phase	Conceptual design phase	Embodiment design phase	Detailed design phase	Implementation phase
Booz et al. (1967)	X	New product strategy development	Idea generation ¦ Screening & evaluation	Business analysis; Development	Testing	Commercialisation
Archer (1968)	X	Programming ¦ data collection	Analysis; Synthesis	Development	Communication	X
Svensson (1974)	Need	X	Concepts	Verification; Decisions	X	Manufacture
Wilson (1980)	Societal need	Recognize & formalize ¦ FR's & constraints	Ideate and create	Analyze and/or test	Product, prototype, process	X
Urban and Hauser (1980)	Opportunity identification	Design		Testing		Introduction (launch) ¦ Life cycle management
VDI-2222 (1982)	X	Planning	Conceptual design	Embodiment design	Detail design	X
Hubka and Eder (1982)	X	X	Conceptual design	Lay-out design	Detail design	X
Crawford (1984)	X	Strategic planning	Concept generation	Pre-technical evaluation	Technical development	Commercialisation
Pahl and Beitz (1984)	Task	Clarification of task	Conceptual design	Embodiment design	Detailed design	X
French (1985)	Need	Analysis of problem	Conceptual design	Embodiment of schemes	Detailing	X
Ray (1985)	Recognise problem	Exploration of problem ¦ Define problem	Search for alternative proposals	Predict outcome ¦ Test for feasible alternatives	Judge feasible alternatives; Specify solution	Implement
Cooper (1986)	Ideation	Preliminary investigation	Detailed investigation	Development ¦ Testing & Validation	X	Full production & market launch
Andreasen and Hein (1987)	Recognition of need	Investigation of need	Product principle	Product design	Production preparation	Execution
Pugh (1991)	Market	Specification	Concept design		Detail design	Manufacture ¦ Sell
Hales (1993)	Idea, need, proposal, brief	Task clarification	Conceptual design	Embodiment design	Detail design	X
Baxter (1995)	Assess innovation opportunity	Possible products	Possible concepts	Possible embodiments	Possible details	New product
Ulrich and Eppinger (1995)	X	Strategic planning	Concept development	System-level design	Detail design	Testing & refinement ¦ Production ramp-up
Ullman (1997)	Identify needs ¦ Plan for the design process	Develop engineering specifications	Develop concept	Develop product		X
BS7000 (1997)	Concept	Feasibility	Implementation (or realisation)			Termination
Black (1999)	Brief/concept	Review of 'state of the art'	Synthesis; Inspiration	Experimentation; Analysis / reflect	Synthesis; Decisions to constraints	Output; X
Cross (2000)	X	Exploration	Generation	Evaluation	Communication	X
Design Council (2006)	Discover	Define	Develop	¦ Deliver		X
Industrial Innovation Process 2006	Mission statement	Market research	Ideas phase	Concept phase	Feasibility Phase	Pre production

Table 3 Overview of phases of the design process, after Howard et al. (2008)

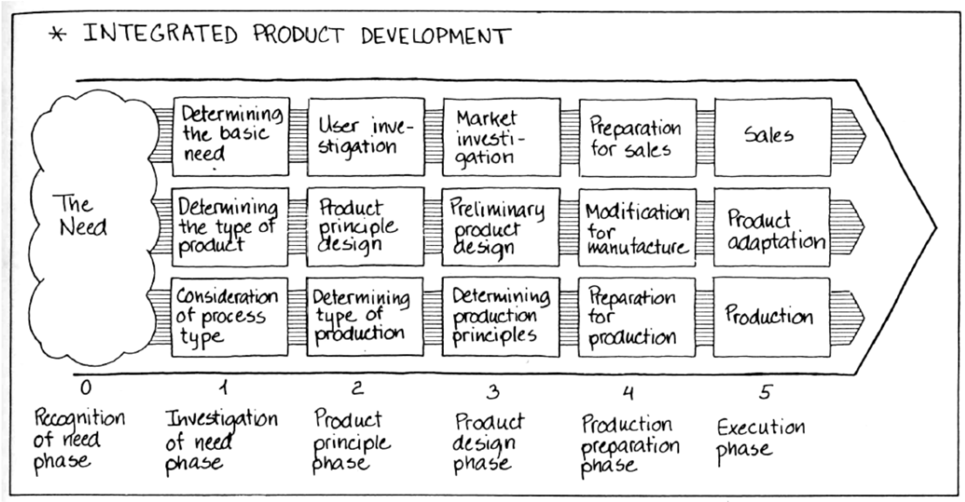


Fig. 13 The Integrated Product Development Model depicting the sequence of the development task for marketing, development and production, adapted from Andreasen and Hein (1987)

3.2 Relating design theories to the electronics and software domain

From the previous descriptions we observe that we have theories for describing the technical system, dispositions made by the designers affecting the product's life phases as well as models describing the design process. The question is then whether equivalents can be found for the electronics and software areas and whether the concepts are comparable or even similar.

3.2.1 Software domain

The use of software in products has increased since Alan Turing first pointed out the possibility of a machine to auto-process a set of code lines in 1936. Large defence systems in the USA and the endeavours into space led to the development of Systems Engineering out of the need to manage large software projects. It has later been developed to incorporate both software and hardware. In this context hardware should be considered as electronics (Blanchard and Fabrycky 1998). The organization called INCOSE has an enhanced perception of the coverage of Systems Engineering. They claim Systems Engineering to be capable of being applied to multi-disciplinary designs also comprising the mechanics and the civil engineering disciplines (Haskins and Forsberg 2011). Systems Engineering is strongly rooted in software and draws upon process models such as the V-model, the Waterfall model and the Spiral model (see Fig. 14 - Fig. 16) (Blanchard and Fabrycky 1998). Other process models within software include SCRUM and RAD, which are aimed at 'agile development' (CMS 2007). The process models consist of stages with key decision points. The described activities in the models express a gradual determination of the design. The typical stages found in the software process models comprise goal setting, requirement definition, conceptual clarification including functional descriptions, implementation and testing (Haskins and Forsberg 2011; Blanchard and Fabrycky 1998; Sage and Rouse 2009; CMS 2007). Evaluated on the design process the mechanical and the software development process are comparable in terms of stages and key point decisions (which is influenced by project management ideas (Cooper 2001)) as well as the gradual determination of the design. Though there are differences between engineering disciplines such as the absence of a production phase in software, the focus is on commonalities in terms of describing the technical system as well as describing the design process to find common traits that can facilitate a shared understanding across the engineering disciplines.

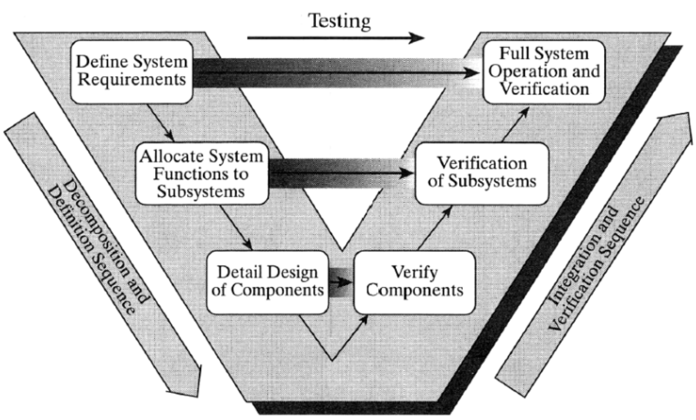


Fig. 14 The V model as printed in Blanchard and Fabrycky (1998)

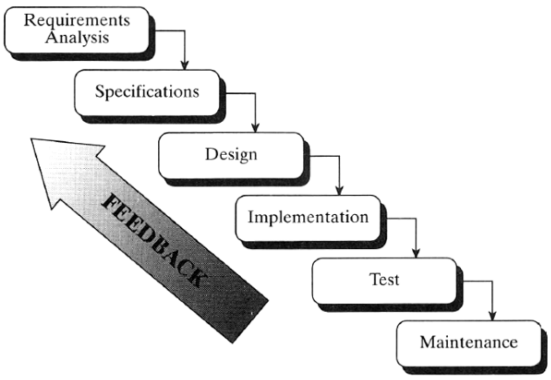


Fig. 15 The Waterfall model as printed in Blanchard and Fabrycky (1998)

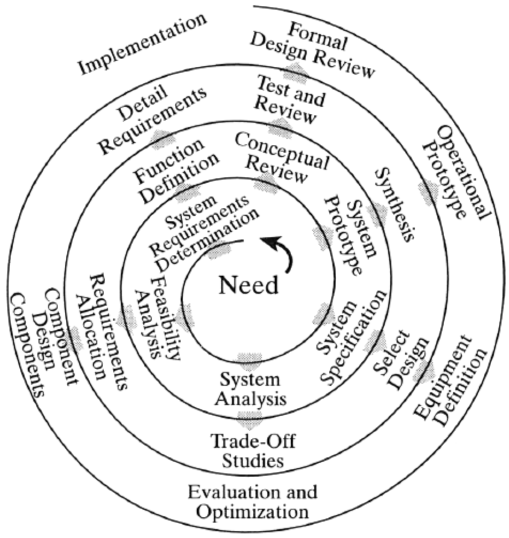


Fig. 16 The Spiral Model as printed in Blanchard and Fabrycky (1998)

Defining the functionality of the product plays a central role in specifying, developing and testing the software in most software development models (Robertson and Robertson 2006; Haskins and Forsberg 2011; Blanchard and Fabrycky

1998; Sage and Rouse 2009). In references on software the functionality of a product is often described via the term 'function' (Blanchard and Fabrycky 1998) or via the term 'functional property' (Robertson and Robertson 2006). Both terms are covering the capability of the software to create an effect, thus describing what the product 'is able to do'. In Robertson and Robertson (2006) 'properties' are classified as either 'functional properties' or 'non-functional properties, both contributing to the behaviour of the product. *Properties* seem to be more loosely defined in publications on software systems compared to the description of *properties* in the Theory of Domains by Andreasen. Yet, the concepts seem to be equivalent for the use of the terms *functions* and *properties*. Even though there might be a slight difference in how *functions* are defined within references from the literature, the overall concepts are comparable between software and mechanical engineering. Within software engineering, solutions to *functions* can be solution principles such as architectural patterns, procedures (blocks of executable software code) or lines of the software code (Pressman 2005). In this sense these solutions are *means* to realise the desired functionality of the software.

Therefore it seems that the terms *functions*, *properties* and *means* have equivalents within the mechanical and software discipline. Furthermore, it seems that the behaviour of a product can be expressed by its *functions* and *properties* collectively.

3.2.2 ELECTRONICS DOMAIN

Even though the electronics discipline is at least as old as the software discipline, literature sources describing design science within the electronics discipline are scarce or might even be categorised as completely lacking. The reason seems to be that design of electronics is application-specific and to a large extent based on the adaption of standard designs. In electronics design emphasis seems to be on mathematical analysis and simulation of the behaviour of the system. Despite searching throughout the PhD project for procedures for developing electronics or ways to describe the technical system from a design point of view, descriptions similar to what can be found within the mechanical or the software domain have not been found. Apart from using search engines of different scientific databases, interviews with several electronic engineers did not reveal references for procedures. Included in the group of interviewed electronic engineers were two consultants and a university teacher in analogue circuitry. Two of the best procedural descriptions of designing electronics are presented in what follows, which indicate the maturity level of those available in the literature. As a consequence I had to consider other means for evaluating what theories could be used for bridging the disciplines when the design process or the product need to be described. Besides the few available references on the design process, proposals were evaluated for multi-disciplinary methodologies and frameworks claiming to comprise the electronics domain.

A design process can be patched together from Williams (1991) and Jones (2004) due to lack of references. They describe the conceptual and detailed design phase and the creation of the layout of traces on the PCB respectively. In the book by Williams he suggests a procedure based on his own experience of designing analogue circuits. The procedure is presented below:

1. *Draw a 'front panel' of the instrument to be designed. "Try out" its functions.*
2. *Make a simple circuit model for one function or aspect of the instrument. The model should emphasize that one aspect and deemphasize other aspects.*
3. *Make simplifying assumptions and analyze the circuit by inspection where possible. Go back and forth between time domain and frequency domain analysis. Check your assumptions.*
4. *Change the model and analyze again until the results are acceptable.*
5. *Repeat steps 1-3 for other aspects of the instrument.*

6. *Design the full circuit with circuit blocks that behave like the ideal blocks in the model.*
7. *Test a prototype of the instrument to see if it behaves like the models.*

From the procedure we can see a gradual determination of the design and the focus on the *functions* in the design as a way to decompose the design task. However, based on the descriptions by Williams (1991), development stages or key point decisions are not specifically advocated. Subsequent to the design of the schematics, the PCB and routing has to be designed. Once again, no general description has been found describing the process in a design research context. This is the reason why a procedure has been obtained from a guideline on how to create the tracing (Jones 2004).

1. *Throw down all the components onto the board.*
2. *Divide and place your components into functional "building blocks" where possible.*
3. *Identify layout critical tracks on your circuit and route them first.*
4. *Place and route each building block separately, off the board.*
5. *Move completed building blocks into position on your main board.*
6. *Route the remaining signal and power connections between blocks.*
7. *Do a general "tidy up" of the board.*

Even though there does not exist a general design theory for design of electronics, the references (Williams 1991; Jones 2004) state a procedure which might mirror design practice in industry.

On the matter of how the term 'function' is used within electronics, Williams (1991) describes his conception of a 'function' in the following sentence: "The 'front panel' helps to evaluate the *functions* that were specified and to investigate interactions between *functions*. In other words, does it do what you wanted, the way you wanted it to?" Even though it must be considered as a personal statement this conception is equivalent with the conception of a *function* within the mechanical engineering discipline.

As a consequence of the scarce references for describing electronics in a design science context, multi-disciplinary methodologies claiming to cover electronics will briefly be described in what follows. The VDI2206 guideline (Association of German Engineers 2004) is aimed at development of mechatronics, and should as such comprise terms applicable within the design of electronics. The guideline draws on work by Pahl et al. (2007) as well as the V-model originating from the software engineering discipline (proposed by Boehm (1984) and formalised by e.g. IABG (1997)). No references are found in VDI2206 used for referencing terminology used specifically in electronics. Though terminology from the VDI-guideline primarily adopts the terminology from Pahl et al. (2007), it could be assumed that the terminology used in the guideline could be applied to the electronics domain, if the methodology is believed to be truly multi-disciplinary. Zha et al. (2005) describe how co-design of hardware and software can be performed by use of an extended UML model (Group 2011). In the process of co-designing, the activity of 'functional partitioning' between software and hardware is stressed as central; once again indicating that functional reasoning might well be applied in the electronics domain. Considering the found references, the terms *function*, *property* and *means* should have the potential of being used across the mechanical and the electronics engineering disciplines.

3.3 Conclusion

The starting point is to use TTS and ToD as the founding theories for describing 'a mechatronic design'. The purpose of the previous sections has not been to advocate that design theories and terminology are equivalent and can be used across the engineering discipline boundaries. The purpose has been to assess the consequence of using a group of terms obtained from systems theory from the mechanical research area to describe 'a mechatronic design' and the design process without creating linguistic barriers. In this research project I try to balance the need for an accurate vocabulary of well-defined terminology with a set of terms, which can be used across domains. The investigations of the terminology used within the engineering disciplines point towards the following terms having equivalents within each of the engineering disciplines: *behaviour*, *function*, *property* and *means*. I choose to use these terms rooted in TTS and ToD with the meanings reported in the following paragraph.

The *functions* and the *properties* of a product constitute the *behaviour* of the product. *Functions* express the capability of an object to deliver a desired effect or a purpose (Hubka and Eder 1988). The structure is what realizes the *behaviour* of the product. Structure and solution principles are considered as *means*. In mechanical engineering the physical structure realizes the *behaviour*, in electronics the electronic components are realising the *behaviour* and within software engineering the software code is creating the *behaviour* of the products.

In addition, the theories TTS, ToD, Function/Means Law, TD, IPD will constitute the theoretical basis for the research project. The investigation of theories from the software and the electronics discipline does not indicate conflicts using these theories as the theoretical basis, which is supported by investigations by Buur (1990).

4 PRACTICAL BASIS

As a consequence of my job as a product development consultant I have participated in a number of mechatronic projects and gained insights into the challenges related to the mechatronic phenomena encountered by a development team. I have worked as a team member, as a domain specialist and as a project manager in these projects and thus have seen the encountered problems from different angles. Some of the projects I participated in are described briefly in the following section in order to highlight some aspects of the mechatronics challenges. Subsequent to the examples, I have stated reflections achieved as a consequence of my participation in mechatronic projects and my contacts with the industry.

The product in **Fig. 17** represents a ”New Product Development” aimed at the high-end market for outdoor sport watches. The project was initiated by Morten Linde and Jørn Werdelin, co-founders of the company Linde Werdelin. The product idea is based on combining a classical mechanical watch with a digital instrument containing advanced *functions*. The watch and the instrument are two separate units which can be joined for the purpose of e.g. alpine skiing. An additional two external units wirelessly transmit heart rate information and temperature information to the instrument. The high demand for the product to be compact required a high level of integration between the electronic and mechanical solution. The state-of-the-art functionality required in the instrument meant that the electronics and software needed to be developed in parallel. These constraints forced the electronics, software and the mechanical solutions to be developed concurrently, which required a high demand for integration of solution-finding across domains and a great need for collaborative and productive decision making.

Fig. 17 The Linde Werdelin watch and instrument

The second product is called a ‘Dolly’ (**Fig. 18**) and is used on an oil drilling rig to fix a 45 tonne drive unit rotating the drilling string. The Dolly is mounted in a rack and can move the huge DC motor and gear assembly up and down in the tower of the oil rig. It can also extend and retract the motor and gear assembly by hydraulic actuators to be positioned over the drilling hole for faster assembly and disassembly of the drilling string. The product was a line-extension to an already existing line of Dollies. The Dolly although 5 meters high was just one of many sub-systems, which was contracted. Due to the knowledge of preceding products and the complexity of the contracted system we emphasized a stringent definition of interfaces and pre-assignment of components to subsystems. Yet, the specified interfaces still had to be redesigned during the design process since the ‘reality’ was not always ideal and due to changes of external constraints. This required the cad model to be flexible to avoid tedious and expensive redesigns of the cad model. Simulating the movements of the sub-systems posed a challenge as well. Many wires and cables were going to and from the DC motor and a high demand for compactness was required. The challenge appeared because the cad system was not capable of perceiving the cabling as flexible during the simulation of the movement of the sub-systems. Therefore manual checks and estimations of locations of the cabling had to be performed. If the cabling obstructed the movements at installation, the economic impact could be severe and cause fines for delaying the shipment of the finished product.

Fig. 18 Dolly for oil rigs

This following project was about updating an already existing system used for testing antennas for satellites (**Fig. 19**). Due to advancements of the testing technology a new system for carrying and rotating the disc antenna was needed. The disc antennas were heavy and just the slightest deformation of the supporting axle rotating the disc would cause inaccurate measurements. Yet, the only way to ensure electrical connection to the antenna and to the equipment on the antenna was to place the cabling in the centre of the axle and have it go through the side of the axle for mounting the required connectors. Choices performed in the mechanical domain affected the electronic domain and the plausible electronic solutions would dictate the solution space in the mechanical domain. *Properties* would have to be considered across the engineering boundaries to achieve the project objectives. The need for carefully considering the solutions from a holistic point of view instead of from within each separate domain was mainly caused by the need for pushing the material capabilities to the limits, strength-wise. In addition, the life phase scenarios had to be considered from mechanical and electronic aspects. Some of the product's life phase scenarios were: 'mounting the disc antennas', 'adjustments of the position of the disc antennas' and 'test and measurement of the disc antennas'. An example: If the cable from the antenna had to be connected to the axle-system after the antenna had been mounted there was a need for creating space for hand tools right behind the antenna. This would dictate possible positions for electric sockets, which again would cause different stress dissipation in the material.

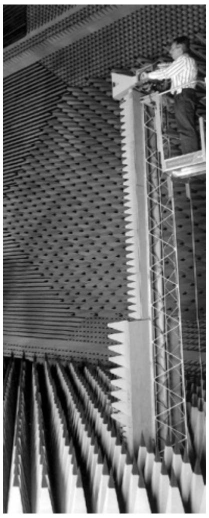


Fig. 19 Test facility for antennas

The projects described above highlight some concrete examples of mechatronic challenges faced by the development teams. Based on my prior experience with mechatronic projects and my contact to the industry, some central questions and problem areas appear related to mechatronic development, which have caught my interest at the initiation of this research project. These are summed up in the following:

- **The balance between integration and disintegration between the engineering disciplines in a project**. An extreme extent of integration between teams from different engineering disciplines will increase the complexity of the project since a vast number of relations have to be monitored, updated and agreed upon. A complete disintegration between the engineering disciplines does not seem to be desirable either, since loss of potential synergy in the solution will be likely. Both extremes seem to be undesirable for new product development while total separation might work to some extent for line-extension products. As part of managing the complexity in development projects it seems advantageous to companies to consciously decide what integration level is needed for each particular project.
- **The problem of 'silo-thinking'**. The more disintegrated the departments or teams from different engineering disciplines are, the risk of neglecting needs from other engineering disciplines increases, leading to what we might call 'silo-thinking'. Problems occur due to lack of attention to consequences of one's own decisions in other domains. Solutions are chosen based on a best fit basis, but only within the one domain. Had the decision process been more holistic a solution which would have been less optimal for the given domain would have led to a better overall solution.
- **Synchronization of design activities**. Aligning deliverables from the mechanical, electronics and software teams does not come easy. The nature of the development process is different within the different engineering disciplines and special care should be taken to design the project activities to obtain milestones at which the progress of the project can be monitored and evaluated; it being a conceptual description or an integration test. An example is that software does need time allocated for 'manufacturing', which might lead to situations where printed circuit boards for e.g. the integration test have to be ordered prior to the initial testing of the software.
- **Control of *properties* across engineering disciplines**. *Properties* of the product are likely to be dissipated onto different modules, which in addition might be developed by different engineering disciplines. Without proper control of the *properties* in both assigning them to modules and tracking them, it seems to be very hard to obtain the required performance of the product to create the necessary competitive advantage.
- **Sequential or concurrent engineering**. Some companies arrange the development of mechanics, electronics and software in a sequence in what seems to be an effort to keep complexity low. The down-side is that the development project will be prolonged as a consequence of not running the development tracks concurrently. There must be huge potential in being able to align the development tracks in terms of shorter lead-time and the possibility of utilizing the potential synergy in the joint collaboration if, noteworthy, the complexity of the concurrent process can be restrained.

The observations presented above are reflections from mechatronic projects, and are as such not scientifically validated. Yet, my reflections seem to be supported by the reporting from researchers (e.g. Andreasen and McAloone (2001)) on the need for integrating mechanisms in terms of methods and tools and mind-sets to support various aspects of mechatronics design.

5 RESULTS

Achieved results from the research are presented in this section. The results are captured by five scientific papers. The papers are labelled A, B, C, D and E. Papers A-C are used for addressing Research Questions 1 and 2. Paper D is used for addressing Research Question 3, whereas Paper E is used for addressing Research Question 4.

The first paper (Paper A) contains a case study used for highlighting important aspects argued by the authors to be central when developing mechatronic products. Literature is reviewed to investigate to what extent the aspects are acknowledged and described. The paper serves as a 'deep-dive' into concrete challenges of performing multi-disciplinary product development. Paper B is a conference proceeding paper, which is elaborated and extended into a journal paper (Paper C). Only Paper C is used for the reporting of the results. Paper C is a broad and systematic investigation of the challenges linked to the integration phenomenon of performing mechatronic product development. The aim is an overview of these challenges and their solutions along with interesting insights, which enables answering the first two research questions. In Paper E the focus is aimed at one of the stated challenges from Paper C. The difficulty of 'modelling and controlling multiple relations in the product concept' is addressed in Paper C by searching for a classification and definition of 'a dependency'. The last paper (Paper E) utilizes the established classification of 'a dependency' and a 'Mechatronic Integration Concept' is proposed capable of capturing the dependencies to facilitate a shared understanding across domains at integration meetings.

5.1 PAPER A

Title: "A Mechatronic Case Study Highlighting the Need for Re-thinking the Design Approach" (Torry-Smith and Mortensen 2011)

Specific research question for this paper: How are integration phenomena handled in the design process of developing mechatronics products and how well are they described in literature? The phenomena to be investigated have been selected by the authors prior to the investigation.

Contribution to the PhD Research Questions: A case study description has been emphasised in Paper A, and it brings insight into design phenomena and integration patterns in mechatronic development. Thus it contributes to clarifying Research Question 1. The reviewed literature is limited compared to what is included in Papers B and C. Hence Paper A contributes partially to Research Question 2.

Research method: The article is the first attempt to describe what is important and central in the design process of developing mechatronic products. A number of phenomena assumed by the authors to be important in the design process are selected for the investigation. Seven phenomena (labelled aspects in the paper) are selected, though they might not represent an exhaustive list of all mechatronic integration phenomena. A case study is used to elucidate the context in which the seven aspects appear, while showing the relevancy of the selected aspects. The case study is about the development of an external temperature unit, which is a part of a watch system (see **Fig. 20**). A literature study is performed on how well the seven aspects are described and to what extent methods and tools are available to better control these aspects when designing.

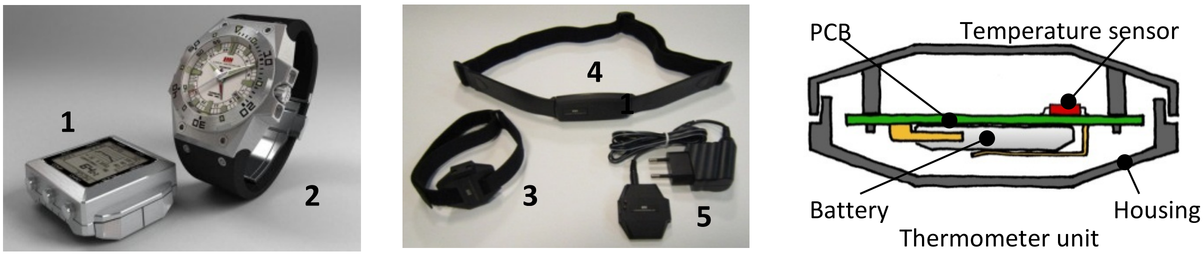


Fig. 20 1: Instrument, 2: Watch, 3: Temperature unit, 4: Heart rate unit, 5: Instrument charger

Results and conclusions: In the paper the seven aspects are categorised as belonging to each of the following categories: the design process, the product, and the user-perceived value of the product. Yet, being more knowledgeable towards the end of the PhD project I would characterize them all to belong to the phenomenon of designing. The seven aspects are reported in **Table 4** along with the reviewed literature sources. The coverage of the aspects in the literature is rated, in the table, as follows:

'0': The aspect is not described (also marked as white cells).
'1': The aspect is acknowledged and a characterisation may have been performed (marked as grey cells).
'2': The aspect is treated thoroughly and a method for handling the aspect is suggested (marked as black cells).

Table 4 Overview of coverage of literature sources related to the seven aspects

	VDI 2206 [1]	Systems Engineering[2]	V-modell XT[3]	J. Buur[4]	J. Gausemeier[5]	S. Jansen/E.G. Welp[6]	V. Salminen/ A. Verho [7]	R. Isermann[8]	R. H. Bishop[9]	R. H. Bracewell[10]	Pahl/Beitz[11]
Synchronization of mechanical, electronics and software development process (A1)	1	1	0	1	1	0	1	2	0	0	1
The domains seen as iterations (A2)	0	0	0	0	0	0	1	0	0	0	0
Function allocation and alternatives (A3)	1	0	0	1	1	2	1	1	1	0	0
Distribution of *functions* and *properties* (A4)	1	0	1	0	1	1	1	0	0	0	1
Sharing schemes for *functions* and *properties* (A5)	0	0	0	0	0	0	0	0	0	0	0
Handling of physical interfaces (A6)	1	1	1	1	1	1	1	0	0	0	0
User-perceived value in the life phases (A7)	0	0	1	0	0	0	1	0	0	0	0

References: [1](Association of German Engineers 2004), [2](Sage and Rouse 2009; Blanchard and Fabrycky 1998),[3](IABG 1997), [4](Buur 1990, 1989), [5](Gausemeier et al. 2001; Gausemeier et al. 2008), [6](Jansen 2007; Welp and Jansen 2004), [7](Salminen and Verho 1989; Salminen and Verho 1992; Verho et al. 1995), [8](Isermann 2005), [9](Bishop 2002), [10](Bracewell and Sharpe 1996; Bracewell et al. 1993), [11](Pahl et al. 2007).

The investigation of the seven aspects in the case study highlights the need for close collaboration between the engineering disciplines. It reveals the need for being able to see the consequences of mechatronic decisions propagating down and into the domains affecting a vast amount of tasks to be performed. It also shows the need for handling dependencies and the need for coordinating the effort in laying out the sequence of clarifications which will lead to a fully functional product with adequate performance. The literature study shows that the aspects are covered partly by the literature and that methods and tools for handling them are sparse or lacking. A part of the explanation is that many of the scientific contributions found in the literature on mechatronics are dominated by a control engineering perspective to the mechatronic challenge. The publications reported on in Paper A that have this control focus tend to assume that the overall system is predetermined, thus moving the focus away from the synthesis process and toward control optimization issues. Though the literature studies in Papers B and C are more comprehensive, Paper A describes a number of very detailed situations in an actual product development, which poses a range of challenges faced by the development team. The main conclusion based on Paper A is that though there might be suggestions in literature on how to manage the seven aspects on a general level (e.g. interfaces in modularisation research), there are very few sources directed at providing a mechatronic-specific support for the aspects. There is a shortcoming in the research carried out for Paper A, caused by us selecting the aspects based on our experience. This presents a potential bias to the investigation. Yet, the value lies within highlighting aspects difficult to handle in the design process, for which we do not have easy solutions. The bias is omitted in Papers B and C due to the research setup. The research for Papers B and C is described in the following section.

5.2 PAPERS B & C

Paper B (Torry-Smith et al. 2011) is a conference proceeding presented at the "ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2011". Paper C (Published in JMD) is an extended version of Paper B. Since results obtained for Paper B are included in Paper C, only Paper C will be discussed in this section (section 5.2).

Title: "Challenges in Designing Mechatronic Systems" (Torry-Smith et al. 2012)

Specific research question for this paper: The research question is twofold. What challenges related to the integration phenomenon within the development of mechatronics are stated by the most prominent researchers within the research field? What solutions to these challenges are proposed in the literature and how well do they mitigate the challenges?

Contribution to the PhD Research Questions: Paper C is continuing the quest started in Paper A of finding an answer to Research Question 1 of the PhD thesis. In the paper, solutions are investigated, thus presenting findings aimed at Research Question 2. Paper C therefore answers Research Questions 1 and 2.

Research method: To be able to assess the stated challenges in mechatronic development a literature study is performed using contributions from two sources. Source one consists of papers published in a five year period (2007-2011) in the proceedings of the ASME IDETC/CIE conference, to serve as the first basis. The second source is a general search for relevant papers about mechatronics design. The search targets relevant journals and conference proceedings. The first source is used for finding the most prominent researchers within the ASME IDECT/CIE community and extracting their statements on mechatronic challenges. For that purpose data processing is used. The most cited researchers within the community are found by data processing the collective sum of references used in mechatronic-relevant papers from source one. From source two a number of researchers are added to the list. 3-5 papers from each of the researchers from this list are reviewed and statements on mechatronic challenges are extracted. By further processing this data an overview of challenges is created. Turning our attention to the literature once again, proposed solutions to the challenges are reviewed. Some selected challenges and their solutions are illustrated through a mechatronics case study from industry.

Results and conclusions: The analysis of source one reveals the most cited researchers. In Paper C only the researchers who have been cited no less than five times in different papers are stated. **Table 5** shows an extended list of the cited researchers to present a more comprehensive picture. The table is dominated by research groups, and some of the researchers are referenced due to their contributions on fundamental design theories or other fundamental contributions (e.g. Pahl et al. (2007), Suh (2001), Ulrich and Eppinger (2000). Hence, researchers not addressing mechatronics-specific challenges are omitted in the study. The investigation of the selected researchers and research groups reveals a collection of challenges appearing in design of mechatronics. The list of challenges is presented in **Table 6**.

Table 5 List of most cited researchers within the ASME IDECT/CIE mechatronics community based on the conference proceedings 2007-2011. The number in the brackets indicates the number of times the researcher has been referenced out of the 30 extracted papers from the proceedings

Column 1	Column 2	Column 3	Column 4
G. Beitz (16)[a]	K. T. Ulrich (6)[f]	D. Steffen (5)[c]	B. Schulz (4)[c]
W. Pahl (16)[a]	Y. Umeda (6)[b]	N. P. Suh (5)	Y. Shimomura (4)[b]
T. Tomiyama (11)[b]	M. Yoshioka (6)[b]	VDI (5) (guideline)	R.D. Sriram (4)[g]
J. Gausemeier (9)[c]	A. Albers (5)	R. M. Burkhart (4)[h]	S. Szykman (4)[d]
K. H. Grote (7)[a]	A. A. A. Cabrera (5)[b]	T. R. Browning (4)	H. Takeda (4)[b]
K. Wood (7)[d]	S. D. Eppinger (5)[f]	P. J. Clarkson (4)	D. G. Ullman (4)
J. Feldhusen (6)[a]	S. Fenves (5)[g]	A. Diaz-Calderon (4)[e]	H. Vöcking (4)[c]
U. Frank (6)[c]	J. Hirtz (5)[d]	J. Gero (4)	K. Witting (4)[c]
D. McAdams (6)[d]	OMG (5) (software)	H. Giese (4)[c]	D. Zimmer (4)[c]
C. J. J. Paredis (6)[e]	S. Pook (5)[c]	G. J. Muller (4)	U. Lindemann (4)
R. B. Stone (6)[d]	A. Schmidt (5)[c]	R. S. Peak (4)[h]	Mathworks (4) (software)

[a]Pahl group; [b]Tomiyama group; [c]Gausemeier group; [d]Wood group; [e]Paredis group; [f]Ulrich group; [g]Fenves group; [h]Burkhart group

Table 6 Overview of stated challenges and the researchers stating them

Type	#	Challenges	Researchers/Research Groups														
			Source 1							Source 2							
			Tomiyama [a,1]	Gausemeier [a,2]	Wood [a,3]	Paredis [a,4]	Albers [5]	Cabrera [6]	Fenves [7]	Adamsson [8]	Buur [9]	Salminen [10]	Andreasen [11]	Lindemann [12]	Browning [13]	Shea [14]	Bradley [15]
Product	A	Lack of a common understanding of the overall system design	X	X		X	X			X		X		X	X	X	X
	B	Difficulty in assessing the consequences of choosing between two alternatives	X	X		X	X				X	X			X		X
	C	Lack of a common language to represent a concept	X	X	X	X	X	X		X	X	X	X			X	X
	D	Modelling and controlling multiple relations in the product concept	X	X		X	X			X				X	X		
	E	Being in control of the multiple functional states of the product		X	X		X						X				
	F	Transfer of models and information between domains (expert groups)		X	X	X	X	X	X							X	X
Activity	G	Synchronizing development activities to attain concurrent engineering		X	X			X	X				X		X		
Mind-set	H	Different tradition within the domains for how to conduct creative sessions										X					
	I	Reluctant to interact with engineers from other disciplines										X					
	J	Different mental models of the system, task and design-related phenomena	X	X		X		X		X	X	X	X		X		X
Competence	K	Lack of common language to discuss freely at creative meetings	X	X				X		X	X	X	X			X	X
	L	Education within disciplines do not call for integration in professional life									X				X		X
	M	The nature of design is different	X	X						X	X		X	X			
Organizational aspects	N	Product complexity affects the organization complexity	X	X											X		
	O	Knowledge transfer between domains is inadequate (even in cross-disciplinary teams)	X	X				X	X			X					X
Other aspects	P	Lack of a broadly accepted methodology	X	X	X	X					X	X	X	X			
	Q	Mechatronic ownership is lacking								X		X	X				
	R	System engineers are lacking detailed information of the system										X					
	S	Complexity as a generic problem	X	X	X	X	X		X	X				X	X		X

[A]Research groups.

References: [1](Tomiyama et al. 2007), [2](Gausemeier et al. 2009a), [3](Nagel et al. 2008a) Nagel is part of Wood group, [4](Shah et al. 2010) Shah is part of Paredis group, [5](Albers et al. 2011), [6](Cabrera et al. 2010), [7](Fenves 2001), [8](Adamsson 2004), [9](Buur 1990), [10](Salminen and Verho 1989), [11](Andreasen and McAloone 2001), [12](Kreimeyer et al. 2008), [13](Danilovic and Browning 2007), [14](Wolkl and Shea 2009), [15](Bradley 2010).

The challenges presented in **Table 6** range from product-specific challenges to company-level challenges and various means can be used for addressing them ranging from mechatronics-specific methods and tools to management and organizational theories. Due to the scope of the research project (i.e. product development) the search for solutions is mainly directed at solutions claimed to be useful in a mechatronics development setting aimed at the development activities. The found solutions are grouped and presented in the following:

- Activities based on functional approaches and functional decomposition (Buur (1990); Nagel et al. (2008a); van Beek and Tomiyama (2009); Suh (2001)), applying *functions*, *means* patterns (Nagel et al. 2008b), C&C-A (Albers et al. 2011), state and event relations, hierarchical approach (Hehenberger et al. 2010).
- Relationship management e.g. DSM and DMM (Braun and Lindemann 2007; Danilovic and Browning 2007), QFD (Hauser and Clausing 1988), FunKey (Bonnema 2011).
- Controlling design activities through requirements management (Systems engineering (Sage and Rouse 2009), Woestenenk et al. (2010)).
- A process model containing activities for the development process (Isermann (2005), VDI2206 (Association of German Engineers 2004), Salminen and Verho (1989), Systems Engineering (Sage and Rouse 2009)).
- Informal description consisting of a number of modeled/described aspects to specify systems, A3 overviews (Borches and Bonnema 2010), Salminen and Verho (1989), Buur (1990).
- Modelling languages to describe the system as a whole, formally or semi-formally. SysML (Object Management Group 2010), SFSL by Gausemeier et al. (2009a), AM tool by Cabrera et al. (2011).
- Model transformation from a design model in one domain into a design model in another domain (Gausemeier et al. (2009b); Shah et al. (2010)).
- Formalized specification of interfaces. (ISO/IEC 81346 (2012), Systems engineering (Sage and Rouse 2009)).
- Simulation of phenomena that incorporate elements from the different domains (e.g. Modelica (Modelica Association 2010), Bond Graphs (Wu et al. 2008), and integrated simulation (Albers and Ottnad 2010)).
- Setting up a systems integration group in the project (Adamsson (2004); Andreasen and McAloone (2001)).

The main conclusions from Paper C are summarised below:

- The available solutions only partly cover the stated challenges.
- A large part of the identified solutions appear to support analysis activities rather than synthesis activities.
- The solutions which are not analytical in nature are typically based on functional reasoning
- Solutions based on functional reasoning have the potential to be applied across domains.
- Solutions based on functional reasoning only seem to support the initial steps in the conceptual phase. It appears that introducing *means* to the *functions* obstructs a common model describing the product concept as a whole.
- In the absence of a common model the designers must rely on informal exchange of information (integration meetings) and/or model-transformations (parameter exchange between computer-based tools).

Papers A, B and C addressed Research Questions 1 and 2 and Papers D and E will be addressing Research Questions 3 and 4. Research Questions 3 and 4 are primarily aimed at the challenge "Modelling and controlling multiple relations in the product concept" (challenge D). The assumed effect of finding sufficient answers to Research Questions 3 and 4 is that potential integration problems can be detected and handled early in the development process. Being able to design and control relations in the product's concept will bring the engineering disciplines closer when synergistic solutions must be found. Aiming at these opportunities the last two research questions are undertaken.

5.3 Paper D

Title: "Classification of Product-related Dependencies in Development of Mechatronic Products" (Torry-Smith et al. submitted 2012a)

Specific research question for this paper: What classification can be identified for dependencies appearing in a mechatronic product concept?

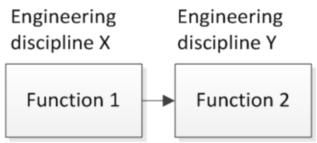
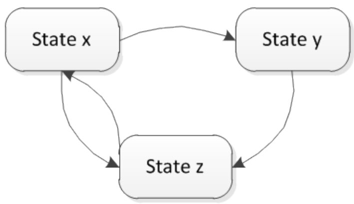
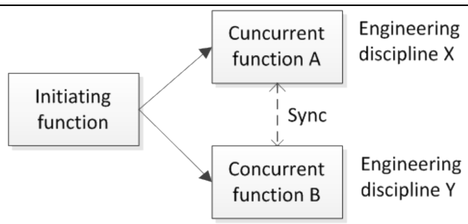
Note that the definition of 'a dependency' is stated on page 4 in this thesis. As a reminder, note as well that the term 'a product-related dependency' will just be referred to as 'a dependency' in this thesis.

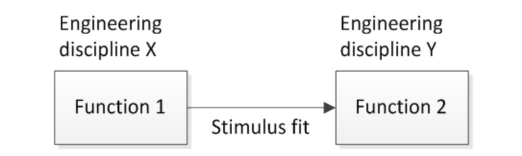
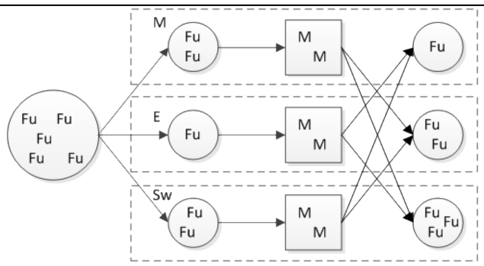
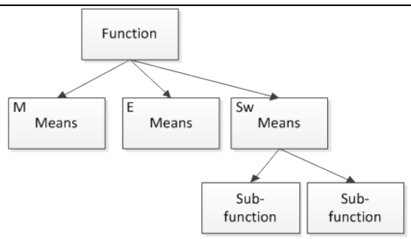
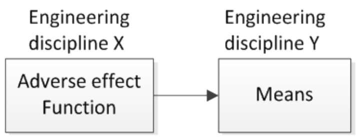
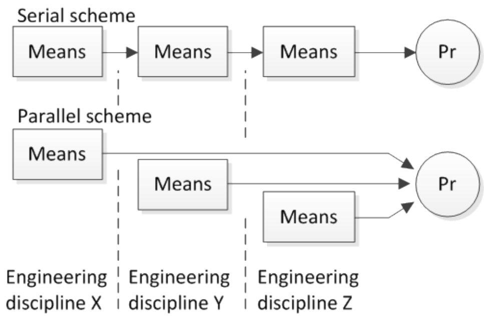
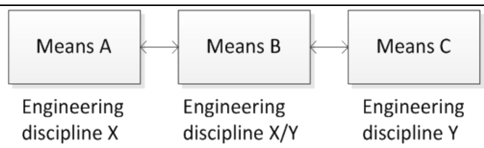
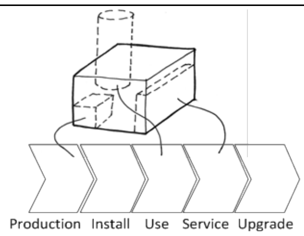
Contribution to the PhD Research Questions: The research question for Paper D is identical to Research Question 3 of the PhD thesis.

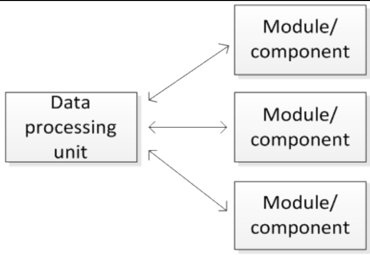
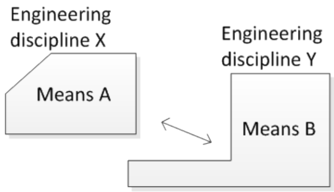
Research method: To begin with, 'a technical system' is described by use of the Theory of Domains (ToD) to establish the frame of reference and the terminology needed for describing product attributes and dependencies. Then, three cases from industry are used for identifying types of dependencies. The dependencies are grouped into a classification by use of an affinity diagram and by applying the theoretical framework from ToD. A classification of 13 types of dependencies appears. To evaluate their significance to the design process the classification is applied to an industrial development project, in which the classification is used to reveal potential harmful dependencies in the project causing delays, degraded performance of the product etc. The result is evaluated in terms of *usability*, *applicability* and *usefulness* as proposed in the DRM framework by Blessing and Chakrabarti (2009).

Results and conclusions: Based on the theoretical framework from Theory of Domains, dependencies (as defined on page 4) will appear between *functions*, *properties* and *means* in a product as a consequence of the design process. The research into the three cases from industry leads to a classification of the dependencies in which 13 dependency-types emerge. The classification is presented in **Table 7**.

Table 7 The classification of dependencies

ToD categories	Id #	Identified dependencies	Description	Illustration of the dependency
Function-function dependency	1	Causal function	Interactions between *functions* when the functionality of the product is seen as a process flow	
	2	State/time function	Dynamic relations between *functions*, where *functions* are executed at a specific time or during specific events.	
	3	Sync function	Dynamic relation between *functions* where *functions* react on stimuli and the timing of the stimuli is important	

	4	Response function	*Functions* react on stimuli from other *functions*. The size and type of the stimuli have to be matched between the *functions*.	
Function-means dependency	5	Fu-M disposition	Proposing *means* to *functions* in one domain will often have consequences in other domains in terms of supporting *functions*.	
	6	Cumulative Fu-M	The realisation of a *function* may require *means* from various disciplines.	
	7	Adverse effect	A *means* may have an adverse effect associated to it. The undesired adverse effect can be formulated as a *function* (e.g. 'create vibration').	
Property-means dependency	8	Property scheme	A single *property* of a product may have influencing factors allocated to various *means*. How these *means* contribute to the one *property* is important to clarify to optimise the product's performance.	
Means-means dependency	9	Multi-disciplinary means	Some *means* have to satisfy boundary conditions, which are important to more than one engineering discipline.	
	10	Volume allocation	Physical *means* have to be located spatially in the product and the volume may have changing restrictions during the life phases of the product.	

	11	Liveliness	The flow of information between electronics and software must be designed without causing a system-lock, which requires a cross-disciplinary effort.	
	12	Physical interface	Physical interfaces between modules and components have stakeholders from electronics and mechanical engineering.	
	13	Communication interface	Digital components may have analogue communication incorporated and analogue components may have a digital port. To ensure seamless integration, communication protocols must be evaluated and agreed upon.	

The classification is evaluated by assessing the *usability*, *applicability* and *usefulness* of utilizing the classification in a design setting in industry. In terms of the overall research project framed by the PhD project, two aspects are interesting to clarify: (i) can the 13 types of dependencies be identified in an on-going mechatronic development project and (ii) are the dependencies, as presented in the classification, significant to the design process? A development project from industry is selected for carrying out the evaluation. The project is aimed at developing an actuated hand for arthritis patients. The concept proposal suggests the use of mechanical, electronics and software solutions for the final product. **Fig. 21** is an illustration of the vision for the product used in the early phases of the development process. **Fig. 22** presents pictures of the fully operational functional model.

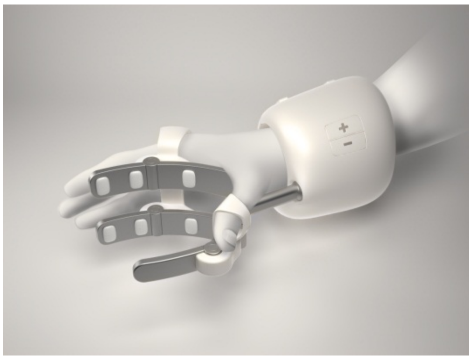


Fig. 21 The vision for the product (computer rendering)

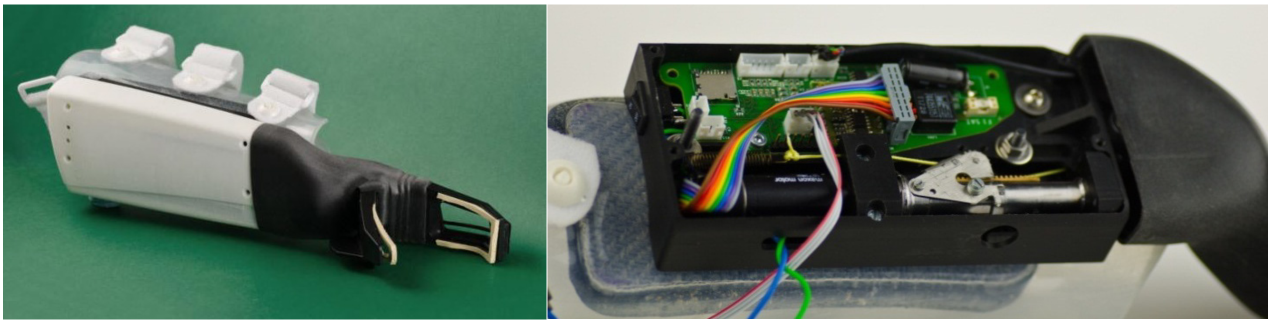


Fig. 22 The functional model of the product (fully operational)

The classification was utilised in the synthesis process of creating the concept and the functional model for the actuated hand. The classification was applied in creative sessions with the involved design engineers from the three domains. **Table 8** shows the number of revealed dependencies for each of the 13 types of dependencies.

Table 8 Number of revealed dependencies in the product concept

ToD categories	Id #	Identified dependencies	Number of revealed dependencies
Function-function dependency	1	Causal function	6
	2	State/time function	10
	3	Sync function	1
	4	Response function	2
Function-means dependency	5	Fu-M disposition	8
	6	Cumulative Fu-M	3
	7	Adverse effects	7
Property-means dependency	8	Property scheme	4
Means-means dependency	9	Multi-disciplinary means	3
	10	Volume allocation	3
	11	Liveliness	1
	12	Physical interface	3
	13	Communication interface	3

54 dependencies are revealed in the course of utilising the classification framework. Assessments of the consequence of not having identified them in due time, strongly indicate that it would impact the project in terms of delays due to rework, lack of functionality, degraded performance and quality issues of the end product. Based on the research documented in Paper D we show that a classification is possible and that the types of dependencies used in the classification can be identified in an industrial project and that they are significant to the design process. The research does not rule out the existence of more classes of dependencies. The research reveals 13 types of dependencies, which play a significant role in the three cases analysed. Further work could be aimed at consolidating the classification by targeting a larger variety of mechatronic projects.

5.4 Paper E

Title: "The Mechatronic Integration Concept" (Torry-Smith et al. submitted 2012b)

Specific research question for this paper: How can a classification of dependencies be used as a basis for modelling and describing dependencies in a mechatronic product concept?

Contribution to the PhD Research Questions: The research question for Paper E is identical with Research Question 4 of the PhD thesis.

Research method: The idea here is to further build on the knowledge created in Paper E about the classification of dependencies. The research presented in Paper E is an effort to further operationalize the use of dependencies in mechatronic development. If the use of the classification of dependencies should be integrated into the development activities, the description and modelling of the dependencies should not appear alien to the design engineers. To be able to propose a way to handle the dependencies, which will appear familiar to the design engineers, three mechatronic projects are analysed. The scope of the analysis is to obtain information about how concepts are modelled and described to obtain a shared understanding of the concept and issues related to the concept at integration meetings. The conclusion of the analysis is compared to design theory to ensure that the empirical results are in concordance with the theory. The projects used for the analysis comprise consumer products from a Danish producer of audio/visual products, plus two projects previously used in the research: The Linde Werdelin watch project and the blood sugar measurement project. Since the two latter projects are not analysed for groupings of dependencies but instead on how concepts are described at integration meetings, the use of the projects do not pose a bias to the research. As a result of the analysis and the knowledge on the classification of dependencies a *Mechatronic Integration Concept* is proposed. This concept is then tested in an industrial mechatronic project to evaluate how well dependencies can be elucidated and modelled by use of the concept. The project used for testing of the concept is the same project used in Paper D. The project is about the development of an actuated hand for an arthritis patient. Where the project was used in Paper D for verifying that the groups of dependencies could be identified and that the dependencies were significant, the project is used in Paper E for evaluating how well dependencies can be elucidated and modelled by the use of the *Mechatronic Integration Concept*.

Results and conclusions: The analysis of the three mechatronic projects with respect to how concepts are modelled to facilitate a shared understanding reveals a pattern of how concepts are modelled. Apart from a common conceptual description primarily based on functional modelling, it seems that the conceptual description is split into two. One description and modelling of the product is modelled with the aim of reflecting concerns important to the mechanical and the electronics engineers, whereas the other description and modelling of the product concept is modelled with the aim of reflecting concerns important to the electronics and the software engineers. Thus, three views are needed to describe important issues at integration meetings. The views are called: (i) the M/E/Sw view, (ii) the M/E view and (iii) the E/Sw view (where M is for mechanical, E is for electronics and Sw is for software). What is typically captured in these views based on the analysis of the projects is stated in **Table 9**.

Table 9 Overview of descriptions used in each of the three views needed for the *Mechatronic Integration Concept*

M/E/Sw view (Functional description)	M/E view (Physical structure and spatial arrangement)	E/Sw view (Data structure and signal processing)
Aspects to cover in the M/E/SW view • Task analysis for life phases • *Functions* and *function* carriers • Sequence of the *functions*	Aspects to cover in the M/E view • Spatial configuration • Connectivity between components • Force and physical effects	Aspects to cover in the E/SW view • Data and signal flow • Data structure (architecture) • Timing and sequencing
Models to describe the M/E/Sw view • Life phase scenarios • Task flow diagram with description of the technical process performed at each step including sensors and actuators involved • Function/Means tree • Finite State Machine diagram	Models to describe the M/E view • Spatial drawing or sketch of product's outer shape and MMI elements • Spatial drawing or sketch of the main components/*means* • Overview of force distribution in product or critical loads on structure. • Interface diagram containing main components and their interfaces	Models to describe the E/Sw view • Use case diagram • Data Flow Diagram showing main components and the data/signals transferred • Data structure diagram defining the architecture • Critical executable blocks modelled with pseudo code

The *Mechatronic Integration Concept* is based on these three views in combination with an overview of dependencies classified according to **Table 7**. A modelled example of the *Mechatronic Integration Concept* is illustrated in **Fig. 23**.

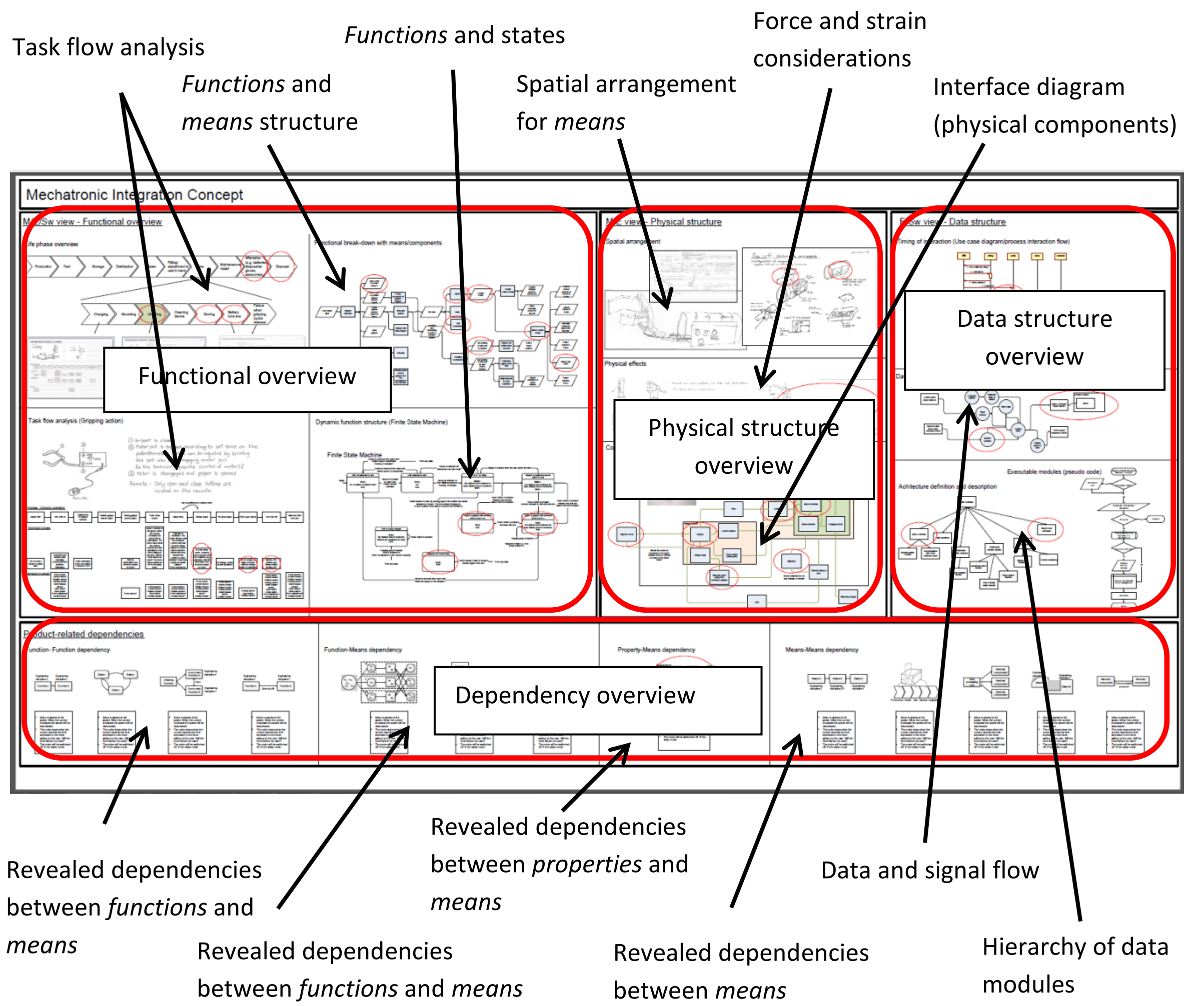


Fig. 23 A modelled example of the *Mechatronic Integration Concept*

By deploying the MIC in the development project of the actuated hand, the value of using the concept was evaluated. The MIC aided in clarifying the dependencies. The application in the industrial setting showed that it is possible to model the dependencies explicitly and that the Mechatronic Integration Concept can facilitate a cross-disciplinary discussion. Although the use of the MIC does not ensure that all relevant dependencies are revealed, the suggestion of using the MIC will form a structured process around the handling of the dependencies increasing the likelihood of revealing important dependencies early in the development process. The result from applying the concept in the project was a potential cut in the lead-time and increased efficiency of resources used, thereby pointing in the direction of being able to help reduce costs in a development project.

6 CONCLUSION

Out of the need to stay competitive or due to the prospect of creating a competitive leverage companies engage in mechatronic development. It is a daring task especially for companies entering the mechatronics arena for the first time. Many companies are overwhelmed by the increased complexity and the many dependencies, which are created between the domains in the course of the development. This PhD thesis has investigated the types of challenges encountered in the development of mechatronics and the phenomenon of 'dependencies' based on theoretical and empirical research. This section of the thesis summarises the research findings, determines the core scientific contributions, and evaluates the research performed and its impact on industry.

6.1 RESEARCH FINDINGS

The four research questions used for directing the research of the PhD project were presented in section 1.3 of the thesis. The Research Questions have been addressed by five scientific papers (Papers A to E). In paper A a case study was used to investigate challenges in the synthesis process of creating a mechatronic product and a literature study was used to shed light on proposed solutions. In Papers B and C a broader search for mechatronic-specific challenges and solutions was deployed to gain a more comprehensive overview, thus being able to answer Research Questions 1 and 2 of the PhD thesis. RQ3 and 4 were aimed at one of the identified mechatronic-specific challenges identified in Papers A to C. The challenge was 'Controlling multiple relations' and it was addressed in Paper D by searching for a classification of these relations (labelled dependencies in this thesis). The classification was established by use of three case studies. In the last paper a proposal is made for how to model and describe the dependencies in a product concept. The proposal was tested in an industrial development project.

In the following sections, the research findings are presented by going through the Research Questions one by one and providing the answers obtained from the research.

> ***RQ1: What are the central integration phenomena posing a challenge to companies when developing mechatronic products?***

The result of the systematic search on stated challenges utilizing statements from renowned researchers (Papers B and C) encapsulate the majority of the proposed aspects needing special attention in the synthesis process of mechatronics design (Paper A). The findings show that the engineering disciplines being different in nature leads to a number of challenges causing problems in the development of mechatronics. The main reason for the disciplines being different is that the fundamental theories founding the disciplines rely on different axioms. This fact, even though it appears as a scientific technicality, has some noteworthy consequences. Firstly, additional challenges appear as a consequence of the disciplines being different in nature. It also affects the prospect of finding a common language and/or common methodology, which is discussed in RQ2. The fundamental differences will drive the disciplines away from each other on an organizational level rather than bringing them closer, if the managerial behavioural pattern is not proactive in that respect (e.g. planning for cross-domain activities in projects). The research reveals a number of specific challenges evaluated to be most significant to the design of mechatronics, based on an evaluation of how often the challenges are stated by internationally prominent researchers. Although the evaluation introduces the assumption that the most significant challenges found in companies are reported equally by the international researchers, the advantage of using the described research process is that the investigation is much broader and wide ranging compared to a limited number of case studies in companies, which would have been feasible within the timeframe of a PhD project. The identified challenges range from product-related challenges to company-level challenges, and have been classified according to their affiliation to either: 'the product', 'the design activity', 'the mind-set of the design engineers', 'the competences of the design engineers', 'the organizational aspects' or to a group comprising 'other aspects'. All challenges are relevant in order to enhance the mechatronic design competence for a company. Yet, in this conclusion I would like to highlight the challenges related to 'the product' and 'the design activity', although they have not been scientifically verified to be more important than the other challenges in the context of this thesis. These are:

- Lack of a common understanding of the overall system design
- Difficulty in assessing the consequences of choosing between two alternatives
- Lack of a common language to represent a concept
- Difficulty in modelling and controlling multiple relations in the product concept
- Difficulty to control the multiple functional states of the product
- Complexity of transferring models and information between domains (expert groups)
- Difficulty to synchronize the development activities to attain concurrent engineering

Having addressed Research Question 1, we turn towards Research Question 2.

RQ2: What solutions in terms of methods, tools and mind-sets exist which can facilitate integration between the involved engineering disciplines in the development of mechatronic products?

The research reveals a number of proposed solutions to challenges encountered in the design of mechatronics. The research shows that the solutions are only partly addressing the challenges in designing mechatronics. The primary cause is due to a lack of a common methodology. Due to the lack of the common methodology the proposed methods appear as a landscape of 'islands', which are not connected. The nature of design research which potentially could be directed at bridging the island, is, however, also diverging. Design research can be based on a range of different design methodologies even for the mechanical design (Blessing and Chakrabarti 2009), and thus is not based on one founding design methodology. The fragmented landscape of methods to be used in the design of mechatronics is typically based on different founding methodologies, thus increasing the difficulty of using them together to provide a cohesive framework for the design process. The lack of a common methodology is reflected in the types of proposed solutions for describing the product concept. A strategy of coping with the absence of a common methodology emerges when comparing the available solutions. The strategy seems to favour functional modelling in the early phases of design where functional descriptions are adequate for modelling purposes. When the description of the solution becomes domain-specific and a common functional description across disciplines is no longer adequate the strategy seems to be to create transformations of models between the domains (formal or informal transformations) or to propose a number of 'views' to model in order describe the product concept adequately. Descriptions of the design process can be found but the integration activities are not described in terms of what should be integrated or how the actual process of integration should be performed. It seems that the models anticipate that by specifying a number of activities, which indicate a relation between the disciplines (e.g. 'functional specification'), the integration will appear as a natural consequence hereof. However, what should be done to ensure the integration is not to be found. When addressing Research Question 1 it has been argued in this thesis that the domains will not converge or synchronise naturally without dedicated efforts, thereby not indicating the possibility for automatic integration by the stated activities in the procedures. Deploying 'systems integration groups' seems to be used to fill in the gaps of the missing integration activities described in the procedures. The role of the 'system integration group' appears vaguely described in literature and tends to rely more on experience of the practitioners of 'doing things right'. Thus the results can vary greatly from team to team. Finally, specification techniques seem to be used for mapping and maintaining agreements across disciplines. Matrix-based dependency management via e.g. DSM and specification of interfaces and traditional requirement management belong to this group. However, specification techniques have a strong relation to the documentation activity, even though it can be used as processes in-between and as support for the synthesis activities. Viewing all the proposed methods they tend to support analysis rather than synthesis activities. Support for mechatronic design can be found in the proposed methods; however, the lack of a common methodology results in a need for further development of the methods and a need for possible new methods, tools and mind-sets.

RQ3: What classification can be identified for dependencies appearing in a mechatronic product concept?

By addressing Research Question 3 we zoom in on the stated challenge: 'Modelling and controlling multiple relations in the product concept'. Based on empirical findings while leaning on Theory of Domains for adequate terminology for describing a design, a classification of dependencies is established. 'Dependencies' are often seen to cause difficulties in the design of mechatronics (Tomiyama et al. 2007; Felgen et al. 2005; Gausemeier et al. 2008); however, a further definition of a dependency or further characterization or classification has not been revealed in the reviewed literature for this research. Claiming that dependencies are problematic without providing a characterisation or a classification is peculiar but also offers room for improvement. Thus, a categorisation and definition of a dependency is a first step towards dealing, more systematically, with dependencies in the development of mechatronics. Thirteen types of dependencies are identified based on empirical research carried out in this research project. Based on the definition of 'a dependency' (see page 4), it is a relation in the product concept, which appears as a consequence of the designer being involved in the creative process of designing. The thirteen dependencies are labelled as follows (see **Table 7** for a more detailed description):

1. Causal function
2. State/time function
3. Sync function
4. Response function
5. Fu-M disposition
6. Cumulative Fu-M
7. Adverse effects
8. Property scheme
9. Multi-disciplinary means
10. Volume allocation
11. Liveliness
12. Physical interface
13. Communication interface

The three product attributes obtained from the Theory of Domains: *function*, *property* and *means* become essential in the description of the dependencies. A dependency is characterised in this research as appearing between two product attributes. A combinatorial exercise will suggest a dependency between a *function* and a *property* as well as between two *properties*. However, due to the precondition that we view one product concept and do not compare alternative concepts there are no direct relations between two *properties*. *Properties* as well as *functions* are mediated through *means* in a product concept, which is the second reason why a dependency as it is defined cannot appear between two *properties* (see illustration in **Fig. 24**). Furthermore, it is also the reason why a dependency cannot appear between a *function* and a *property*. *Functions* and *properties* will always be mediated through a *means*.

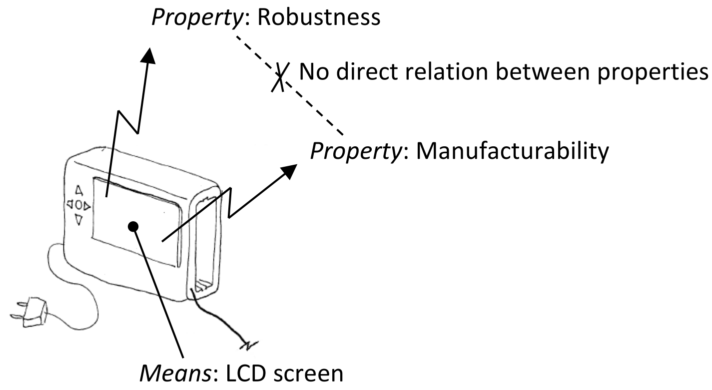


Fig. 24 No direct relation exists between two *properties*. *Properties* are mediated through a *means* (the LCD screen)

The proposed classification was evaluated in an industrial mechatronic project and the evaluation shows that it is possible to identify dependencies according to the proposed classification and that the dependencies are significant to the design process. The classification can provide guidance and structure to the management of dependencies. However, only a classification has been proposed so far. So in the search of operationalizing the use of the classification the last Research Question is formulated as:

RQ4: How can a classification of dependencies be used as a basis for modelling and describing dependencies in a mechatronic product concept?

The research shows that the classification of dependencies can be utilized in a design setting by creating a so-called Mechatronic Integration Concept. The concept is capable of capturing the information particularly important to share between the mechanical, the electronics and the software engineering disciplines. By use of empirical research, preferred descriptions of a product concept to be used for integration meetings for discussing dependencies are revealed. The views are systematized into three views labelled M/E/SW view, M/E view and E/Sw view (M, E and Sw are abbreviations for mechanical, electronics and software respectively). A fourth view contains an overview of revealed dependencies grouped according to the classification, and the four views comprise the Mechatronic Integration Concept. By using the Mechatronic Integration Concept the stated dependencies can be modelled and the involved and affected engineering disciplines can reach a clarification of each dependency.

A classification enables the designer in a systematic way to reveal dependencies and the Mechatronic Integration Concept provides the basis for representing, modelling and clarifying the dependencies to facilitate cross-domain integration meetings with the aim of addressing the dependencies. Uncontrolled dependencies appearing randomly in the design process can lead to the design process being perceived as complex. However, it seems that if dependencies are consciously controlled and manipulated through the design process, the perceived complexity of the task will therefore be reduced. The evaluation of the proposed Mechatronic Integration Concept, performed in an industrial setting, points in the direction that using it has potential to reduce lead time and decrease rework.

Design Structure Matrices (e.g. Braun and Lindemann (2007)) and Domain Mapping Matrices (e.g. Danilovic and Browning (2007)) are other means for modelling dependencies. Yet, they are different compared to the Mechatronic Integration Concept in a number of areas. DSM and DMM only deal with high level dependencies such as *function-function* dependencies and *function-means* dependencies. Literature on DSM and DMM do not offer a further classification nor has it been found how a dependency is to be understood in relation to the design activity. DSM offers the advantages of algorithmic optimization to show affinity between elements, but the drawback is the need for a matrix-based representation of the dependencies, which renders even a simple product with few components difficult to grasp in its totality. The reason is the information representation, which for a product comprising 10 components will generate 100 data-fields linking text strings without visual cues to grasp its totality. The Mechatronic integration Concept offers a visual representation of both the concept and the dependencies, which facilitates and supports the discussion among the engineers. Where the Mechatronic Integration Concept is aimed at the synthesis process the DSM and the DMM are essentially analysis activities.

6.2 CORE CONTRIBUTIONS

The main finding of the research performed in this PhD project is the established classification of dependencies and the proposal for the Mechatronic Integration Concept, which enable the designers to identify and handle potential harmful dependencies in due time. The stepping stones for reaching these main findings are summarized in the following and are considered as the core contributions of this research.

- Clarification of main challenges within the design of mechatronic products along with a description of proposed solutions to the challenges.
- Definition of the term 'product-related dependency'.
- Classification of dependency types.
- Suggestion for the content of a '*Mechatronic Integration Concept*' to describe and model dependencies for a mechatronic product.
- Proposal of a visual representation of a '*Mechatronic Integration Concept*' to facilitate a shared understanding of the dependencies across domains at integration meetings.

6.3 EVALUATION OF THE RESEARCH

As cited above the evaluation of the research is conducted by evaluating the usability, applicability and the usefulness (Blessing and Chakrabarti 2009) of the proposed classification (RQ 3) and the proposed Mechatronic Integration Concept (RQ 4). An evaluation of the three aspects is best suited for results which can be applied in a design setting. The results obtained from investigations of the integration phenomenon (RQ1) and the proposed solutions (RQ2) will be evaluated in terms of generality, validity and completeness of the findings.

6.3.1 RESEARCH CONDUCTED ON PAPERS A, B AND C

The results obtained from investigating the integration phenomenon in Papers B and C is based on a broad analysis of the insights from researchers at an international level, which is considered to reveal results that are *general* in nature. In evaluating the *validity*, it has to be considered that the statements regarding the integration phenomenon are obtained from researchers and not directly from companies. Thus the findings are valid to the extent that the researchers are reporting on actual challenges related to the integration phenomenon experienced by companies. The stated challenges are reported on by internationally prominent researchers who favour the *validity*. Furthermore, 3-5 papers from each of the investigated researchers in **Table 6** have been reviewed, and from these papers statements regarding challenges were obtained. This provides a broad basis (vast amount of statements) for creating the list of 19 challenges by use of an affinity diagram (see **Table 6**). However, these two circumstances are only indicators and cannot be considered as proof of *validity*. The research performed in Paper A on the integration phenomenon is different in terms of *generality* and *validity*. The case study in Paper A exemplifies challenges related to the integration phenomenon, thereby underlining the *validity* of the described challenges for that particular case. The generality aspect is influenced by the authors' ability to select the important challenges based on their experience with mechatronic development. No proof of the general applicability is thus presented in Paper A. In this sense Paper A has an emphasis on *validity*, whereas Papers B and C have emphasis on g*enerality* and to a certain extent the *validity.* When comparing the stated challenges in paper A with the findings in Papers B and C five of the seven aspects are represented in the findings from Papers B and C indicating the *generality* of the selected seven challenges in Paper A. In terms of completeness it is not claimed that the seven challenges from Paper A constitute a complete picture of challenges, and compared to the stated challenges from Papers B and C the seven aspects, although considered important, are not complete. In Papers B and C the stated challenges are headlines for groups of challenges stated by the researchers in published scientific work. The groupings of the challenges into the 19 categories in **Table 6**, render the overview of challenges more complete compared to Paper A, but it cannot be concluded that all challenges are captured by the research carried out in Papers B and C.

The results of investigating solutions are obtained from literature reviews. Sources for suggested solutions comprise papers by the most cited researchers, from the mechatronics design community, in which state of the art solutions have been described. In addition, other publications, which the authors of Papers B and C were aware of, were reviewed. Papers A and B have been peer-reviewed and published in conference proceedings and Paper C has been peer-reviewed and published in the Journal of Mechanical Design. As a part of the peer-review for the journal, the completeness of the found solutions was specifically evaluated. In this process an additional solution was suggested to be added. Therefore, the completeness of the described solutions appears justified. Regarding the evaluation of how well each solution would cover the identified challenges related to the integration phenomenon, we had to rely on reviews of the solutions. Due to the timeframe for Papers B and C as well as for the research project it was not possible to test the solutions in an actual design setting. However, the evaluation of the coverage of the solutions was conducted by viewing the documented effect in papers describing the effect of using the proposed methods. Thus the evaluation of the effect of applying the solutions relies on conducted experiments by other researchers.

6.3.2 Research conducted on Papers D and E

The results obtained from Papers D and E are evaluated based on the following aspects obtained from Blessing and Chakrabarti (2009):

- Usability: the ease with which the method can be used for the intended task;
- Applicability: whether the method has the direct intended effect on the design process;
- Usefulness: Whether the introduction of the method leads to an overall success in the project measured on a number of parameters taking into account possible uncontrollable influencing factors.

The classification of dependencies and the Mechatronic Integration Concept are evaluated by testing the results in a development project in an industrial context. Though the test is performed in an industrial project, the evaluation falls into the category of an *initial evaluation* according to Blessing and Chakrabarti (2009), due to the extent to which it has been evaluated. A comprehensive evaluation would e.g. require the use of different evaluation methods to be combined in a triangulation evaluation study; thus also being far more time consuming. For both an *initial* and a *comprehensive evaluation* the three aspects are suggested to be evaluated (Blessing and Chakrabarti 2009). The *usability* and the *applicability* are considered as an *Application Evaluation* and the *usefulness* is considered a *Success Evaluation* according to Blessing and Chakrabarti (2009). To apply the classification to the development project, each of the 13 identified types of dependencies was applied by formulating questions covering the dependency. The questions were both introduced to the designers at integration meetings and in separate one-to-one sessions with the designers. It led to constructive discussions in which more than 50 dependencies were revealed. By using questions for each dependency, the classification was utilized in the project and the *usability* criterion was fulfilled. Since it was possible to reveal a number of dependencies the utilization had the intended direct effect. In terms of the *Success Evaluation*, examples were presented in the research of the presumed effect of not having discovered the dependencies in due time. Since the classification was applied in an actual project, the team had to react to the information on dependencies motivating them to make changes on the design as soon as the dependencies were revealed. Therefore, the usefulness is addressed by logical reasoning of consequences of not having addressed the dependencies, but that was the consequence of seizing the opportunity of testing the classification in an industrial development project.

The Mechatronic Integration Concept was also tested in an industrial project and then evaluated. It was applied in the same project in which the classification was tested. However, the test of the classification was aimed at verifying if the dependencies could be revealed in the project and evaluating the importance of the dependencies to the design process. This is different from the test of the Mechatronic Integration Concept, which is aimed at evaluating if the dependencies can be described and modelled by the use of it. In the creation of the Mechatronic Integration Concept the usability was addressed by the investigation of using conceptual descriptions in integration meetings. This is incorporated into the design of the mechatronic integration concept to build on familiar conceptual representations. The dependencies are applied as short statements (e.g. "A mechanical quick-release may require extra electronics to detect activation of it"). This way of expressing the dependencies appeared logical for the involved designers in the project. The elucidation and modelling of the dependencies is exemplified in Paper E. Modelling the dependencies as exemplified poses a challenge for how many dependencies can be modelled at the same time on the same concept. If the participants were using a hard copy of the Mechatronic Integration Concept printed on a poster, additional posters might be used. If the discussion is formed around a 'smart board' layers can be used to create space for the modelling. The usability has been assessed, and though there may be possibility for improvements in terms of modelling several dependencies on the same 'poster', the Mechatronic Integration Concept was intuitive to use at the integration meetings. The usability is thus evaluated to be satisfactory. The intended effect from applying the mechatronic integration concept was a clarification of the dependencies based on the description and the modelling of them. Examples are stated in Paper E of the clarification achieved by modelling the dependencies in the Mechatronic Integration Concept. Since clarifications of dependencies are achieved, it is evaluated that the aspect of applicability can be claimed. The positive effects in terms of less rework, shorter lead-time and increased performance of the product have been argued for in Paper E, which indicates the potential for claiming the achievement of the aspect; *usefulness*. The evaluation of the usefulness is based on the same conditions as the usefulness of applying the classification in a design setting. Having chosen a design setting from industry the revealed dependencies have to be acted upon and thus omits the possibility of proving what would happen if the dependencies were not revealed in due time. Therefore the evaluation of the *usefulness* is based upon logical reasoning of the consequences in the case that the individual dependencies had not been discovered early on in the project.

The use of an actual development project for the evaluation also influences the possibilities of controlling 'uncontrollable influencing factors'. The term is directed at factors which might influence the evaluation, so the usefulness would be perceived better or worse than it actually is. Since the usefulness is based on logical reasoning, a bias in the evaluation is an example of a possible *influencing factor*. This risk has been mitigated by stating concrete examples in both papers of likely events in the case of not addressing the dependencies in due time. A possible *influencing factor* which might affect the evaluation negatively, might be the time allocated in the project for revealing the dependencies. If too short a time has been set aside for revealing the dependencies, fewer dependencies would have been identified, thus rendering the *usefulness* as being less than its full potential. For this particular aspect, the answer will always be that more time can be set aside for revealing more dependencies. Yet, it reflects normal prioritising between activities in a development project and might even render the evaluation more realistic. Though, I evaluate that the time spent on revealing and modelling the dependencies (approximately 30 man-hours in total) was adequate for the activity and in balance with the other activities performed in the project. As mentioned earlier in the section the possibility of controlling *influencing factors* is affected by the choice of performing the evaluation in an industrial setting. The *usability* and *applicability* can be obtained by observation whereas the *usefulness* has been evaluated by logical reasoning based on examples and experience of such events by the involved designers. A controlled study in a simulated development environment could contribute to confine possible *influencing factors*, which then poses the challenge of rendering the simulated environment as close as possible to the real design practice. Using both types for the evaluation would strengthen the evaluation since it would reveal additional information about the use of the classification and the Mechatronic Integration Concept.

6.4 INDUSTRIAL IMPACT

The conclusion of the thesis implies that the classification of dependencies enables the designer to identify and handle potential harmful dependencies in due time, minimizing the risk of rework and risk of prolonged lead-time. The conclusion is based on testing and evaluating the results in a design setting from industry, which showed good and promising results. The industrial setting along with the good results indicates that the scientific contribution presented in this thesis has industrial impact.

In an industrial development project one might ask how central and important it will be to allocate resources to create the Mechatronic Integration Concept compared to all the other activities that are needed to be performed. Projects are notoriously under-resourced and too little time is available before the next deadline for the planned activities. Adding in extra activities will require that the benefit from performing the activity not only justifies the time spent on the activity, but that possible benefits of having spent the same amount of time on alternative activities should be taken into account as well. In the final product the electronic components will be co-located with mechanical components and software code will be embedded in electronics components. *Functions* within each of the domains will have to interact to realise the main *function* of the product. The integration is thus required and if the integration fails due to lack of attention to dependencies, so does the product. It should be a constant goal in product development to ensure the right functionality, components that match and *properties* fulfilling the requirements. Without attention to the dependencies in mechatronics development, it will jeopardize the success of the development. Thus attention to dependencies must always be present in a mechatronic project whether or not the dependencies are handled inconsistently and informally or handled systematically and treated formally. As argued previously in this thesis 'integration' will not happen without considerable effort. Here, the classification of dependencies and the Mechatronic Integration Concept provide a systematic and formal way to reveal and handle dependencies when developing mechatronic products.

In the following I will present my considerations about how to deploy the scientific contribution from this PhD project on dependencies in industrial projects. The considerations relate to the following aspects: the type of project, when should it be applied, who should drive the process and who should participate, how should the process be and will it inflict on other commonly used methods in product development?

What types of projects are suited for the deployment of the Mechatronic Integration concept? In principle it should be possible to apply it to various types of projects as long as the three domains (M, E and Sw) or at least mechanical and electronics engineering are represented in the project. If more than one domain is present, dependencies will appear in the product concept as argued earlier in the thesis. Thus, the project will benefit from clarifying the dependencies. Since dependencies appear in mechatronic development projects the classification and the Mechatronic Integration Concept should be applicable to a wide range of them. Yet, it must be expected that the practise of using the classification and the Mechatronic Integration Concept will differ and will be adjusted to fit the individual project. In one project the classification might be applied as a mind-set only when designing or as a checklist, whereas the dependency might be revealed systematically and modelled explicitly in another project.

When should the Mechatronic Integration Concept be applied? Some projects may have the development of mechanics, electronics and software to run in parallel, while others may have them done sequentially (see **Fig. 25**). A common point for the two configurations is a prerequisite that the project plan should favour the possibility of collaboration between the domains even in the case where the development is closer to a serial than a parallel configuration. Depending on the project set-up based on either of the two configurations the Mechatronic Integration Concept might be utilised differently. If the configuration is of the parallel type the creation of the Mechatronic Integration Concept can be used as baseline descriptions needed at e.g. decision points or gates between project phases. It can also be used as a continuous modelling of the dependencies in the product concept as the concept evolves and the detail level increases (see **Fig. 25**). In this case the Mechatronic Integration Concept can be used to avoid integration problems as well as obtain synergy in the solution finding. In the case where the configuration is serial, base lines are harder to establish because development is on-going at the same time for the three domains. Instead, a continuous modelling of the product concept in the form of the Mechatronic Integration Concept can be achieved to reveal dependencies as early as possible. In addition, it might be advantageous to create the Mechatronic Integration Concept in the transition between two domains; e.g. where the product concept is handed from the mechanical development to the electronics development (see **Fig. 25**). In the case of a serial configuration the main objective would be to avoid integration problems and to a lesser degree to obtain synergistic solution finding.

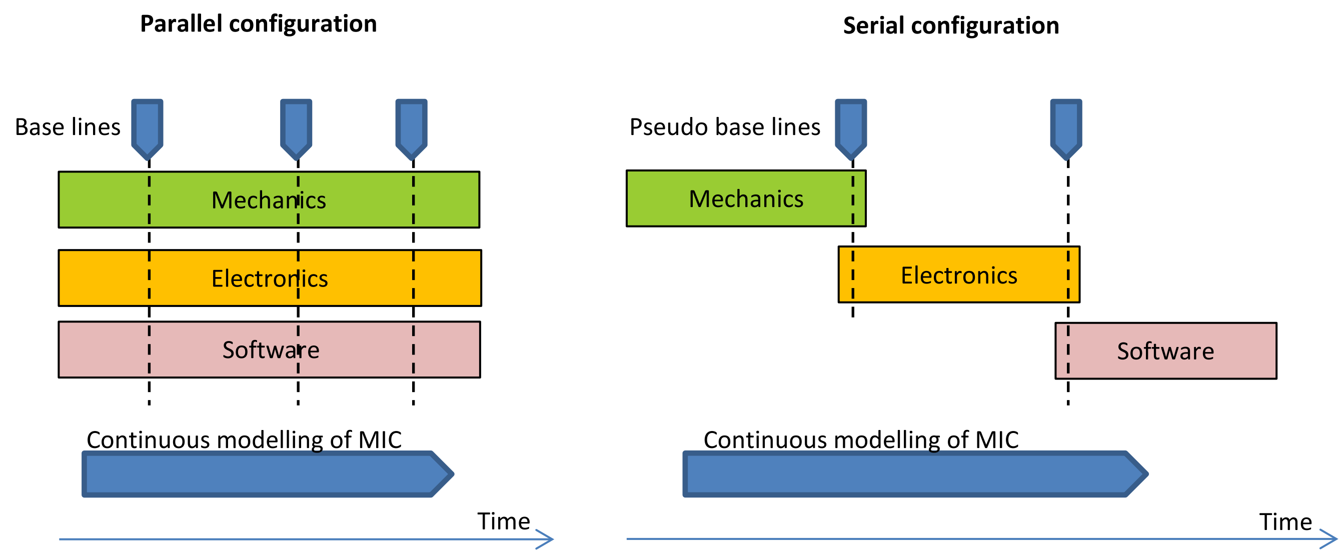


Fig. 25 Parallel and serial configuration of the development tracks (mechanics, electronics and software) including possible base lines and 'continuous modelling' utilizing the *Mechatronic Integration Concept*

Who should drive the process? If the project organisation is not familiar with the concepts of handling dependencies a person would be needed to drive the process with good understanding of the dependency phenomenon and the integration phenomenon. If a 'systems integration group', such as described in paper C, has been established in the project, the dependency handling (use of the Mechatronic Integration Concept) would naturally be nested in this group. In smaller projects resolving integration issues are in the hands of the project manager and they would be in charge of driving the process of finding and clarifying the dependencies. This would require training of the manager within the use of the framework.

Who should be included in the process and how should the process be? The person driving the process would need to set up sessions with selected team members where a few team members are gathered to discuss the content of the product concept and possible integration issues. At least two of the three disciplines should be represented at these sessions. Too many team members would probably not make the process cost-efficient. Thus aiming for 2 to 5 team members for these sessions would be ideal in terms of resources spent versus diversity of the knowledge in the group, but it will of course rely on the type of project and the situation. In addition it might require separate discussion sessions with only one team member to clarify some specific issues regarding the product concept or the dependencies. However, the main drive in this process is obtained when engineers from the different disciplines meet and discuss these integration issues. When an overview has been created, lead-engineers from the different domains would gather to resolve the revealed dependencies and associated issues.

Will the use of the 'frame-work' conflict with other methods or project models such as a stage-gate model? The strategy with the Mechatronic Integration Concept has been to build it on already used and familiar domain-specific models. Due to the content and thus the possibility of re-using conceptual descriptions from the domains to create the Mechatronic Integration Concept, the deployment of the Mechatronic Integration Concept will not conflict with other models used in the development. In contrast, a design practice where models and sketches are used will support the deployment of the Mechatronic Integration Concept. Two methods closely related to dependency handling need a closer description in terms of the investigation of possible 'conflicts'. These are the DSM/DMM method and the SysML language. If a DSM has been established it can be used to reveal some of the dependencies. However, the Mechatronic Integration Concept should be seen as an alternative to the more cumbersome method, which DMM and DSM represent. If a project is utilizing the SysML language to capture the product concept the found dependencies from the Mechatronic Integration Concept can be included and modelled in the SysML model. However, since this discussion is on possible 'conflicts' in the design practise found in companies, it must be noted that especially DSM/DMM and also SysML must be considered as experimental methods not fully adopted by the industry yet. As an exception, SysML seems to have been adopted in companies where the development is heavily influenced by the software engineering discipline and/or where control engineering plays a central role in obtaining the competitive advantage (Qamar et al. 2009).

The Mechatronic Integration Concept presents an agile and cost-effective way of modelling dependencies for two reasons: firstly, conceptual descriptions normally used within each of the engineering disciplines can be re-used when creating the M/E/Sw-view, the M/E-view and the E/Sw-view of the Mechatronic Integration Concept. Secondly, the effort in revealing dependencies in a product concept can be scaled to fit the needs in a project. The classification can be used as a simple check-list for guiding a development team through a series of types of dependencies to look for on an informal basis. Alternatively, the classification can be deployed 'full scale', where all dependencies are modelled and captured in the Mechatronic Integration Concept. The scalability facilitates a tailoring of the classification and the Mechatronic Integration Concept needed to fit different types of projects.

As it is with other mind-sets and methods, they will not be adopted by a company without considerable efforts. To integrate a mind-set or method into a company will typically require a dedicated person; a so-called 'ambassador'. He would most likely need to prove the benefits in a pilot project before moving on to implement it to the portfolio of development projects within the company. Thus, the scientific contribution presented in this thesis has yielded positive effects in the initial testing in industry and the further potential has been argued for; but it will require an undeniable effort to implement it in companies.

7 Suggestions for further research

The suggestions for further research have been divided into two sections. The first section is about the continuation of the research within management of dependencies. The second section is about further research in the area of mechatronics seen from a broader perspective. It comprises suggestions for further work on process models related to dependencies and process models related to the generic synthesis of mechatronic products.

7.1 Framing the work on finding dependencies

- **A continuation of understanding the dependency-phenomenon:** Handling dependencies in the design process is aimed at facilitating integration between the domains. There seems to be some intermediate purposes we aim for when we work with the dependencies. Some dependencies seem to be aimed at highlighting areas in the product which require continuous collaboration between the domains. An example is the dependency ‘Multi-disciplinary means’. In the case of a DC motor in a linear actuator design there is a need for continuous collaboration between the electronics and mechanical engineering disciplines. Other dependencies seem to be aimed at creating a clear cut between the domains until the next integration test. Two examples are the dependencies ‘Response function’ and ‘Physical interfaces’, which will allow an agreement ‘to freeze’ some of the design parameters to simplify the design task. However, if the dependencies are not controlled the integration will fail at the integration test. A further investigation into the phenomenon of dependencies would add to the understanding and strengthen the scientific basis for describing them.
- **On the process of how to reveal dependencies:** When the dependencies were revealed in the industrial project questions were used representing each of the 13 types of dependencies. As a reflection on the process there might be a pattern in terms of a sequence, in which the types of dependencies could be revealed. In addition some conceptual models may be better than others for supporting the discussion aimed at revealing a certain type of the dependencies. In Appendix A a suggestion of the procedure for revealing dependencies is presented. In Appendix B a suggestion of which conceptual models to use for identifying each of the dependencies is illustrated. The conceptual models suggested originate from the models contained in the Mechatronic Integration Concept. Whether or not the use of the procedure and the use of the models as support have an enhanced effect in identifying the dependencies (e.g. if dependencies are found faster with less effort or if more dependencies can be identified) is not known yet. Thus, it would be an interesting research topic.
- **On the process of clarifying the identified dependencies:** Focus was on the elucidation and modelling of dependencies when testing the Mechatronic Integration Concept in the industrial project. The aim was to enable the designers to clarify the dependencies. In the project the dependencies were clarified by the designers due to their competences and their experience with problem solving. A question that would be interesting to find the answer to is: Is there a pattern for how to clarify identified dependencies? If there is a pattern, it would add to the systematics in handling dependencies in projects. An assumption to be tested by research could be that dependencies are clarified by use of one of the following principles: (i) the dependency is accepted as is and is monitored as long as it serves a purpose (ii) design changes are performed to change the relations in a dependency to find the ‘best solution’ according to the situation (iii) The design is changed to eliminate the identified dependency.
- **On the prediction of the maturity level for finding ‘all dependencies’ in a product concept:** When should we stop to look for dependencies in a product concept? This is an open question, but the answer is extremely interesting. In Paper E it is argued that the maturity level is hard to determine and that the situation can be compared to solution-finding, where you never can be sure if you have found all relevant solutions. The mitigation within solution-finding is to use structured methods in addition to unstructured creative sessions. At the current state of research it is evaluated to be similar to the phenomenon of solution-finding. We support the likelihood of finding relevant dependencies by providing systematics in terms of a classification of types of dependencies and a way to model them by use of the Mechatronic Integration Concept. However, if further research could provide an answer

to how the maturity-level of the found dependencies can be made quantifiable, companies would be immensely interested in acquiring the results. Also a quantifiable indication of how important (critical) a given dependency is would be of great interest to companies.

- **The use of the Mechatronic Integration Concept at integration meetings:** The discussions of dependencies at integration meetings were facilitated by use of a paper based version of the Mechatronic Integration Concept. The suitability of different media to be used to show the Mechatronic Integration Concept could be subjected to research. An alternative to a hard copy version of the concept could be a projected version on a whiteboard on which the designers could model each dependency and the modelling could be captured by 'camera snap-shots'. Alternatively, 'smart boards' could be used where the drawing on the concept is recorded digitally and new layers can be added. The medium chosen for the Mechatronic Integration Concept, might affect the dynamics of using and modifying the concept. Investigations into the effects of the medium chosen to create the Mechatronic Integration Concept could enhance the use of it.

7.2 RELATING THE WORK ON DEPENDENCIES TO PROCESS MODELS OF MECHATRONIC DESIGN

Seen in a broader perspective the knowledge of the phenomenon 'dependencies' could be linked to process models describing activities in the development of mechatronics. Since dependencies are aimed at relations in the product between the domains, it would be interesting to direct research towards what point in the development integration baselines are advantageous to make. If for example the electronics team has proposed solutions to a given product and are reluctant to change what they have achieved at a point where mechanical and software engineers have not considered their contribution to the overall functionality, potential synergy may be lost in the investigation specification of *functions* for the product. It can be called 'Windows of integration', and if one discipline moves ahead these 'windows' might be missed. The question is then where and between which activities 'Windows of integration' appear? The illustration in Appendix C serves as a visualisation exercise for a procedural model describing design activities for mechatronic development in which possibilities for integration meetings are marked. It should only be used as inspiration, since it is procured from my accumulated knowledge gained from the PhD project in combination with my experience of 8 years in industry. As such it has not been established by use of a scientific method.

If we widen the scope for further research, a more detailed process model used for pointing out possible integration meetings and synchronization of activities, would be of high value to research as well as for practitioners doing design. Mechatronic-specific process models have been investigated thoroughly in Papers A-C, however they are often high level descriptions of the design activities. Examples are the VDI2206 (Association of German Engineers 2004) and Systems Engineering (Blanchard and Fabrycky 1998). Thus, a detailed mapping of activities within each of the engineering disciplines: mechanics, electronics and software, would be a step in the right direction to identify activities to align in order to facilitate integration meetings. Appendix D contains a list of detailed development activities within each of the three engineering disciplines. The suggestion for the activities within mechanical and software engineering is a mix of prescriptive and descriptive models from literature combined with my own experience of performing mechanical development. The activities for electronics development have been synthesized based on interviews with three electronics engineers. Therefore, the activities listed in Appendix D should only be used as inspiration. Rigorous research into mapping design activities in mechatronic development would be of high value to the mechatronics community, and based on my experience of models found in literature I would suggest making such a mapping based on empirical studies in companies.

8 CONCLUDING REMARKS

The PhD has been a remarkable journey for me. The endeavour into design research brought new insights to knowledge areas of product development, which I thought I knew well due to my experience from industry. Today I can look back and see that the process, even though it was directed at design science, has changed me as a person. However, it is not about me and how I perceive the course of events. It is about you and how you perceive the work I have presented in this thesis. I hope that you have found the reading as interesting as I have found it interesting to perform the research and report on the findings.

9 References

Adamsson N (2004) Model-based Development of Mechatronic Systems - Reducing the Gap Between Competencies? In: Tools and Methods of Competitive Engineering, vol 2. pp 405-413

Albers A, Braun A, Sadowski E, Wynn DC, Wyatt DF, Clarkson PJ (2011) System Architecture Modeling in a Software Tool Based on the Contact and Channel Approach (C&C-A). Journal of Mechanical Design 133 (10):101006-101008

Albers A, Ottnad J (2010) Integrated Structural and Controller Optimization in Dynamic Mechatronic Systems. Journal of Mechanical Design 132 (4):041008-041008

Andreasen MM (1980) Machine Design Methods Based on Systematic Approach. PhD thesis, Lund University, Sweden

Andreasen MM (1998) Conceptual design capture. Proceedings of EDC 98

Andreasen MM (2005) Functional Reasoning in Mechatronic Design, Lecture notes. Lecture notes edn., Lyngby

Andreasen MM (2007) How to spell a product?, Lecture notes. TU Ilmenau

Andreasen MM (2009) Complexity of Industrial Practice and Design Research Contributions – We Need Consolidation. In: Meerkamm H (ed) 20. Symposium Design for X, Neukirchen. pp 1-9

Andreasen MM, Hein L (1987) Integrated product development. IFS, Bedford

Andreasen MM, Hein L, Kirkegård L, Sant K (1989) Udviklingsfunktionen - Basis for fornyelse. Jernets Arbejdsgiverforening, København

Andreasen MM, McAloone TC (2001) "Joining Three Heads" - Experiences From Mechatronic Projects. In: 12th Design For X Symposium, Neukirchen. pp 151-156

Andreasen MM, Olesen J (1990) The concept of dispositions. Journal of Engineering Design 1 (1):17-36

Ashby WR (1956) An introduction to cybernetics. Taylor & Francis, London

Association of German Engineers (2004) VDI 2206, Design Methodology for Mechatronic Systems. VDI Guidelines. Beuth Verlag, Berlin

Bishop RH (2002) The mechatronics handbook. CRC Press, Boca Raton, Fla. ; London

Blanchard BS, Fabrycky WJ (1998) Systems engineering and analysis. 3rd edn. Prentice-Hall, New Jersey

Blessing LTM, Chakrabarti A (2009) DRM, a Design Research Methodology. Springer, London

Boehm BW (1984) Software engineering economics. Software Engineering, IEEE Transactions on (1):4-21

Bonnema GM (2011) Insight, Innovation, and the Big Picture in System Design. Systems Engineering 14 (3):223-238. doi:http://dx.doi.org/10.1002/sys.20174

Borches PD, Bonnema GM (2010) A3 Architecture Overviews - Focussing Architectural Knowledge to Support Evolution of Complex Systems. Paper presented at the 20th Annual INCOSE Symposium (IS2010), Chicago, USA,

Bracewell R, Sharpe J (1996) Functional descriptions used in computer support for qualitative scheme generation-'Schemebuilder'. Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM 10 (4):333-345

Bracewell RH, Bradley DA, Chaplin RV, Langdon PM (1993) Schemebuilder, A design aid for the conceptual stages of product design. In: International Conference on Engineering Design (ICED'93). pp 1311-1318

Bradley D (2010) Mechatronics - More Questions Than Answers. Mechatronics 20 (8):827-841

Braun SC, Lindemann U (2007) A Multilayer Approach for Early Cost Estimation of Mechatronical Products. In: International Conference on Engineering Design (ICED'07), Paris. pp 187-188

Buur J (1989) A framework for mechatronics design methodology. In: International Conference on Engineering Design (ICED'89), Harrogate.

Buur J (1990) A Theoretical Approach to Mechatronics Design. PhD thesis, Technical University of Denmark, Denmark

Cabrera AAA, Foeken MJ, Tekin OA, Woestenenk K, Erden MS, De Schutter B, van Tooren MJL, Babuška R, van Houten FJAM, Tomiyama T (2010) Towards Automation of Control Software: A Review of Challenges in Mechatronic Design. Mechatronics 20 (8):876-886

Cabrera AAA, Woestenenk K, Tomiyama T (2011) An Architecture Model to Support Cooperative Design for Mechatronic Products: A Control Design Case. Mechatronics 21 (3):534-547

CMS (2007) SELECTING A DEVELOPMENT APPROACH. Department of Health & Human Services/Centers for Medicare& Medicaid Services. http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf.

Cooper RG (2001) Winning at new products : accelerating the process from idea to launch. 3rd ed. edn. Perseus, Cambridge, Massachusetts

Danilovic M, Browning TR (2007) Managing Complex Product Development Projects With Design Structure Matrices and Domain Mapping Matrices. International Journal of Project Management 25 (3):300-314

Felgen L, Deubzer F, Lindemann U (2005) Complexity Management During The Analysis of Mechatronic Systems. In: International Conference on Engineering Design (ICED'05), Melbourne. p 409

Fenves SJ (2001) A Core Product Model for Representing Design Information. National Institute of Standards and Technology, Gaithersburg, MD 20899, USA,

Gausemeier J, Flath M, Mohringer S (2001) Conceptual Design of Mechatronic Systems Supported by Semi-formal Specification. In: IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Como, Italy. pp 888-892

Gausemeier J, Frank U, Donoth J, Kahl S (2009a) Specification Technique for the Description of Self-optimizing Mechatronic Systems. Research in Engineering Design 20 (4):201–223

Gausemeier J, Low CY, Steffen D, Deyterl S (2008) Specifying the Principle Solution in Mechatronic Development Enterprises. In: 2nd Annual IEEE Systems Conference, Montreal, Canada. p 1

Gausemeier J, Schafer W, Greenyer J, Kahl S, Pook S, Rieke J (2009b) Management of Cross Domain Model Consistency During the Development of Advanced Mechatronic Systems. In: Stanford (ed) International Conference on Engineering Design (ICED'09). pp 1-12

Group OM (2011) OMG Unified Modeling Language (UML) Specification V2.4.1. http://www.omg.org/spec/UML/2.4.1/.

Hansen CT (1997) Towards a tool for computer supported structuring of products. In: International Conference on Engineering Design (ICED'97), Tampere. pp 71-76

Hansen CT, Andreasen MM (2002) Two Approaches to Synthesis Based on the Domain Theory. In: Chakrabarti A (ed) Engineering Design Synthesis. Springer-Verlag, London,

Haskins C, Forsberg K (2011) Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities; INCOSE-TP-2003-002-03.2. 1. INCOSE-TP-2003-002-03.2.1 edn. Incose, San Diego

Hauser JR, Clausing D (1988) The House of Quality. Harvard Business Review:11

Hehenberger P, Poltschak F, Zeman K, Amrhein W (2010) Hierarchical Design Models in the Mechatronic Product Development Process of Synchronous Machines. Mechatronics 20 (8):864-875

Howard TJ, Culley S, Dekoninck E (2008) Describing the creative design process by the integration of engineering design and cognitive psychology literature. Design Studies 29 (2):160-180

Hubka V (1967) Der grundlegende Algorithmus für die Lösung von Konstruktionsaufgaben. In: XII Int. Wiss. Kolloquium der TH Ilmenau 'Konstruktion'.

Hubka V, Eder WE (1988) Theory of Technical Systems: A Total Concept Theory for Engineering Design. Springer Verlag,

IABG (1997) Vorgehensmodell Zur Planung und Durchführung von IT-Vorhaben; Entwicklungsstandard für IT-Systeme des Bundes. www.v-modell.iabg.de.

Isermann R (2005) Mechatronic Systems Fundamentals. Springer-Verlag, London

ISO/IEC 81346 (2012) ISO/IEC 81346 Standard. http://81346.com/english/.

Jansen S (2007) Eine Methodik zur modellbasierten Partitionierung mechatronischer Systeme. PhD thesis, Ruhr-Universität Bochum, Achen

Jensen T (1999) Functional Modelling in a Design Support System. PhD thesis, Technical University of Denmark, Lyngby

Jones DL (2004) PCB Design Tutorial, RevA. www.alternatezone.com.

Jørgensen K (1992) Videnskabelige arbejdsparadigmer. Institut for Produktion, Ålborg Universitet

Klir GJ, Valach M (1967) Cybernetic modelling. Iliffe Books,

Kreimeyer M, Braun S, Gurtler M, Lindemann U (2008) Relating Two Domains via a Third: An Approach to Overcome Ambiguous Attributions Using Multiple Domain Matrices. In: ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2008, New York, vol 43253. ASME, pp 297-306

Modelica Association (2010) Modelica Language Specification Version 3.2. http://www.modelica.org/documents/ModelicaSpec32.pdf.

Mortensen NH (2000) Design Modeling in a Designer's Workbench. PhD thesis, Technical University of Denmark

Nagel RL, Stone RB, Hutcheson RS, McAdams DA, Donndelinger JA (2008a) Function Design Framework (FDF): Integrated Process and Function Modeling for Complex Systems. In: ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2008, vol 43284. ASME, pp 273-286

Nagel RL, Vucovich JP, Stone RB, McAdams DA (2008b) A Signal Grammar to Guide Functional Modeling of Electromechanical Products. Journal of Mechanical Design 130 (5):051101-051110

Object Management Group (2010) OMG Systems Modeling Language Specification V1.2. http://www.omg.org/spec/SysML/1.2/PDF/.

Olesen J (1992) Concurrent Development in Manufacturing: Based on Dispositional Mechanisms. PhD thesis, Institute for Engineering Design, Technical University of Denmark, Lyngby

Pahl G, Beitz W, Feldhusen J, Grote K (2007) Engineering Design - A Systematic Approach. Second edn. Springer-Verlag, London

Pressman RS (2005) Software Engineering : A Practitioner's Approach. International ed., 6th ed. edn. McGraw-Hill Higher Education, Boston, Mass. ; London

Qamar A, During C, Wikander J (2009) Designing Mechatronic Systems, a Model-based Perspective, an Attempt to Achieve SysML-Matlab/Simulink Model Integration. 2009 IEEE/ASME INTERNATIONAL CONFERENCE ON ADVANCED INTELLIGENT MECHATRONICS, VOLS 1-3:1299-1304

Robertson S, Robertson J (2006) Mastering the requirements process. Addison-Wesley Professional,

Sage AP, Rouse W (2009) Handbook of Systems Engineering and Management. Second edn. John Wiley & Sons, Oxford

Salminen V, Verho A (1992) Systematic and innovative design of a mechatronic product. Mechatronics 2 (3):257-275

Salminen V, Verho AJ (1989) Multi-disciplinary Design Problems in Mechatronics and Some Suggestions to Its Methodical Solution in Conceptual Design Phase. In: International Conference on Engineering Design (ICED'89), Harrogate. pp 533-554

Shah AA, Kerzhner AA, Schaefer D, Paredis CJJ (2010) Multi-View Modeling to Support Embedded Systems Engineering in SysML. Graph Transformations and Model Driven Engineering- Lecture Notes in Computer Science 5765/2010:580-601. doi:10.1007/978-3-642-17322

Suh NP (2001) Axiomatic Design: Advances and Applications. Oxford University Press, New York

Tomiyama T, Amelio VD, Urbanic J, ElMaraghy W (2007) Complexity of Multi-Disciplinary Design. Annals of the CIRP 56 (1):185-188

Torry-Smith JM, Mortensen NH (2011) A mechatronic case study highlighting the need for re-thinking the design approach. In: International Conference on Engineering Design (ICED'11), Copenhagen. pp 276-287

Torry-Smith JM, Mortensen NH, Achiche S (submitted 2012a) Classification of product-related dependencies in development of mechatronic products. Research in Engineering Design:(under review)

Torry-Smith JM, Mortensen NH, Achiche S, Ploug O (submitted 2012b) The Mechatronic Integration Concept. International Journal of Industrial Engineering: Theory, Applications and Practice:(under review)

Torry-Smith JM, Qamar A, Achiche S, Wikander J, Mortensen NH, During C (2011) Mechatronic Design - Still a Considerable Challenge. In: ASME 2011 Design Engineering Technical Conferences & Computers & Information in Engineering Conference, Washington, USA, vol 54860. pp 33-44

Torry-Smith JM, Qamar A, Achiche S, Wikander J, Mortensen NH, During C (2012) Challenges in Designing Mechatronic Systems. Journal of Mechanical Design 135:011005-011001

Ulrich KT, Eppinger SD (2000) Product design and development. 2nd ed. edn. Irwin/McGraw-Hill, Boston

van Beek TJ, Tomiyama T (2009) Integrating Conventional System Views with Function-Behaviour-State Modelling. In: 19th CIRP Conference - Competitive Design. pp 65-73

Verho, Salminen, Kaynak, Ozkan, Bekiroglu, Tunay (1995) Customer needs, function structure and strategic development of mechatronic product. In: International Conference on Recent Advances in Mechatronics. pp 105-110

Welp EG, Jansen S (2004) Domain Allocation in Mechatronic Products. In: International Design Conference (Design2004), Dubrovnik. pp 1349-1354

Williams J (1991) Analog circuit design : art, science and personalities. Butterworth-Heinemann, Boston, Mass.

Woestenenk K, Tragter H, Bonnema GM, Cabrera AAA, Tomiyama T (2010) Multi Domain Design: Integration and Reuse. In: ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2010) Montreal, Canada, vol 44137. ASME, pp 519-528

Wolkl S, Shea K (2009) A Computational Product Model for Conceptual Design Using SysML. In: ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2009, San Diego, California, USA, vol 48999. ASME, pp 635-645

Wu Z, Campbell MI, Fernandez BR (2008) Bond Graph Based Automated Modeling for Computer-Aided Design of Dynamic Systems. Journal of Mechanical Design 130 (4):041102-041111

Zha XF, Fenves SJ, Sriram RD (2005) A feature-based approach to embedded system hardware and software co-design. Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference 2005, Vol 3, Pts A and B:609-620

10 Appendices

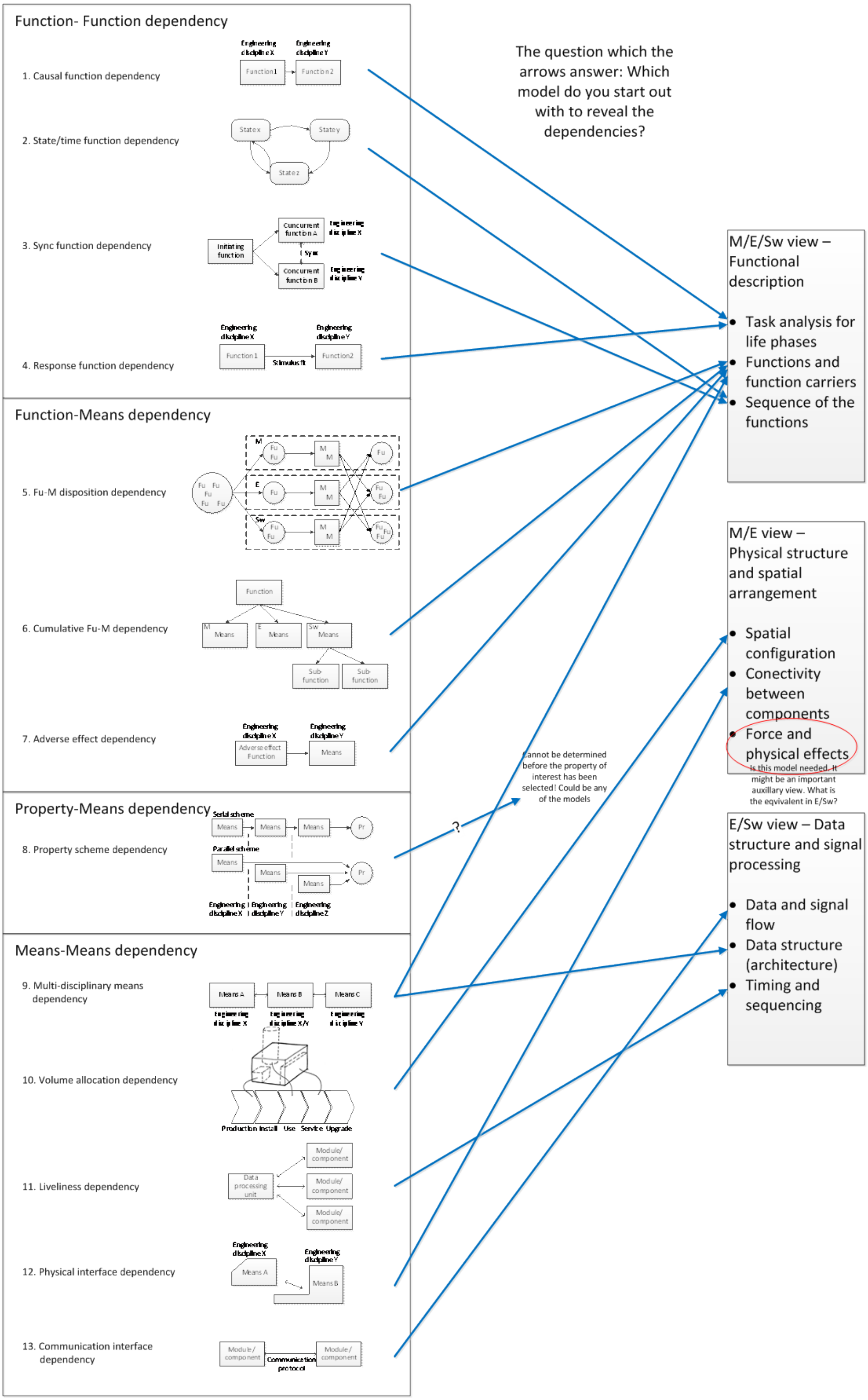
Appendix A

The following is a suggestion for at step-wise procedure for establishing the Mechatronic Integration Concept, thereby revealing dependencies. The types of dependencies to look for are marked as 'dep1' to 'dep13' and refer to the dependency number in Paper D.

1. Establish the presumed life phases for the product concept.
2. Create a functional overview by describing the task flow for each life phase.
3. Synthesize the technical process based on the previous two steps.
    a. Mark those domains involved in realizing each of the processes, and find out how that particular interaction between the domains will be for each process. (dep1 and dep4). Also ask how the transition between the *function* processes will be.
    b. Find transitions between processes belonging to one domain which create stimuli on other process *functions* and ask if the size and type of the stimuli can or needs to be specified (dep4).
4. Create a Function/Means tree structure
    a. Mark the lowest level *functions* or *means*. Find out what the next level *functions* and *means* will be (dep5). Identify if it is solved mono-disciplinary or if the 'next level' shift domain.
    b. Mark the function-allocation (thus what *means* are assigned to which domains), and find *means* and *functions* that are supported by more than one domain (dep6).
5. Create a FSM (finite state machine) overview of the product, starting with the use phase. Create one for each domain and see if they can be aligned (synchronized) (dep3). Create a common FSM for all domains and mark which engineering disciplines which most likely will be assigned the given *function* (causing the transition between the states). Discuss where there are boundaries between the domains as a consequence of the sequence of the *functions* (dep2).
6. List the most important *properties* and locate by which *means* they are realized. Identify those *properties* which are realized by *means* allocated to more than one domain. Identification of *means* can be done on basis of the Function/Means tree, a sketch of the structure with components, or via signal diagrams. Find out and discuss how the domains contribute to each of the *properties* (dep8).
7. Begin on sub-module level and describe what the module is expected to do; what it does, when does it do what it does, and what *properties* are allocated to the module. With the expected *means* in mind search for plausible 'adverse effects' and ask what will happen if unideal situations appear (dep7).
8. Identify the expected *means*/technology to be used in the product e.g. based on the Function/Means tree. Identity which of the *means*, which have to be handled by more than one engineering discipline (e.g. a sensor or a dc motor). Determine when the specification of the components can be agreed. Determine the uncertainty of the final design/shape/weight in order to make buffer zones around the components (dep9).
9. Make a rough sketch of the spatial location of modules and important components. Identify E, M and E/M components and check if available space has been allocated to them. Go through the life phases of the product and check for dynamical movement of parts and components (replacing a component, movement of a component, needed space for installing modules etc.)(dep10).
10. Determine how critical it will be if the data processing system freezes or if an input to the system is overlooked. Can it be solved with WDT ('Watch Dog Timer') and interrupts? If not, determine the flow and timing of events, and maybe programming paradigm to ensure the liveliness of the system (dep11). Use schematics and FSM models as the start of the discussion.

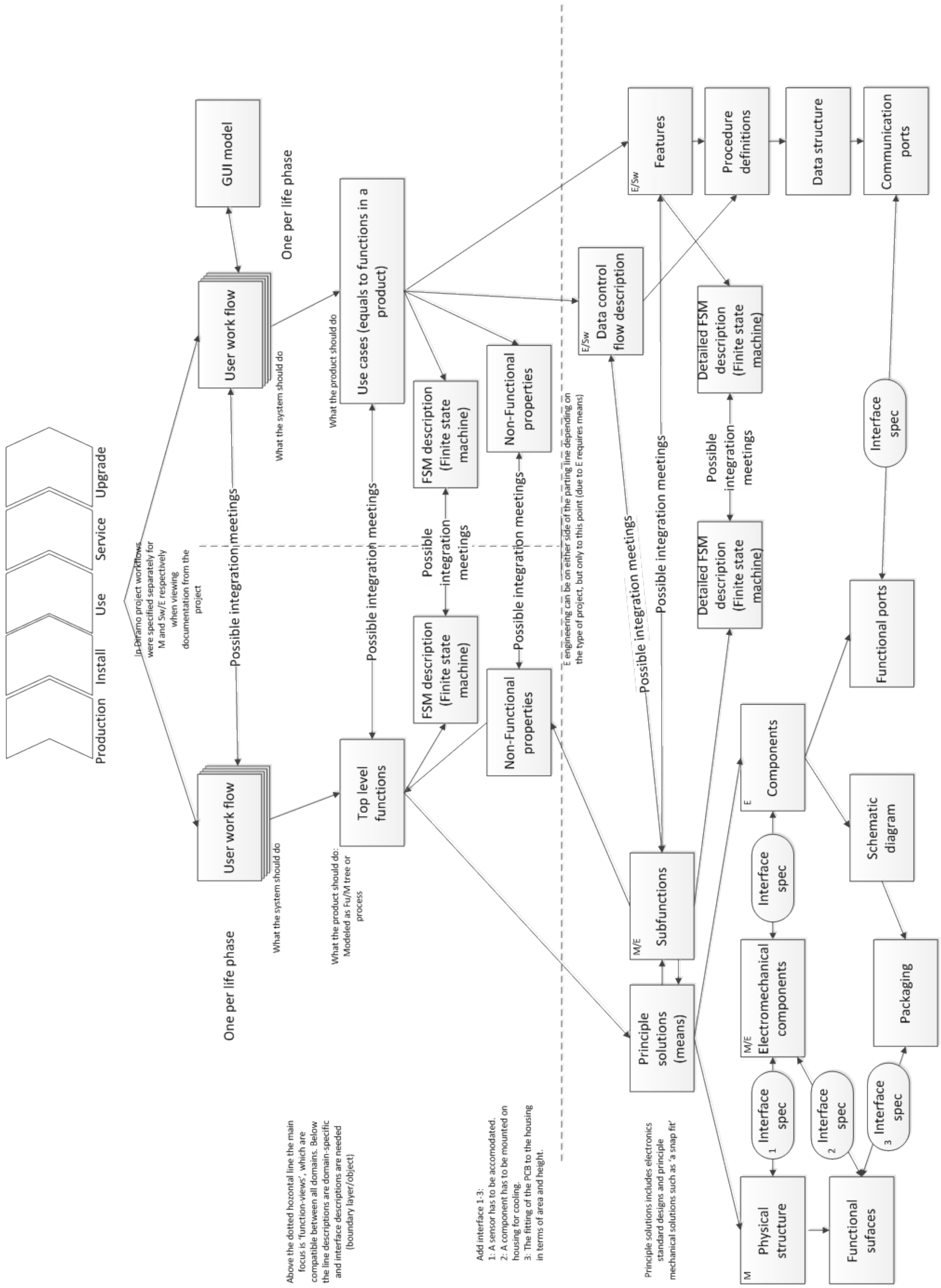
11. Determine the interfaces between modules and/or components and agree when they can be specified. What clarifications are needed before the specification can be made (dep12). Use rough 3D sketch, SysML model etc.
12. Determine the protocols and data interfaces between modules and/or components and agree when they can be specified. What clarifications are needed before the specification can be made (dep13).

Appendix B



Function- Function dependency
1. Causal function dependency
2. State/time function dependency
3. Sync function dependency
4. Response function dependency
Function-Means dependency
5. Fu-M disposition dependency
6. Cumulative Fu-M dependency
7. Adverse effect dependency
Property-Means dependency
8. Property scheme dependency
Means-Means dependency
9. Multi-disciplinary means dependency
10. Volume allocation dependency
11. Liveliness dependency
12. Physical interface dependency
13. Communication interface dependency
The question which the arrows answer: Which model do you start out with to reveal the dependencies?
M/E/Sw view – Functional description
Task analysis for life phases
Functions and function carriers
Sequence of the functions
M/E view – Physical structure and spatial arrangement
Spatial configuration
Conectivity between components
Force and physical effects
Is this model needed. It might be an important auxillary view. What is the eqvivalent in E/Sw?
Cannot be determined before the property of interest has been selected! Could be any of the models
E/Sw view – Data structure and signal processing
Data and signal flow
Data structure (architecture)
Timing and sequencing

APPENDIX C



Production
Install
Use
Service
Upgrade
In Diramo project workflows were specified separately for M and Sw/E respectively when viewing documentation from the project
User work flow
User work flow
GUI model
One per life phase
One per life phase
What the system should do
What the system should do
Possible integration meetings
Top level functions
What the product should do: Modeled as Fu/M tree or process
Use cases (equals to functions in a product)
What the product should do
Possible integration meetings
FSM description (Finite state machine)
FSM description (Finite state machine)
Possible integration meetings
Non-Functional properties
Non-Functional properties
Possible integration meetings
E engineering can be on either side of the parting line depending on the type of project, but only to this point (due to E requires means)
E/Sw
Data control flow description
E/Sw
Features
Procedure definitions
Data structure
Communication ports
Possible integration meetings
Possible integration meetings
Detailed FSM description (Finite state machine)
Detailed FSM description (Finite state machine)
Possible integration meetings
Interface spec
Functional ports
M/E
Subfunctions
Principle solutions (means)
E
Components
Interface spec
Schematic diagram
M/E
Electromechanical components
Packaging
Interface spec 1
Interface spec 2
Interface spec 3
M
Physical structure
Functional sufaces
Above the dotted hozontal line the main focus is 'function-views', which are compatible between all domains. Below the line descriptions are domain-specific and interface descriptions are needed (boundary layer/object)
Add interface 1-3:
1: A sensor has to be accomodated.
2: A component has to be mounted on housing for cooling.
3: The fitting of the PCB to the housing in terms of area and height.
Principle solutions includes electronics standard designs and principle mechanical solutions such as 'a snap fit'

Appendix D

Mechanical development with inspiration from Pahl & Beitz among others

- Search for product idea
- Clarify objectives
- Formulate product idea
- Formulate plausible technology building block (several options may co-exist in this early phase)
- Define match between business concept, product idea and activity (related to product)
- Establish life phases for product
- Establish workflow diagrams (process diagram)
- Define user interaction
- Base functional analysis on life phase analysis and work flow diagrams
- Elaborate a requirement list
- Abstract to identify essential problems
- Establish function structures (functions and subfunctions)
- Search for working principles that fullfill the subfunctions
- Combine working principles into working structures
- Select suitable combinations
- Firm up into principle solution variants (concepts)
- Evaluate against technical and economic criteria
- Identify embodiment-determining requirements
- Produce scale drawings of spatial constraints
- Identify embodiment-determining main function carriers
- Develop preliminary layots and form designs for main function carriers
- Select suitable preliminary layouts

Electronics developmemnt primarily established based on interviews

Initial investigation of idea

- Description of the user interaction
- Function analysis (What the product should do)
- Analysis of the desired behavior of the product
- Establishing Basic Specification (50% finished)
- Block diagram or similar, which shows the realization of the functions and the sequence of the functions
- Technology platforms for functions
- Feasibility study

Exploration/ Pre-study

- Design workflows (detailed user interaction)
- Identifying critical functions, properties and parameters
- Identifying sensors and actuators
- Considerations of what to reuse of modules, technologies, fu-principles and components
- (If software is apart of the product) Define what functions are solved in software and what functions are solved in electronics.
- Define what functions are solved with analogue components and what will be solved in digital domains (e.g. A digital potentiometer or a physical analogue)
- Decomposition of functional descriptions to functional units, which can be handled by standard designs and/ or specific component
- Finding components which can realize the desired functions (partition in analogue and digital components)
- Find suitable CPU (microcontroller/Microprocessor/ FPGA)
  - Evaluate needed Mips
  - Determine need for I/O pins
  - Evaluating need for I/O pins in relation to possibility of multiplexing and size of footprint compared available PCB area.
- Determining the programming paradigm if it affects the processor, which can be used. Else it can be determined later (e.g. Synchronous, or sequential with interrupts)

Software development primarily adapted from the Systems Engineering Handbook by INCOSE

Stakeholder requirement definition process

- Identify users and stakeholders
- Define needs
- Capture source requirements
- Initialize the requirements database
- Establish the concept of operations
- Generate the systems requirement document

Requirements analysis process

- Requirement analysis concepts
- Characteristics of good requirements
- Define systems capabilities and performance objectives
- Define, derive, and refine functional/performance requirements
- Define other non-functional requirements
- Develop specification trees and specifications
- Allocate requirements and establish traceability
- Generate the systems specification

Architectural design process

- Architectural design concepts
- Define selection criteria
- Define/refine systems element alternatives
- Synthesize multiple system architectures

Continues on next page

Continued from previous page

Develop preliminary layots and form designs for the remaining main function carriers

Search for solutions to auxiliary functions

Develop detailed layouts and form designs for the main function carriers enduring compatibility with the auxiliary function carriers

Develop detailed layots and form designs for the auxiliary function carriers and complete the overall layouts

Preliminary form design, material selection and calculation

Evaluate against technical and economic criteria

Eliminate weak spots

Check for errors, disturbing influences and minimum costs

Prepare the preliminary part list and production and assembly documents

Elaborate detail drawings and part list

Complete production, assembly, transport and operating instructions

Check all documents

Design full circuit diagram made up of the ideal circuit blocks (having ideal behaviour).

Consider non-ideal aspects of the schematic and adjust accordingly (both as a consequence of putting functional blocks together and as a consequence of noise (adding filters), changing power supply voltage (adding stabilisers), physical effects affecting sensors (e.g. vibrations), etc.

Treat each function (functional block) as ideal and analyse/modify going back and fourth between time domain and frequency domain. Thereby simplified schematics are generated for each function.

Simulation in e.g. pSpice (might only be a section of the scematics, which is critical and need simulation)

Sensitivity analysis (Monte Carlo for analogue circuits)

Full scematic diagram for hardware

Test on bread board/evaluation board if needed

Determine PCB technology (thick layer film, normal hard PCB, flex print)
Might be a strategic/conceptual decision, therefore decided much earlier in the process

PCB layout

- Tracing (DA:baneudlægning)
- Traces on PCB
- Component placement on PCB

(For making artwork see pdf-guide for this and extract the best)

Determine physical dimensions of PCB

Test on bread RPT PCB

Finalization of circuit and PCB

Analyze and select preferred system architecture/ element solution

Model, simulate, and prototype system architectures

Define, refine and integrate system physical configuration

Implement requirements and design feedback loops

Implementation process

Implementation concepts

Integration process

Integration concepts

Systems build

System integration with external systems

Verification process

Verification concepts

- Development test
- Qualification test
- Acceptance test
- Operational test

11 APPENDED PAPERS

Paper A:
"A Mechatronic Case Study Highlighting the Need for Re-thinking the Design Approach"

Paper B:
"Mechatronics Design – Still a Considerable challenge"

Paper C:
"Challenges in Designing Mechatronic Systems"

Paper D:
"Classification of Product-related Dependencies in Development of Mechatronic Products"

Paper E:
"The Mechatronic Integration Concept"

11.1 Paper A

"A Mechatronic Case Study Highlighting the Need for Re-thinking the Design Approach"

Published in the proceedings of the International Conference on Engineering Design (ICED'11)

A MECHATRONIC CASE STUDY HIGHLIGHTING THE NEED FOR RE-THINKING THE DESIGN APPROACH

Jonas Mørkeberg Torry-Smith, Niels Henrik Mortensen
Technical University of Denmark, Denmark

ABSTRACT
Developing mechatronic products is a great challenge for many companies due to the multi-disciplinary nature of the development process. In this article the main objective is an investigation of seven aspects related to the synthesis process of developing mechatronic products. The role and effects of these aspects are illustrated by a case study. A literature study is performed regarding how well the seven aspects have been covered in the literature. It reveals that some suggestions for support can be found in terms of semi-formal modelling suggestions and proposal for procedures, but that the context of the proposed support often originates from a control engineering dominated research area. This circumstance leaves a vast amount of other types of mechatronic products with only sparse development support with the potential of being made operational.

Keywords: Mechatronics, development process, synthesis, literature review, case study, conceptual design

1 INTRODUCTION
Companies involved in developing mechatronic products face the challenge of 'orchestring' the different engineering disciplines involved. Long has it been acknowledged that there is a need for integration and long has it been acknowledged that central areas of mechatronic development lack theory and methods [1], [2], [3]. Without a well covering theory and applicable methods companies cannot exploit the full potential of mechatronics [4].
This article is investigating central aspects, which the engineers must face and must be able to handle in the synthesis of mechatronic products. The investigation is build upon a case study to show the context in which the aspects appear. The investigation of the aspects is then continued in a literature study. The scope of the study is to investigate to which extend the seven aspects are acknowledged in literature and to clarify if methods or tools have been suggested for better handling the aspects. The seven aspects can each be categorized within one of the following three areas: Process related aspects, product related aspects and aspects related to user perceived value. The result of the research presented in this article will be used for directing the search for support for the remaining part of the PhD project.
The investigation is limited to incorporate aspects related to the field of mechanical, electronics and software design. For this article the term 'mechatronics' is used when these engineering fields are combined in the product development. The control engineering field is regarded as a competence in this context similar to many other competences needed for the vast amount of different types of mechatronic products.
The words 'function', 'property' and 'structure' are used in this article. The definitions are adopted from the work done by Mogens Myrup Andreasen [5]. In short, functions and properties describe 'what the product does', whereas the structure describes 'what the product is'. A function has an effect such as the function 'provide power', whereas a property does not have an effect such as 'robustness'.
The article is structured as follows. In section 2 the research steps are explained. In section 3 the selected mechatronic aspects are described. In section 4 the aspects are illustrated in the case study. Section 5 contains the literature review and section 6 concludes on the article.

2 METHOD
Both authors have each nine years or more of hands-on experience with industrial mechatronic projects. This experience has been used to select the seven aspects. The case study is used to illustrate the importance of being able to handle the aspects in the design process. The case study has been built by use of several means: i)Personally recorded experience, since one of the co-authors participated in

the project. ii)Analysis of documents and files from the project and iii)Semi-structured interviews with the project managers. For the literature study, a limited number of eleven references is carefully chosen to reflect the state-of-the-art within providing theory and methods. An overview of the coverage found in the literature is established and is illustrated in Table 1. The case study, the literature review and the overview in Table 1 are used to form the final conclusion.

3 ASPECTS TO BE INVESTIGATED

The aspects chosen as the focus for the case and the literature study are multi-disciplinary of nature. It is not the intention of the authors to create an exhaustive list of relevant aspects when designing mechatronic products. Instead the aspects have carefully been selected within three main areas, which we stipulate have an utmost significant impact for the ability to synthesize successful mechatronic products. The three areas are i) the process of developing a mechatronic product, ii) the product itself and iii) the value created in the meeting between the user and the product. In the following the selected aspects are described and categorized according to the three areas. The identification (A1...A7) is used for tracking each of the aspects in the case study.

Process

- Synchronization between the mechatronic process model and the process models of the separate domains (A1). A mechatronic process model should not conflict with the normally found flow of activities in the domains. Instead it should support the synchronisation of the concurrently performed development within the domains. Without an understanding of the synchronisation aspect in mechatronic development, deliverables between the domains cannot be planned, which will cause the level of integration to decrease.
- Normal iterations occur when we go through the design cycle and improve the solution for each iteration. The iteration aspect to be described in relation to this article is different in nature. When working in e.g. the mechanical domain we must assume the electronics are fixed and does not change in terms of interfaces and functions important to the mechanical domain. Thereby work in one domain must be perceived as evolving in iterations seen from the other domains in between ‘integration meetings’ (A2). Because the design is constantly evolving in every domain it becomes important to clarify the areas likely to change. The relations can be many and without an overview or a strategy the risk for failure in the project will increase.
- The allocation of functions to the domains can be regarded as a balance between the domains as described by Buur [6]. Relevant balances must be synthesized as alternative concepts to investigate the solution space to reach the ′best fit’ solution (A3). The function allocation determines the size of the task assigned to each of the teams representing the domains. Furthermore it will have a direct effect on the physical interfaces needed to connect the technology from each domain. Therefore it has a significant impact on the design process. The allocation can be made, based on various strategies ranging from product related considerations to organisational related considerations.

Product

- Distribution of functions and properties between domains (A4). Functions and properties have to be considered carefully during a development process [5]. Mechatronic projects pose an increased challenge due to the multi-disciplinary nature. The development task has to be decomposed into ‘chunks’, which can be handled by the different teams thereby risking a separation of closely connected functions or properties. An example could be a property such as “measurement accuracy”. Such a property can have contributing factors/elements in each of the three domains. To create and optimise the property several domains have to be considered at the same time, which is a major challenge due to the vast amount of properties and functions found in a product.
- Sharing schemes of the functions and properties in the product to be developed (A5). The design engineers should be well aware of which and how elements contribute to a certain function or property in the product. As an example, the effort of optimizing the property ‘accuracy’ to the desired extent might be easier achieved in the E domain compared to the M domain in a particular case. The understanding of the sharing schemes gives insight into how the functions or properties should be optimised. For example, the sharing principle in a concrete situation could be ‘the weakest link of the chain’, or each of the contributions could add to the property in a

multiplicative or additive scheme. Each of the sharing schemes would call for a different optimisation strategy.

- Interface handling (A6). Specifying physical interfaces in development is widely used in the industry to create an architecture by which the various development teams will have fix points for their physical realisation of the product. Decisions in the software domain can affect physical interfaces but interfaces in the physical sense belong to the mechanical and the electronics domain.

User perceived value

- The manipulation of the design to obtain the desired perceived value (A7). Careful attention is needed to simulate, model or by other means try to predict the user-perceived value of the product. The user's interaction with the product during the life phases will generate a perception of value. Typically the value perception is directly influenced by choices we make in the development process, choices that can be of high technical character.

If the members of the development team with multiple engineering discipline backgrounds efficiently can handle the described aspects, a more transparent and rigorous development process can be obtained.

4 CASE STUDY – THE LINDEWERDELIN WATCH SYSTEM

The project chosen for the case study is well suited because the aspects selected for the investigation are well represented in the project and because the case study contains design considerations from each of the M, E and Sw domains.

The case describes the development of a temperature sensing unit, which is a part of a watch system, see Figure 1. The product is targeted the high-end market for outdoor sport watches, and is produced by the company LindeWerdelin. The product idea of the watch system is based on a mechanical watch, on which an instrument with some advanced functions can be attached. An external temperature measuring unit can be positioned away from the watch to measure the actual surrounding temperature. An external heart rate unit can be positioned around the person's chest to measure the heart rate. The external units wirelessly transmit the measured data to the instrument for displaying the information. The system also contains a battery charger for the instrument. It is a part of the product idea to be able to attach the instrument to the thermometer unit, thereby restricting the shape of the temperature unit.

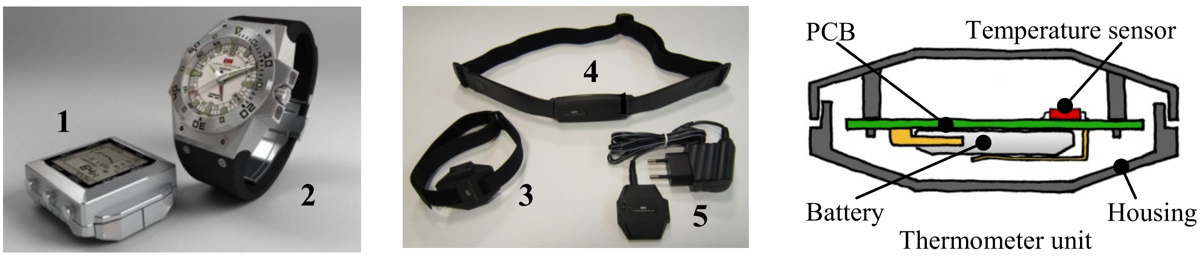


Figure 1. 1: Instrument, 2: Watch, 3: Temperature unit, 4: Heart rate unit, 5: Instrument charger

The project was initiated by the co-founders of the newly, at that point in time and for that purpose, established company. The product idea was developed by the co-founders. The mechanical development was outsourced to one consultancy company and the electronics and software development to another consultancy company. The mechanical watch was to be developed and produced by a Swiss watch company. During the development two mechanical engineers, one electronic engineer and up to four software engineers were working on the project not including the resources for developing the clock mechanism. Figure 2 is a reconstruction of the development phases for the thermometer unit with a short description of the main activities. The description of the case study is sectioned according to these phases.

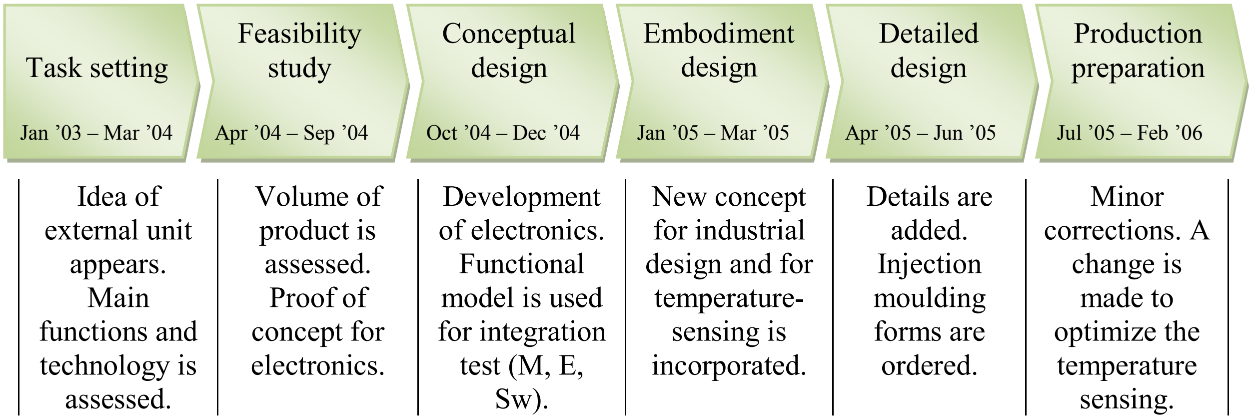


Figure 2. Development phases of the thermometer unit

4.1 Task setting

In the early stage of the conceptual phase the main focus in the project is to get the concept right for the watch and the instrument. Developing the external units is considered feasible and is therefore not the centre of attention. However, the primary functions for the thermometer unit are considered as being: “to measure temperature” and “to wirelessly transmit the temperature data”. To be able to make a feasibility study of the watch and instrument, communication with the external units has to be taken into consideration. This requires the task setting for the thermometer unit to be defined further. Based on the desired functions of the thermometer unit the following main components are suggested: Housing, battery, print board, antenna.

Within this initial suggestion for means to achieve the functions, an allocation of the functions is being made, which can be seen by the stated means in Figure 3. The figure shows the initial Function/Means Tree. The allocation principle is based on ‘the most obvious choice’. Trying to force a different allocation will make the solution to become obscure. The underlying functions necessary to realise a certain means have to be allocated to one or more domains. A means, which would be described as belonging to one domain, can have supporting functions from the other domains. One example is the PCB, which needs connection support to the housing. The function allocation is occurring throughout the design phases of the project (A3).

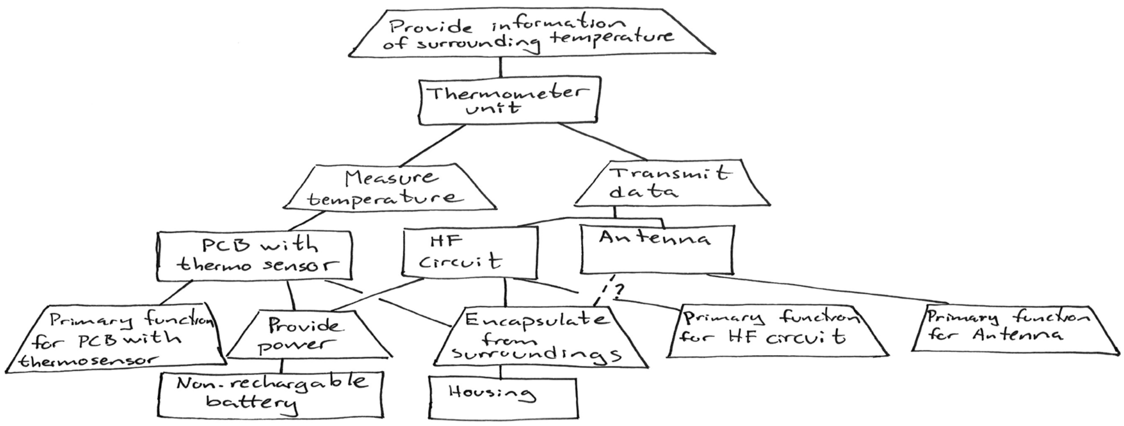


Figure 3. Function/Means Tree

The intention with the watch system is to brand it as being ‘luxurious’ and ‘high-tech’. The product should therefore contain properties leading to this user perception. Two of the needed properties are: ‘Low tolerance on the temperature measurement’ and ‘low power consumption’. The property ‘low power consumption’ is derived from the conclusion that changing battery too often is not leading to the perception of a ‘luxurious’ and ‘high-tech’ product (A4) (A5).

4.2 Feasibility study

The E engineers begin the development of the electronics based on the conceptual idea of the thermometer unit. The two main issues they begin to consider are the power consumption of the electronics and the technology needed to establish the wireless communication. A very rough cad model (undetailed) is made to assess the volume needed and to align it with the industrial design wishes (A1) (A6). Volume requirements are based on the initial guess of needed components. Regarding the power consumption two principally different solutions are considered; namely to preserve energy by different means or to be able to recharge the unit in a charging station by which the problem of power consumption seems reduced. A system where the unit should not be recharged is evaluated to be more user-friendly. Furthermore it is estimated that several means within the E and Sw domain can be utilized to conserve energy, and ultimately a switch can be used to turn off the unit when it is not in use. A switch would have considerable impact on the M domain due to the waterproof requirement. From this it can be seen that there are multiple relations between functions, properties and structure across the three domains when trying to optimize the power consumption (A4) (A5) (A6). The suggested means for the electronics are illustrated in Figure 4.

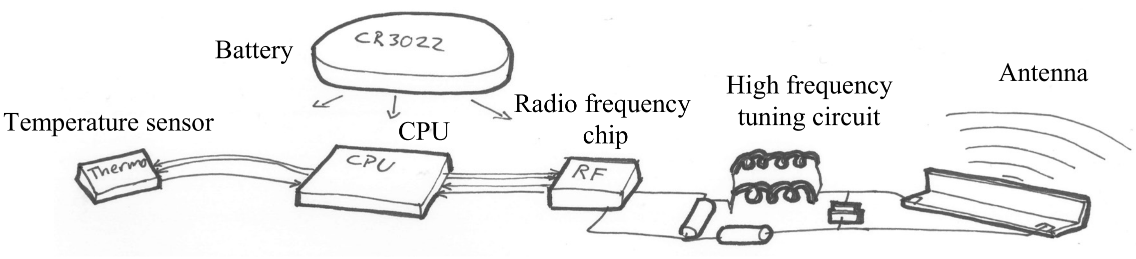


Figure 4. Suggested means for functions, which are allocated to the electronics domain

The feasibility study regarding the instrument and its ability to communicate with the external unit forces the electronics to be developed slightly ahead of the mechanical solution. It means that the electronic diagram and the PCB (Figure 5) are made for initial testing at the time where the structure of the housing is only roughly sketched (A1).

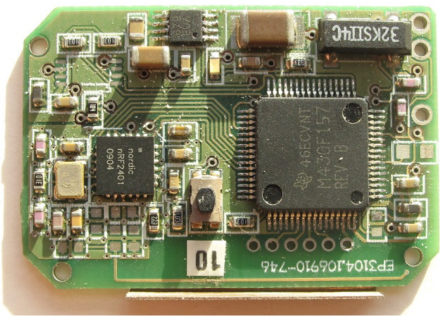


Figure 5. PCB for thermometer unit. Dimensions for PCB: 37 x 25 mm

4.3 Conceptual design

At this stage some of the mechanical development resources are redirected to the thermometer unit. It is assumed that the PCB will remain unchanged with respect to the size, the mounting holes, the shape of the antenna and the position of the antenna. A specific battery is suggested. These components have to be assumed to remain unchanged to admit the mechanical engineers to begin their work based on the industrial designer's suggestion for an outer shape. However, it is known to the design engineers that several components in the electronics design can change including another design of the antenna. The changes might include switching from an on-board antenna to an external antenna, change of the type of the battery and maybe a change due to a requirement to incorporate an on/off switch. This is an illustration that development within one domain has to assume the other two domains as fixed for a certain duration of time. Of course the developers are aware that some and maybe even predefined elements or aspects may change, but the other domains must still be assumed to be fixed until the next iteration of the product (A2).

In the conceptual design phase of the thermometer unit the life phases of the product are considered. Two of the many aspects considered, are the use phase and the service phase in order to optimize the user perceived value (A7). The use phase requires watertight seal of the electronics from the surroundings and the service phase requires easy change of the battery with low risk of harming the electronics by this operation. Several suggestions are made for positioning the battery and designing the battery terminals. Some of those solutions lie within only one domain whereas other solutions require a mix of solutions from several domains. The pool of solutions can be regarded as balances between the domains according to where the functions are allocated. This indicates that extreme balances can be used for generation alternative concepts and for investigating the solution space (A3). In Figure 6 some of the solutions regarding the battery terminals can be seen.

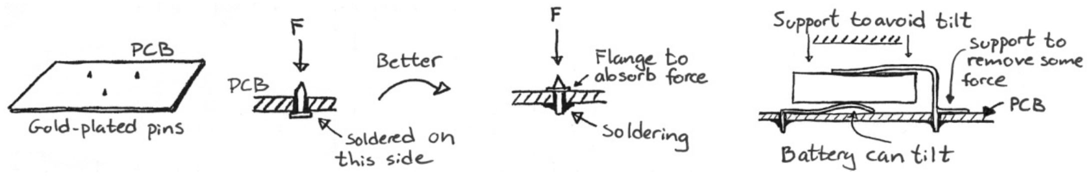


Figure 6. Illustrations of some of the sketched solutions for the battery terminals

The property 'transmitted signal quality' is considered throughout the phases of the project. Many relations between functions and means from all domains influence this property. To illustrate this, the shape of antenna, the chosen electronic components, the position of the battery and other metal objects in the design, the capacity available due to the selected battery and the software code are some of the contributors to this property (A5). The property of 'transmitted signal quality' is different from the property 'robust device', which was also handled in the project. The clear signal can be described as a sequence of instances, which all have to be optimized considering the 'the weakest link of the chain' principle. Robustness can be located many different places in the product. This property can be considered as parallel instances each separately contributing to robustness (A5).
A functional model is made in RPT material incorporating the suggested means (see Figure 7). The functional model of the thermometer is field tested at a ski resort together with the functional models of the instrument and the heart rate monitor.

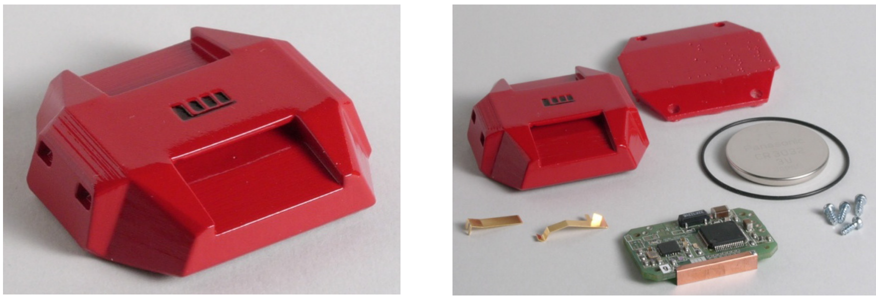


Figure 7. The functional model used for the first field test

4.4 Embodiment design

The integration test shows that the thermometer unit is sufficiently accurate in reading the temperature, but changes in temperature is not detected rapidly enough as anticipated with the temperature sensor located on the PCB. The test also shows that the wireless transmission has to be improved to reach the high standard expected by the users (A7).
At this stage of the development process the E development team is focused on improving the HF transmission by tweaking the discrete components. Furthermore the E development team has to solve issues related to the electromagnetic noise from the transmission, which degrades the performance of the electronics. Concurrently the software engineers are working on controlling of the HF digital chip, which is a more resource intensive task than first anticipated (A1) (A4) (A5).
Based on the appearance of the RPT model the designers suggest a changed shape of the thermometer unit making it appear lighter. The suggested design is shown in Figure 8.

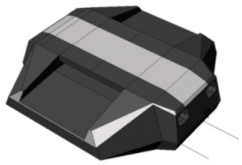


Figure 8. New proposed shape of the thermometer unit

The new shape even though it does not seem significant has consequences. The most important consequence is that the unit has to be re-modelled in the mechanical CAD system (A1). In the new design an aluminium decal on top of the thermometer unit is incorporated to conduct the surrounding temperature to the sensor quickly for the sensor to rapidly detect temperature changes. The idea is to place an aluminium rod between the decal and the sensor mounted on the PCB. In Figure 9 the sketch of the concept is shown as well as a simulation of the heat flux.

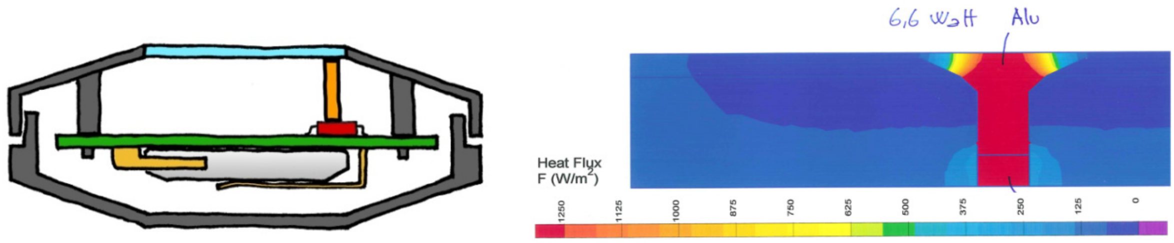


Figure 9. Concept with aluminium rod and the simulation of the heat flux

Power saving schemes are implemented in the software, the electronic components are tweaked by experiments made by external hired specialists to optimize the transmitted signal quality, and the cad model is made so a second field test and laboratory test can be made. Both tests show good results.

4.5 Detailed design

The mechanical design is improved to the stage were injection moulds can be ordered. One of the tasks is to decouple the forces from the battery when the unit is dropped or vibrated so the forces will travel into the housing and not into the PCB or terminals. A vibration test is performed which reveals that the temperature occasionally will not be updated for a short duration of time. After an investigation the cause turns out to be that if the battery is disconnected from the terminal in the range of just microseconds, the µ-processor will re-boot and the instrument and the unit will lose their transmission synchronization. Until reconnected the temperature will not be updated on the instrument. The terminals for the batteries act as springs and should have been able to make a secure connection. However, since the disconnection of just a microsecond can cause re-boot of the µ-processor, eigenfrequencies or similar vibration phenomena could be the cause. Instead of improving the mechanical system surrounding the terminal springs, a capacitor is added to the electric circuit, which will compensate for disconnections (A3). The solution is robust since it is insensitive to the cause for such small disconnections.

4.6 Production preparation

Having the thermometer in the almost finished design more mechanical, electronics and software testing is performed. Due to the wish for high-tech perception the unit should indulge, it is decided to increase the speed at which the unit can detect temperature changes. After some tests and conceptual work, it is evaluated that one particular solution will improve the temperature sensing and only cause minor changes in the mechanical and electronic design. Since the injection moulds have already been manufactured it is important that the change only will require minor changes of the design. Three of the suggested solutions can be seen in Figure 10.

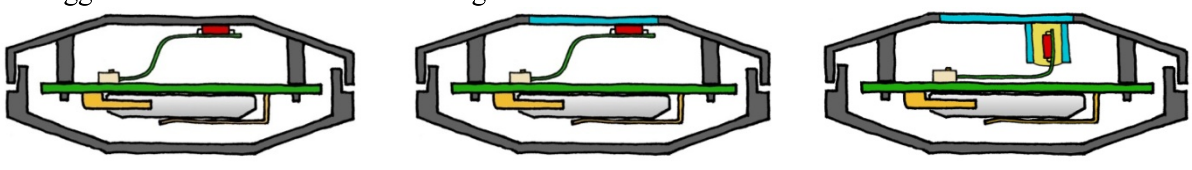


Figure 10. For illustration purpose the position of the terminals has been changed on the illustration

The solution is based on positioning the sensor on a flex print and locating the sensor as close as possible to the aluminium plate as possible considering other requirements. The new version of the PCB can be seen in Figure 11.

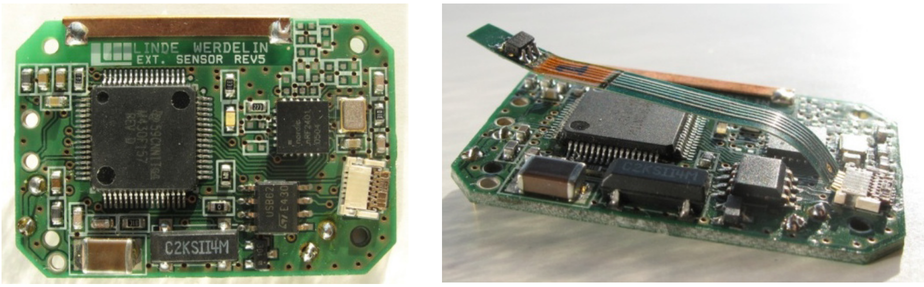


Figure 11. The PCB is shown without and with the flex print inserted in the connection terminal

The case study illustrates the seven aspects related to the design process, the product, and the user-perceived value. The case illustrates that it is essential for the designer to be able to handle the aspects across the engineering disciplines and not just see the issues locally from within a single domain. With the amount of relations to handle, it is hard to imagine that it can be done without applying some sort of systematically approach. Great many relations between functions, properties and structures as well as dependencies between activities in the design process can be observed in the case study. When systems become larger and the number of persons involved in the design process increases so does the potential relations and dependencies. This further underlines that it is important to be in control of the described aspects in the design process.

5 LITERATURE REVIEW

In the following relevant literature is investigated with the intent of revealing the support found in the literature regarding each of the selected mechatronic aspects (A1) to (A7). First an overview is presented in Table 1 showing the rating of how well the literature is covering the particular aspects. Then each of the references is described in a general form to show the context and the intention of the literature to have incorporated mechatronic aspects in the text. To go through every aspect for each of the references would be tedious for the reader, and is therefore omitted. Thereby the reader must rely on the judgement by the authors to have performed the rating systematically and unbiased. The selected literature is investigated in the context of mechatronic synthesis. Therefore, if the a literature is describing for example life phases but not in the context of the synthesis process and without addressing the particular impact on mechatronic development, it will be rated as “not describing” the particular aspect. Types of references include text books, scientific papers and PhD thesis. The legends used for ranking are described in the following.
‘0’: The aspect is not described.
‘1’: The aspect is acknowledged and a characterisation may have been performed.
‘2’: The aspect is treated thoroughly and a method for handling the aspect is suggested.

VDI2206 guideline [7]**.** VDI2206 is a broad introduction to the subject explaining the fundamental challenges of mechatronic engineering. The proposed methodology have great similarities with the methodology for mechanical development suggested by Pahl and Beitz in their book “Engineering Design” in the strong focus on machine design. The V model is used as the process model for illustrating the phases of development of mechatronic products. Besides a general introduction, the guideline describes the phases ranging from the goal setting of the project trough system design over domain specific and validation and verification of the intended product and also including organisational aspects of corporation between team members across disciplines. All these aspects are described in a page wise very compact format, thereby not capable of incorporating descriptions of guidance and methods for performing essential tasks of the synthesis process.
Systems Engineering [8]**,** [9]**.** In Systems Engineering the main idea for handling multi-domain development is to break down the task into subtasks thereby breaking down the product into modules which can be handled. Having performed decomposition, the important relations are modelled in e.g. IDEF and/or via specification management. Multi-disciplinary issues are solved by use of traditional management tools such as project planning, staffing, resources, risk handling, TQM etc. and not by

specific mechatronics related methods. It lacks description of the synthesis steps especially for the mechanical area. In this sense it shares more commonalities with development procedures for software such as those found in the book "Software Engineering: A Practitioner's Approach" by R. S. Pressman [10] than with the procedures for mechanical development such as Pahl and Beitz [11] or Ulrich and Eppinger [12].

Table 1. Overview of how well the aspects are covered by the literature

	VDI 2206	Systems Engineering	V-modell XT	J. Buur	J. Gausemeier	S. Jansen/E.G. Welp	V. Salminen/ A. Verho	R. Isermann	R. H. Bishop	R. H. Bracewell	Pahl and Beitz
Synchronization of M, E and Sw process (A1)	1	1	0	1	1	0	1	2	0	0	1
The domains seen as iterations (A2)	0	0	0	0	0	0	1	0	0	0	0
Function allocation and alternatives (A3)	1	0	0	1	1	2	1	1	1	0	0
Distribution of Fu and Pr (A4)	1	0	1	0	1	1	1	0	0	0	1
Sharing schemes for Fu and Pr (A5)	0	0	0	0	0	0	0	0	0	0	0
Handling of physical interfaces (A6)	1	1	1	1	1	1	1	0	0	0	0
User-perceived value in the life phases (A7)	0	0	1	0	0	0	1	0	0	0	0

The V-model XT [13]**.** The model is based on the 'V-model' suggested in 1997, which was solely aimed at software development. The V-model XT is intended for products containing electronics and software, also called embedded systems. The role of the mechanical domain in the development process is not considered even though the embedded systems in most cases will have to interact with the mechanical elements. A framework is suggested for how to configure the V-model XT to fit a particular project. For each configuration of the V-model XT, different entities of the process model will appear. The interesting part of the process model is, however, that the life phases play a central role in the process description, which makes it stand out compared to the other references in the literature study. Even though mechanical development is omitted, the V-model XT is included in the literature study, because the V-model concept is one of the most referenced models in mechatronic literature.

J. Buur [6]**,** [14]**.** The literature comprises a very comprehensive categorisation of differences and similarities between the domains based on the theoretical view of "The Theory of Technical Systems" by Hubka [15] and "The Domain Theory" by Andreasen [16], [5]. Methodologies from before 1990 are discussed and phenomena linked to the development of mechatronic products are described. The theoretical and categorisation approach provides a foundation for understanding the area. However, the limitations of the research lie in trying to stretch a theory originally belonging to the mechanically domain to cover electronics and the software domain. The consequence is that only aspects of the development, which have an equivalent in the mechanical domain, are treated in the research. To illustrate this, issues such as those linked to dealing with 'real time systems' cannot be described or made operational by the use of the theories.

J. Gausemeier [17]**,** [3]**.** These two references have been selected among several from Gausemeier. These articles address the early phases in the design process of developing mechatronic products. The focus is on how to specify the principle solution, on how to control the design process and on how to provide an organizational support for the design process. A semi-formal functional model is suggested that should enable designers to specify a mechatronic product in the conceptual phase of a project and thereby overcoming the often mentioned common language gap between domains. The descriptions and models suggested are tangible, but lack the in-depth description of the synthesis process in the domains.

S. Jansen/E.G. Welp [18]**,** [19]**.** Jansen and Welp aim at providing a procedure for development of mechatronic products. The process description is mainly focused on the function allocation aspect, for which he suggests procedural support. It is a suggestion consisting of a process including rules and guidelines for making variations of the function allocation. The suggestions for themes which can be used for creating variants makes it stand out from the other literature contributions within the mechatronic research area. As a part of the process of allocating functions, categorisation and classifications of elements should be done in a written form which then can be modelled in an UML-equivalent model language. The amount of written data needed to make the model operational can prove to be disadvantageous in a synthesis process. The reason is that written information lacks the visual representation needed especially by the mechanical engineers. This disadvantage has also been reported by Bonnema [20]. Furthermore it is the authors experience that large amount of written data tends quickly to be outdated in fast paced projects.

V. Salminen/A. Veho [21]**,** [1]**,** [22]**.** The challenges of developing mechatronic products are thoroughly described and the challenges are categorised according to the development phases they appear in. Aspects needed to be considered in the process of going from user-needs to a functional description while considering strategic issues are highlighted and key questions for support are stated. A vast amount of conclusions are drawn linked to what characterizes the nature of mechatronic development projects. Some guidelines based on best practice are declared and a “metamethodic” is suggested which is a framework for how and when to utilize available methods and tools such as VDI2221, QFD and UML-equivalents in the development process. The integration aspect in terms of the overlapping areas between the domains is only vaguely treated in how it should be handled in a project. The suggestion presented, is to bring designers from each of the domains together to obtain a mutual understanding of the goals and tasks to be performed in the project.

R. Isermann [23]**.** The book has a strong focus on control engineering and control principles. However, a detailed process description is stated by listing activities grouped according to the phases in a development project. Even though description of the process emphasises activities linked to control engineering, the description is unique in the sense of the vast amount of stated activities. The activities are only briefly described and the underlying mechatronic phenomena linked to the activities are thereby not described. A model is illustrated to support the description of the process. However, the model does not show integration activities. In contrary it seems to promote separate tracks for each domain.

R. H. Bishop [24]**.** This book is about mechatronic systems with a strong focus on the control aspect. The chapter of most relevance is called ‘Mechatronic Design Approach’. It presents a framework for understanding the elements of a mechatronic system such as actuators, sensors and the information system, various control strategies and a procedure for the design process. Even though a stepwise procedure is stated, the strong focus on control engineering has the effect of suppressing other design activities and needed framework understandings for performing a synthesis of mechatronic products.

R. H. Bracewell [25]**,** [26]**.** This reference is included in the literature study because the program ‘Schemebuilder’ is claimed by the developers to be ‘a highly integrated “design workbench”’ capable of assisting the design process in problem analysis and in the conceptual and the detailed phase of designing mechatronic products. Suggesting artificial intelligent computer software for product development should be an object for sound scepticism. However, for this literature study the focus is on the design methodology, which is used as the backbone in the Schemebuilder software. The design methodology is based on French’s model of conceptual design, and is as such heavily influenced by traditions of design thinking from the mechanical research area. The suggested procedure is a straight forward functional decomposition procedure, in which the myriad of complex relations between the domains are omitted in sense of phenomena description or tools for handling these challenges. Even though the Schemebuilder is presented as very comprehensive in supporting the mechatronic design process the listed aspects in Table 1 are not covered.

G. Pahl and W. Beitz [11]**.** In the book a short introduction to the phenomenon mechatronics is found followed by a description of three mechatronic products illustrating the benefits of having all three domains working together in a product. The description of mechatronics has been included in a chapter, which also comprises “Mechanical Connections” and “Adaptronics”. The topic “Mechatronics” has not been integrated in the chapters regarding product planning, task clarification, conceptual design, embodiment design, underlining that development of mechanical products is the main focus in the book and not mechatronic products. The authors of the text book acknowledge the brief treatment of the topic. Hence, references are made to the VDI2206 and to Rolf Isermann in terms of suggesting a support for a mechatronic design procedure.

Conclusion on the literature review

The following aspects have been rated '0'or '1' in Table 1: Domain iterations (A2), Distribution of functions and properties between domains (A4), Sharing schemes (A5), Handling of physical interfaces (A6), User-perceived value in life phases (A7). This shows that there is a gap between the need for handling the aspects in a mechatronic synthesis and the support found in the literature. The following aspects have been rated '2': Synchronization (A1) and Function allocation (A3). The rating of "2" has, however, only been achieved by one of the eleven literature sources. This also indicates room for improvement. The control engineering field is mechatronic in nature because it has elements from each of the domains. This is reflected in the amount of references originating from control engineering research communities, including those references achieving a top-rating in this article. The logical consequence is that methods and procedures are heavily influenced by activities closely linked to the control issue of the product development. There are many other types of mechatronic products where the control issue is not the essential problem, where we need a support regarding theory, models, methods and procedures. These other types of mechatronic projects, where the control issue is not the main challenge, as in the case study of the thermometer unit, will be the aim of our further research. In our future work there will be a focus on support for creating and handling alternative solutions in the development process and how to model relations to reveal the consequences of our dispositions in one domain to another. We see it of paramount importance to develop a support that will work in highly dynamic development environment, where decisions and changes occur rapidly, paradigm shifts are expected for the concept and simultaneous concepts are developed.

6 CONCLUSION

The case illustrates that generating a mechatronic solution involves a closely coordinated synthesis process between the mechanical, the electronics and the software engineers in the terms of understanding the multiple relations between the domains, which far exceeds what can be specified by defining physical interfaces and communication protocols. The literature review reveals gaps indicating that we do not have sufficient theory or methods for mapping and handling the relations needed to perform a transparent and rigorous synthesis of mechatronic products. If this is not provided, companies can be forced to resort to incremental innovation to lower the complexity of new products or settle for theories for general collaboration between disciplines. The authors of this article, however, believe that the mechatronic aspects should be treated by use of systematic views, understanding patterns and methodologies, which are specifically linked to the mechatronic area and that any support that can aid the designers in handling functions and properties distributed between the different domains will greatly enhance the quality of the design process. Further research should therefore be aimed at finding support for the seven aspects presented in this article.

REFERENCES

[1] Salminen, V. and Verho, A., Multi-disciplinary design problem in mechatronics and some suggestions to its methodical solution in conceptual design phase, In *Proceedings of International Conference on Engineering Design,* 1989, pp533-554.

[2] Tomiyama, T. et al., Complexity of Multi-Disciplinary Design, *CIRP Annals - Manufacturing Technology,* Vol. 56, 2007, pp185-188.

[3] Gausemeier, J. et al., Specifying the principle solution in mechatronic development enterprises, In *2008 IEEE International Systems Conference Proceedings,* 2008, pp160-166.

[4] Andreasen, M. M.and McAloone, T. C., Joining three heads - experiences from mechatronic projects, *Proceedings of Design for X,* 2001.

[5] Hansen, C. T. and Andreasen, M. M., *Two approaches to synthesis based on the domain theory,* In Engineering Design Synthesis, 2002 (Springer-Verlag, London).

[6] Buur, J., *A Theoretical Aproach to Mechatronics Design,* PhD thesis, Institute for Engineering Design, Technical University of Denmark, 1990.

[7] VDI 2206, *Design methodology for mechatronic systems,* 2003 (Beuth Verlag, Berlin).
[8] Sage, A. P. and Rouse, W. (Ed.), *Handbook of systems engineering and management,* 2nd ed., 2009 (Wiley-Blackwell, Oxford).
[9] Blanchard, B. S., *Systems Engineering and Analysis,* 3rd ed., 1997 (Prentice Hall, Upper Saddle River, N.J.).
[10] Pressman, R. S., *Software engineering : a practitioner's approach,* 6th ed., 2005 (McGraw-Hill, New York).
[11] Pahl, G. and Beitz, W., *Engineering design : A systematic approach,* 2nd ed., 1996 (Springer, London).
[12] Ulrich, K. T. and Eppinger, S. D., *Product design and development,* 2nd ed., 2000 (Irwin McGraw-Hill, London).
[13] V-Model XT, Available at: http://www.v-modell-xt.de.
[14] Buur, J., A framework for mechatronics design methodology, In *Proceedings of International Conference on Engineering Design,* 1989, pp507-518.
[15] Hubka, V. and Eder, W. E., *Theory of Technical Systems,* 1988 (Springer, Berlin).
[16] Andreasen, M. M., *Syntesemetoder på systemgrundlag,* PhD thesis, Lunds Tekniska Högskola, 1980, (In Danish).
[17] Gausemeier, J. et al., Conceptual design of mechatronic systems supported by semi-formal specification, In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics,* 2001, pp888-892.
[18] Jansen, S., *Eine Methodik zur modellbasierten Partitionierung mechatronischer Systeme,* PhD thesis, Ruhr-Universität Bochum, 2007, (In German).
[19] Welp, E. G. and Jansen, S., Domain allocation in mechatronic products, In *Proceedings of the 8th International Design Conference,* 2004, pp1349-1354.
[20] Bonnema GM. *Insight, Innovation, and the Big Picture in System Design.* Wiley Online Library 2010.
[21] Salminen, V.and Verho, A., Systematic and innovative design of a mechatronic product, *Mechatronics,* Vol. 2, 1992, pp257-275.
[22] Verho, A. and Salminen, V., Customer needs, function structure and strategic development of mechatronic product, In *International Conference on Recent Advances in Mechatronics,* 1995, pp105-110.
[23] Isermann, R., *Mechatronic Systems,* 2003 (Springer, London).
[24] Bishop, R. H. (Ed.), *The Mechatronics Handbook,* 2002 (CRC Press, Boca Raton, Fla.).
[25] Bracewell, R. H. et al., Schemebuilder, A design aid for the conceptual stages of product design, In *Proceedings of International Conference on Engineering Design,* 1993, pp1311-1318.
[26] Bracewell, R. H.and Sharpe, J. E. E., Functional descriptions used in computer support for qualitative scheme generation - 'Schemebuilder', *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* Vol. 10, 1996, pp333-345.

Contact: Jonas Mørkeberg Torry-Smith, PhD student
Section of Engineering Design and Product Development
Department of Management Engineering
Technical University of Denmark
Produktionstorvet, building 426
DK-2800 Kgs. Lyngby
Denmark
Tel: Int +45 4525 6274
Email: jtor@man.dtu.dk
URL: www.man.dtu.dk

11.2 PAPER B

"Mechatronic Design - Still a Considerable Challenge"

Published in the proceedings of the ASME 2011 Design Engineering Technical Conferences & Computers & Information in Engineering Conference (ASME IDETC/CIE 2011)

Proceedings of the ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2011
August 29-31, 2011, Washington, DC, USA

DETC2011-48306

MECHATRONIC DESIGN - STILL A CONSIDERABLE CHALLENGE

Jonas Mørkeberg Torry-Smith
Sofiane Achiche
Niels Henrik Mortensen
Department of Management Engineering
Technical University of Denmark
Copenhagen, Denmark

Ahsan Qamar
Jan Wikander
Carl During
School of Industrial Engineering and Management
Department of Machine Design
KTH Royal Institute of Technology
Stockholm, Sweden

ABSTRACT

Development of mechatronic products is traditionally carried out by several design experts from different design domains. Performing development of mechatronic products is thus greatly challenging. In order to tackle this, the critical challenges in mechatronics have to be well understood and well supported through applicable methods and tools. This paper aims at identifying the major challenges, by conducting a survey of the most relevant research work in mechatronic design. Solutions proposed in literature are assessed and illustrated through a case study in order to investigate, if the challenges can be handled appropriately by the methods, tools, and mindsets suggested by the mechatronic community. Using a real world mechatronics case, the paper identifies the areas where further research is required, by showing a clear connection between the actual problems faced during the design task, and the nature of the solutions currently available. From the results obtained from this research, one can conclude that although various attempts have been developed to support conceptual design of mechatronics, these attempts are still not sufficient to help in assessing the consequences of selecting between alternative conceptual solutions across multiple domains. We believe that a common language is essential in developing mechatronics, and should be evaluated based on: its capability to represent the desired views effectively, its potential to be understood by engineers from the various domains, and its effect on the efficiency of the development process.

1 INTRODUCTION

The design of mechatronic products is a multidisciplinary activity and is performed to attain product related advantages, which cannot be obtained by mono-disciplinary efforts. Along with the benefits from having several engineering disciplines involved in the design activity, complexity of the task increases accordingly. Since a mechatronic product is composed by solutions from the areas of mechanics, electronics, and computer software, special attention has to be paid to dependencies in the product and between the design activities. A lack of sufficient attention to the dependencies causes integration problems and increased development cost [1].

The aim of this paper is to gain a good understanding of the challenges related to design of mechatronics (referred to as mechatronic challenges hereafter), in order to help improving the development of solutions for mechatronic designers. A systematic and a thorough literature review is carried out to determine the mechatronic challenges and their proposed solutions as presented by researchers. The remaining part of the paper is organized as follows: Section 2 presents the methods utilized to build up the literature review. Section 3 presents a discussion on selected literature and data analysis to pinpoint the mechatronic challenges. Section 4 evaluates current solution support and builds up an understanding of important challenges, which are evaluated to be not well addressed. The case study in Section 5 is utilized to present real world mechatronic design scenarios, and to argue about how well they are supported through current solutions. The paper concludes by a discussion in section 6 and a conclusion in section 7.

2 METHOD OF INVESTIGATION

The objective of this paper is to identify the mechatronic challenges, assess their solutions, and illustrate those challenges and solutions through a case study. In order to accomplish this, a literature study is carried out, incorporating contributions from two sources. The first source consists of researchers from the ASME mechatronics community, whereas the second source is based on the collective knowledge of the authors of this article about important contributions within the research of mechatronic design complemented with a workshop to add on the list of researchers. For the first source, a filter function is needed to sort the vast amount of contributions in which mechatronic challenges are described. The filter function is based on the idea of extracting a large number of references from mechatronics related articles from the ASME conference, and then selecting the most cited researchers. The proposed solutions to the stated challenges are obtained from sources 1 and 2, and from the knowledge of the authors regarding solutions available from the literature. In order to illustrate the findings in terms of challenges and their solutions a case study is used.

3 LITERATURE STUDY

The goal of the first part of the literature study regarding the ASME community is to find the most reported and described challenges. This will be explained in detail in the following sections.

3.1 The Procedure for gathering the data

The three most recent ASME IDETC/CIE conferences are selected for the search, namely the 2008, 2009 and 2010 conferences. The process of finding significant literature is based on identifying researchers who have published mechatronics related articles, and researchers who are cited in the mechatronics community, since both constitute a significant contribution. The aim is therefore to find mechatronics related articles and subsequently extract the references to see who are referenced the most in the community as a proof of relevance. Firstly, articles dealing with the mechatronic design process have to be identified. This is done by using the keyword 'mechatronics'. If it is ambiguous whether or not the article would describe issues related to the mechatronic design process, the article is read to clarify the content. From the resulting 20 articles, 508 references are extracted.

3.2 Data analysis

The 508 references extracted from the ASME conferences are analyzed by a word-count software to reveal the names that appear the most. This quantitative evaluation is backed up by a qualitative scrutinizing of the reasons why the researchers are ranked as they are. Since it is common that authors cite their own previous work, a precondition is made that an author cannot appear more than once in the reference list of an article. The result of this evaluation is presented below in terms of a name and a numbered code. The first number shows the number of articles in which the researcher has been cited. The second number shows how many times the researcher has been cited in total. The third number is how many articles the researcher has published in the investigated conference proceedings (among the 508 extracted references). G. Pahl (11/12/0), W. Beitz (11/12/0), K.L. Wood (6/13/0), T. Tomiyama (6/13/1), C.J.J. Paredis (5/8/1), R.B. Stone (4/24/1), N.P. Suh (5/6/0), S.W. Szykman (5/6/0), J. Hirtz (4/4/0), D.A. McAdams (4/17/1), T.R. Browning (4/7/0), J.P. Clarkson (4/9/2), J. Gausemeier (4/17/1), U. Frank (4/9/0), U. Lindemann (4/16/2), A. Schmidt (4/11/2), Y. Umeda (4/7/0), M. Yoshioka (4/4/0). Researchers who are cited in less than four articles are omitted from the list, since it is assumed that the above undiscovered list of researchers will cover the needed challenges. Furthermore the number of researchers to consider has to be kept to a manageable level.

The presented search algorithm has limitations. It does not take into account if close colleagues are citing each other, or if the researcher is cited because his/her work is claimed not to be 'sufficiently good' by others. Even though the impact of the research might not be directly reflected by the number of citations, the identified researchers are considered to have contributed significantly to the mechatronic community. Therefore their formulation and insight into the challenges faced by the design teams when developing mechatronic products are of importance to this study.

When investigating the researchers and their co-authors, certain research groups appear due to preferred research partners. In the following, the researchers from the list presented above are listed with their preferred research partners.

- Pahl group: Pahl, Beitz.
- Wood group: Wood, Hirtz, Stone, McAdams.
- Tomiyama group: Tomiyama, Umeda, Yoshioka.
- Gausemeier group: Gausemeier, Frank, Schmidt.

Since researchers within a group tend to have similar views on mechatronics challenges, the grouping simplifies the data analysis.

3.3 Researchers added from the second source

We believe that it is beneficial to extend the systematically generated list with other researchers, who, to our knowledge, have relevant work regarding design of mechatronics. These researchers come from the second source, where the articles are obtained by performing a general search for publications related to design of mechatronics. This search targets the conferences under the design society, relevant journals along with knowledge about mechatronics research groups located at various places around the world. 50 relevant articles from these sources are analyzed and shortlisted based on their significance and relevance towards design of mechatronic products. This provides a list of 19 articles. It is noted that about half of these articles are written by researchers also appearing on the ASME list. Those researchers who are either not cited or who did not publish in ASME proceedings in the last 3 years include Buur [2], Salminen [3], Andreasen [4], and Adamsson [5]. The primary commonality among these researchers is their focus on the conceptual phase of the development life cycle, along with their emphasis on promoting collaboration between designers during the design activity.

The new extended list of researchers was then discussed in a workshop with researchers belonging to the 'Section of Engineering Design and Product Development' at the Technical University of Denmark, and the joined list was judged to be comprehensive.

Table 1: Matrix relating mechatronic challenges and researchers stating them. [1] = Source 1 researchers, [2] = Source 2 researchers

Category	#	Challenges	Researchers/Research Groups												
			Pahl[1] [6]	Wood[1]	Tomiyama [1]	Paredis[1]	Suh[1]	Browning[1]	Gausemeier[1]	Adamsson[2]	Buur[2]	Salminen[2]	Andreasen[2]	Lindemann[2]	Shea[2] [7]
Product	A	Lack of a common understanding of the overall system design			X	X		X	X	X		X		X	X
	B	Difficulty in assessing consequences of selecting between two alternatives			X	X		X	X		X	X			
	C	Lack of a common language to represent a concept			X	X			X	X	X	X	X		X
	D	Modeling and controlling multiple relations in the product concept	X		X	X	X	X	X	X				X	
	E	Being in control of the multiple functional states of the product		X			X		X				X		
	F	Transfer of models and information between domains (expert groups)		X		X			X						X
Activity	G	Synchronizing development activities						X	X				X		
Mindset	H	Different tradition within the domains for how to conduct creative sessions										X			
	I	Reluctant to interact with engineers from other disciplines										X			
	J	Different mental models of the system, task and design related phenomena			X	X		X	X	X	X	X	X		
Competence	K	Lack of common language to discuss freely at creative meetings			X				X	X	X	X	X		X
	L	Education within disciplines do not call for integration in professional life						X			X				
	M	The nature of design is different			X				X	X	X		X	X	
Organizational aspects	N	Product complexity affects the organization complexity			X			X	X						
	O	Knowledge transfer between domains is inadequate			X				X			X			
Other Aspects	P	Lack of a broadly accepted methodology			X	X			X		X	X	X	X	
	Q	Mechatronic ownership is lacking								X		X	X		
	R	System engineers are lacking detailed information of the system										X			
	S	Complexity as a generic problem	X	X	X	X	X	X	X	X					

3.4 Challenges identified

It is now possible to extract the statements regarding challenges in mechatronic design. For this purpose, between three and five papers from each researcher or each research group are investigated. Based on the extracted statements from each researcher, a KJ [8]equivalent methodology is applied, by which clustering of statements can be performed. A headline for each cluster is then formulated which should embrace the statements clustered in it. In Table 1, these headlines for the challenges are listed. The highlighted rows in the table are used for illustration of points discussed in section 4 and, thus will not be discussed in this section. In Table 1, the link to the researchers whose work complies with the stated challenges is also marked. The stated challenges are causally linked. As an example, the 'lack of common methodology' leads to a 'lack of a common representation of a product concept'. However, the causal chains will not be discussed further in this article.

Table 1 cannot be assessed quantitatively since the pool of data, being the number of researchers investigated, is relatively small and because the filtering process has distorted the picture of how many times a specific challenge is mentioned. The distortion occurs because it was chosen to group some of the researchers, which affected the number of times, the challenges appeared.

In this paper, for the sake of simplicity, all the identified challenges in Table 1 are assumed to be generic and hence, not context specific. This is done regardless of the number of times these challenges are repeated by the researchers.

In Table 1, some of the researchers stand out. These are Pahl and Beitz, the Wood group and Suh. Their work is often cited due to their fundamental contribution to design theory, and even though they address mechatronic or complexity issues in their work, a large part of the mechatronic-specific challenges in Table 1 are unaddressed.

What have we gained by introducing the researchers from source 2? The conclusion falls in two parts. Firstly: The fact that a large part of the challenges is also stated by the researchers from source 2 supports the claim that the challenges stated by source 1 are truly generic and thus important to direct attention to in research. Secondly it can be observed that some of the challenges are only described by researchers from source 2. Even though they are not validated to the same extent as those described by both sources, they still add to the understanding of the multidimensional challenges experienced by design teams developing mechatronic products. The researchers contributing with new challenges are Adamsson, Buur, Andreasen, and Salminen.

The most commonly reported sets of challenges are primarily related to the way a product concept can be described and how information linked to the product concept can be shared across engineering disciplines. The commonly observed challenges are (the highlighted rows in Table 1 is not linked to this list)): 'A Lack of common understanding of the overall system', 'A lack of a common language to represent concept', and 'A lack of a common language to discuss freely'. As stated by many of the researchers, the fundamental reason leading to the many challenges is the absence of a common mechatronic design methodology. This is again rooted in the fact that theories building upon different axioms cannot be joined to a common theory, as described by Tomiyama [1].

4 SOLUTIONS PROPOSED

This section will present a number of solutions to the mechatronic challenges, which are compiled through the literature study presented in Section 3.4. When there is sufficient documented evidence that a certain proposal addresses one or several challenges in design of mechatronics, we consider it as a solution as listed in Table 2. The table shows the challenges, which a given solution aims to support. The primary focus of a solution is represented by an orange cell. A 'Y' marks that a challenge is sufficiently addressed by a solution, whereas 'P' indicates that it is partially addressed. The process of allocating the 'Y' and the 'P' was carried out by the authors of this paper who are active researchers within the area of mechatronics.

A general overview of Table 2 shows that the mechatronics challenges are not sufficiently addressed by the proposed solutions. Specifically, solutions for challenges B, C, F, G, K, M, N, O, P, R, and S are either partially defined, or no solution is proposed. Among these challenges, there are challenges that relate to 1) competences (K, M), 2) to activities (G), 3) to organizational level (N, O). These are not treated further in this paper, since our scope is not towards competences of individuals in a company, nor the synchronization of activities or the organizational issues. Challenges B, C, and F, and challenges P, R, and S are strongly connected to each other because of the following reasons: Since there is a lack of broadly accepted methodology (P) in mechatronics, a common language to represent the concepts can be difficult to accomplish. This creates a problem of finding the most suitable design through efforts across different domains. Along with difficulty in assessing consequences, the lack of methodology and the lack of a common language contribute to a higher complexity (S) in mechatronics. In addition to that, the lack of common language and inadequate information transfer between domains are strongly connected to challenge R (system engineer lacking detailed information on the system). Therefore, to gain a detailed insight on some of the core challenges in mechatronics, we will restrict ourselves to challenges B, C and F, which we believe are at the heart of mechatronic challenges. B, C and F are marked in green in Table 2. The other challenges are also important, but not treated further to limit the scope of this paper.

In the following, each solution is discussed and assessed about how well it supports challenges B, C and F.

1) The first solution from Table 2 is about methods based on functional thinking. Buur [2], Wood [9], Tomiyama [1], and Suh [10] are examples of functional approaches. Functional modeling is abstract in terms of the level at which the description of the product concept is performed. Therefore it can serve the purpose of a common modeling language (C) to an abstract level only. It is typically after the functional modeling that the development process becomes domain-specific. Functional thinking is only part of the complete picture of the design activities, and other factors such as structural consequences and effects of system elements onto various system properties is not supported through it (B). Moreover, since models contain much more information than functions; functional thinking is only useful when transferring information between two abstract system models (F).
2) The second solution is about modeling relationships between elements from different mechatronic domains. Design structure matrix (DSM) and domain mapping matrix (DMM) by Lindemann [11], [12], and Browning [13] are examples of modeling relationships between functions, components, physical structure, and resources during a mechatronic product development process. Bonnema [14] proposes a FunKey (function keydrivers) architecting approach to model relationships between functions and key drivers of a product, in an aim to provide good insight to different stake holders while designing. The main aim of these approaches is to support the managing of multiple relations and dependencies during design. However,

analyzing the consequences of selecting between different alternatives becomes too cumbersome through these approaches in terms of effort and efficiency (B). Moreover, they are meant for establishing relations across different domains, and not for information transfer between design models (F). They are partially suitable for aiding a common language (C), since engineers from different domains can discuss dependencies based upon them.

3) The third solution is about controlling integration between domains via requirements. Systems Engineering [15] and work by Tomiyama [16] are examples of such solutions. However, requirements cannot be utilized for accessing consequences (B) of different design alternatives for a mechatronic system. Therefore, model-based system engineering [17] proposes to utilize requirement management tools in addition to system-level modeling (common modeling language (C)) to control system design based on requirements. This provides a better utilization of requirements through a computer support. A model transformation between system-level models and domain specific models (F) is however required to keep the design models consistent with each other.

Table 2: Solutions proposed in literature against the challenges identified in Section 3.4. An orange cell indicates the primary aim of the solution, and the green columns shows the most important challenges

#		Challenges																		
	Solutions	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Activities based on functional thinking (Buur [2], Wood [9], Tomiyama [1], Suh [10]), applying functions, means patterns, state and event relations (process related)	Y		P	Y	Y					P		Y							P
2	Relationship management e.g. DSM and DMM [11], [13], QFD [18], FunKey [14]		P		Y			P			P					P				P
3	Controlling design activities through requirements management (Systems engineering [15], Tomiyama [1])			P			P				P	P		P						
4	A process model containing activities for the development process. (Isermann [19], VDI2206 [20], Salminen [3], Systems Engineering [15])							P						P	P	P				P
5	Informal description consisting of a number of modeled/described aspects to specify systems, A3 overviews [21], Salminen [3], Buur [2]	Y	P	P	P						Y		P			P				P
6	Modeling languages to describe system as a whole, formally or semi-formally. SysML [22], SFSL by Gausemeier [23]	Y	P	P	P						P									
7	Model transformation from a design model in one domain into a design model in another domain (Gausemeier [24], Paredis [25])						P					P		P		P				P
8	Formalized specification of interfaces. (ISO/IEC 81346 [26], Systems engineering [15])				P					P	P			P	P					P
9	Simulation of phenomena that incorporates elements from the different domains (e.g. Modelica [27])		P		P	P	P													
10	Setting up a systems integration group in the project (Adamsson [5], Andreasen [4])	Y			P			P	Y	Y	P	P				P		Y		P

4) Different process models, specifying the activities to be performed during the design process are proposed by several researchers. These process models are usually an extension of a process model in one domain towards covering several domains. VDI2206 [20], Systems Engineering [15], and work by Isermann [19] and Salminen [3] are examples of such models. These models aim at synchronizing the workflow and activities, which the design team must perform during the development. However, these approaches state that dependencies should be handled, not how to actually manage them in relation to assessing consequences (B). A process model urges to utilize a system design language such as SysML for systems engineering. However, process models themselves do not solve the common language challenge (C). The same applies to model transformations (F), which can be made a part of the process models, but process models themselves do not aim at solving challenge F.

5) In the aim of a common language and to solve the communication problems during conceptual design phase, different solutions are proposed (solutions 5, 6). An example of an informal description is the A3 architecture overviews [21], which provides an overview of the complete system, in terms of different system aspects, such

as functional aspects, physical aspects, etc. Although representing system design concepts informally is useful to discuss system design among different domain experts, however, assessing consequences for each design domain while choosing different system alternatives is not addressed (B). Moreover, such overviews have the same potential of becoming a common language as the functional thinking proposed by Buur [2]. Hence, there are other abstraction levels in design that cannot be supported by A3 overviews (C). Presenting concepts in A3 overviews can lead to gaps between domain-specific design activities and system-level design activities, and it is clear that model transformations cannot be utilized with A3 overviews to reduce this gap (F).

6) An attempt towards a language more specifically related to mechatronics is the semi-formal specification language (SFSL) by Gausemeier [24], which aims at specifying a mechatronic concept in terms of a number of aspects, such as a behavior-aspect and a structural-aspect. Modeling languages having formal semantics that describe the system in terms of different views are also proposed such as SysML [22]. The opinions from researchers behind these modeling language approaches contain a contradiction, especially in terms of their usefulness and effectiveness. For example Borches et al. [28] document that formal modeling such as SysML does not usually solve the communication problem between people from different design domains, nor does it produce models that are easy to understand. The fragmentation of proposals for a common modeling language by different groups of researchers indicates a need for further improvement in this area. Therefore it can be said that although a common design language is a need, the nature of such a language in terms of being formal or informal is still unknown, and there is still a need for developing support in this area.
7) Model transformations are proposed as a possible solution to relate two design models. Shah et al. [25] shows how a mapping between two design models can be used to build transformations between them. An example is the transformation between SysML [22] and Modelica [27], which combine the descriptive capability of SysML with analysis and simulation capability of Modelica. Formal models utilized during conceptual design phase have advantages of supporting automated-model transformations to other design models. However, dependencies between mechatronic domains cannot be directly solved through model transformations, because it is not always possible that a model contains a representation of all possible dependencies that arise while accessing consequences of different alternative design solutions (B). The dependencies that are important and the consequences that are critical to be considered are not necessarily known beforehand (the uncertainty element). Moreover, model transformations (F) can be more effective if a proposal for a common design language (C) in mechatronics becomes successful. However this is not an explicit goal of the model transformation community to develop such a language.
8) Besides intra-domain interfaces, interfaces can also be observed between domains, such as a shielding of an electronic sensor. An international standard exists (ISO/IEC 81346) that specifies how to define a physical interface. Furthermore, clearly defined interfaces are stated as being advantageous [15]. The interface description aims at specifying the physical interfaces based on a functional partitioning between the domains. Therefore, interface handling can only provide some of the information needed for assessing consequences. Hence, B is not covered. Clearly stated interfaces cannot be used as a common language (C), even though it can be used as a framework for discussions. Model transformation (F) is decoupled from interface specification, and is therefore not covered.
9) Computer aided multi-domain modeling and simulation provides advantages of building design models with elements from different domains, along with executing them in order to assess certain product properties. These modeling languages are well supported by tools, in order to conduct simulation and analysis. Modelica is one example of such modeling languages. Although such languages provide support for assessing consequences (B) to an extent, they cannot be treated as a common design language (C) for all domain experts. Moreover, they are only good for design modeling when the basic principles and the basic structure of the product have been determined. Current efforts within the Modelica community aim to standardize model transformations (F) between SysML and Modelica. However this will only be useful if SysML is utilized.
10) Adamsson [5], and Andreasen [4] proposed setting up a systems integration group. This group is primarily responsible for facilitating the information flow, and the collaboration between engineers from the different domains to increase performance of the overall system. However, challenges B, C, and F are only supported indirectly by anticipating that an integration group will facilitate closer integration between the domains.

5 CASE STUDY

The purpose of presenting a case study in this article is to illustrate the three selected mechatronic challenges (B, C, and F) highlighted in section 4. This will allow us to relate the rather abstractly described challenges to a very concrete situation in a product development process. Additionally, the product case will help assessing how well the proposed solutions would have helped the design team in their design task. Therefore, the case study is not used for verifying whether or not a challenge can be handled by the proposed solutions. Instead, it is used to bring in a real-world dimension, and create a context surrounding the challenges, and thereby understand better what it might take to create satisfactory solutions.

The aim of the project, chosen as the case study, was to develop a watch system based on the idea to develop a mechanical watch and an instrument, which can be attached to the watch (see Figure 1). The instrument contains advanced functions used for alpine skiing. An additional two external units wirelessly transmit heart rate information and temperature information to the instrument attached to the watch. In this case study, we focus on the external temperature unit showed in Figure 2. It is noteworthy that one of the authors was involved as a development engineer in this specific project.

Figure 1: 1. Instrument, 2. Watch, 3. Temperature Unit, 4. Heart-rate sensor, 5. Charger for instrument

The case study was built up on the experience gained by participating in the development team backed up by document-analysis, and interviews with the project managers for the mechanical, electronics and software development. Due to limitations of describing the development process as a whole, we deem it necessary to only select small fragments from the design process to illustrate the selected challenges. In the following, three scenarios from the case study are presented, which are directly related to what we consider as the most important challenges (B, C, and F). This is followed by a discussion on possible solutions from Table 2, and a conclusion on using those solutions to mitigate the particular challenge.

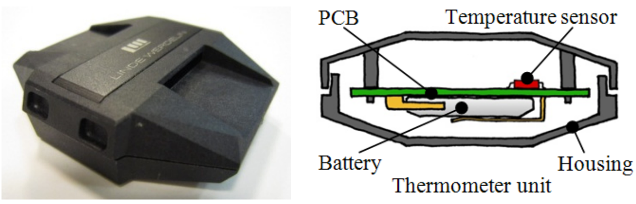


Figure 2: Temperature Unit

5.1 Assessing consequences (Challenge B)

The Power consumption scenario: In the beginning of the project, it was assessed that the power consumption would be one of the major key drivers for the project. The RF chip for wireless transmission and a running processor are the primary sources for the power drainage. The main electronic components are illustrated in Figure 3. Two basic approaches can be chosen: either to minimize the power consumption (thereby the user should change batteries), or to make the whole unit rechargeable. Within the scope of minimizing the power consumption, two main directions can be chosen, which is either to cut the power manually or automatically when it is not in use, or to minimize power usage by features in the electronics, and by clever programming. Solutions are spread over all the domains. Some solutions have a direct effect on the use pattern, hence the user experience. Some solutions require further technology clarifications. Other solutions require the consequences on the products life phases (e.g. change of battery) to be assessed. The main challenge is that there are many conceptually different ways of solving the power issue, but how can we, in the best possible way, reason about the consequences of selecting one product concept above another? The problem of assessing the consequences when choosing between concepts is a general concern in product development. Yet, this concern increases when different domains are involved in the design process while investigating alternative design concepts.

Discussion on solutions: In Table 2 four solutions have been identified, which potentially should embrace the challenge of assessing the consequences by selecting between two product concepts: a) Relationship management; b) Informal descriptions; c) Formal language description; d) Mechatronic concept description and simulation of phenomena. DSM, MDM, QFD as well as formal modeling languages such as SysML and the various simulation programs only provide a description of a single or few closely related properties or aspects. In the case study, a holistic approach is needed to consider the consequences of a product concept, which the mentioned mechatronic solutions cannot encompass. In the project, various concepts were sketched to reveal their potentials and drawbacks and to evaluate the life phases. The product concepts were then discussed on several meetings and the progression of reducing the needed power was continuously assessed.

Modeling languages exist ranging from the formal modeling languages such as SysML over semi-formal modeling languages such as Gausemeier's SFSL, to less restricted modeling such as the A3 overviews. Even the A3 overviews, which is proposed as an informal method, is not sufficient, since it does not address mechatronic specific aspects such as the implications of different allocation of functions to the domains. An informal description different from the A3 overviews, seems to be the best way to mitigate the challenge since an informal description is flexible. The question is, however, is the informal description so flexible that it does not provide any mechatronic specific support? The answer seems to be yes. In the presented case, the solutions from Table 2 seem even less appropriate than evaluated in the table to solve the problem of assessing the consequences by selecting between two or more product concepts.

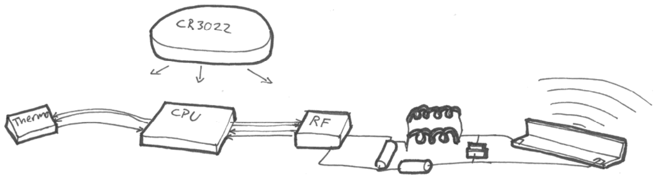


Figure 3: Main Electronic Components

5.2 Common language to represent a concept (Challenge C)

'The custom made gasket' scenario: A request for changing the outer shape to make the unit appear lighter causes a change of the mechanical design (Figure 4). The changed design makes less space for fitting the main gasket, which ensures the water resistant property of the unit. Instead of the previous used standard *O- ring*, a custom shaped gasket must be used unless the outline of the PCB is changed (Figure 4). At this late stage of the electronic development, a change of the PCB would result in reorganizing the electronic components. In a HF circuit, the relative placement of components affects the transmission quality, thereby increasing development cost, risk and time if the PCB layout were to be redesigned. Therefore a custom made gasket is evaluated as 'the best fit' solution.

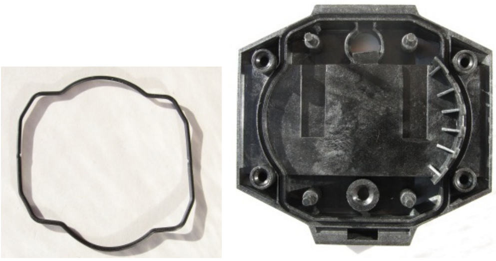


Figure 4: The custom made gasket and the part in which it has to be inserted

Discussion on solutions: The situation as described above is a known characteristic of the design of mechatronics, where the best alternative among few has to be chosen, such as changing the gasket or changing the PCB. However, there are consequences attached to each alternative for different design domains, such as the redesign cost of the PCB, the redesign cost of the gasket and the mechanical module, the packaging of the high-frequency electronics, and the success probability of the integration test. The dependencies between different domains during the design activity are major contributors towards these consequences. For example, the relation between the gasket and the size of the PCB. Moreover, the best solution has to be considered in terms of the overall system, and not just between domains. Considering Table 2, the common modeling language proposals such as SysML, SFSL, and A3 overviews can be considered to build up a system view, hence supporting modeling and evaluation of alternatives in terms of the system as a whole. Moreover, DSM/DMM, and FunKey architecting are also proposals to identify relations between functions and user demands, and between functions and components. However, DSM/DMM and FunKey architecting serves the purpose of relationship management only. Building a holistic system view along with assessing certain characteristics of the system such as performance or cost is not supported. From Table 2, activities based on functional thinking, and controlling design through requirements are also proposed as solutions for a common language to describe the concept. However, functional thinking is proposed to describe only the functional view of the product, thereby leaving out the structural view which is essential to the gasket issue. In the case of requirements, they can be used for goal specifications (of the product to be), or result specifications (documenting the finalized product), but requirements cannot be used to represent a design concept.

Considering SysML, SFSL, and A3 overviews, these languages provide different solutions towards representing the size constraint relation between the gasket and the PCB. This constraint modeling enables mechanical and electrical/electronic engineers to understand the effects of gasket size on the PCB. It also relates this constraint to the complete system model. However, the decision for whether to redesign the gasket, or redesign the PCB requires assessing consequences of each alternative in relation to designer preferences. We believe that availability of an informal and visual language, where designers from different domains can sketch their ideas to each other, and highlight the relationship of their concepts to each other, is a more effective way of managing dependencies such as between gasket and PCB. The sketching can be partially or fully supported by a calculation or a simulation engine (depending upon how open/restrictive the visualization is). SysML, SFSL, and A3 overviews are common in certain aspects; however, they differ in terms of being formal, semi-formal or informal. Especially A3-overviews is an informal medium to discuss such dependencies between views. However, it does not target how can these dependencies be understood and managed during the design activity. It rather defines a medium where these dependencies can be expressed in a way that is understandable to different engineers. The usefulness of formal or semi-formal modeling (such as SysML) is explained to be not useful in the conceptual design phase due to the rate at which models change, and due to decreased communication effectiveness caused by a lack of visual representation of the structure of the product by different engineers [28].

5.3 Transfer of models (Challenge F)

'The ESD protection issue' scenario: Due to a requirement for better temperature sensing, a change of the design is necessary. Discussing the proposed solution with the electronic engineers, it becomes apparent that this type of solution is prone to electrostatic discharges and that mitigations have to be made for the electronics not to be damaged in such a case. The proposed design is shown in Figure 5.

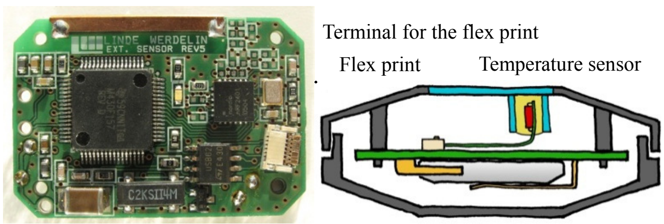


Figure 5: The PCB and the positioning of the flex print and the flex print terminal

For easier handling of the small thermo sensor, it is placed on a flex print which can easily be connected to the PCB compared to five ordinary wires. Due to the stiffness of the flex print, the location and orientation of the terminal is important and this fitting is made in corporation between the electronics and the mechanical engineer. Figure 5 shows the PCB connector placement and the position of the flex print.

Discussion on solutions: In this particular case, the orientation and location of the terminal on the PCB, and the placement of the connector on the flex print is a clear dependency between electronics and mechanical models. In order to reach a solution, both electronics and mechanical engineers had to have several discussions during a number of design iterations. From Table 2, three solutions have been identified which should aid in overcoming this challenge related to information transfer across domains: a) Controlling the design through requirements management; b) Simulation of phenomena incorporating model elements from different domains; c) Integration of models through model transformations.

Requirement specifications play a key role in controlling the design, and hence it is proposed to utilize these specifications as a solution to ease the information transfer between domains. Traditionally, a specification has to direct the search for solutions. What is required (here) for information transfer is the detailed information/representation of needed parameters of a concept from each domain, and not the specification that directs search for those concepts. Simulation of properties is also proposed as a possible solution to ease the information transfer between domains. However, in the above case, the mechanical and electronics engineer need information regarding the location of the terminal. Hence, simulation in the sense of algorithmic optimization cannot be utilized for this task.

Integration of models through a model transformation such as [25] and [24] is proposed as a solution to aid in information transfer between domains. The location of the connector in the mechanical design model can be extracted and represented through a transformed model i.e. an electronic design model to facilitate the electronic engineer during the design process and vice versa. In the following, the relationships between models are more closely described to be able to evaluate challenge F in terms of performing a model transformation.

Different design models are related in terms of system properties which they affect. Although two design models may both affect one system property, there is only a portion of each model that has substantial meaning in the other model. Tomiyama et al. [29] explain that two models can only be integrated with each other if the background theories (that these models are based on) are compatible. The compatibility between two background theories suggests that a concept in one theory can be related to a concept in another theory. For example, inertia has no meaning in electronic PCB design, but has a meaning in controller design. If two background theories are compatible, then a model transformation can be applied to the corresponding models. Model transformation approaches provide a capacity to control which part of the source model is read and what is created in the target model by specifying meta-models and the transformation between them. Therefore, we conclude that model transformation has a potential in addressing challenge F. In the following, model transformation approaches will be discussed further, followed by concluding remarks on the limitations of a model transformation.

One approach for integration of models is to utilize a central product model where all the information is stored. The central product model can be utilized to understand and manage the relationships between different aspect models. The aspect models can also be generated from the central product model. Another approach for managing relationships between models is where an integration at the level of background theories is proposed to support integration between so called ‘multiple aspect models’ [29], [30], and [31]. The approach is based on developing different aspect models based on different background theories, e.g. dynamic models, and geometric models. These aspect models can be integrated through a central meta-model, where the relationships between the concepts of the different background theories are specified. Specifying the concepts and the concept relationships between the different background theories in a meta-model aids in managing the influence of a model element in one aspect model onto a model element in another aspect model. A similar approach is presented in the PACT experiment in [32], where an approach for integration among multiple aspects (agents) during design is discussed.

It is likely that a transformed model does not contain all the information that is required by a modeler, because it is not always possible to know at earliest stage which properties affect each other and hence, should be in the model. This information might be known at a later stage, and if these properties are not explicitly supported by the meta-model of a domain, then a model transformation will not be useful straight away, and will require further efforts. Hence challenge F is not fully addressed through model transformations.

In order to support the design process for mechatronic products, we propose model integration between domain-specific views such as a mechanical view and a system view built through a common system modeling language. This will provide an opportunity to find a best mechatronic design solution for a system. [25] and [24] are examples of steps in this direction. However since the nature of common modeling language is still an unknown, this area has a good potential for further development.

6 DISCUSSION

Most papers about mechatronic design end by stating that a common methodology and a common conceptual model is needed. This statement has been repeated for the last 20 years. If it was possible, it would have been likely that such a method would have been found, or significant findings presented which would be a step towards it. Proposals of a mechatronic concept description always end up by constituting different needed views. Having ‘x’ number of different views on a concept negates the idea of a common conceptual representation. In principle, this is not different from the acknowledgement that you need several different views of a system to be able to describe it, e.g. the proposal of the domain theory by Mogens Myrup Andreasen in the early 80’ies [33], also described in

[34]. Tomiyama [1] states that two theories cannot be joined if they are based on different set of axioms. This is the reason why the so-called 'common mechatronic concepts' always have to be presented by 'x' number of views. For each type of property one has to model, a separate view has to be created [35], [30]. However, there are some theories that by nature share the same axioms across the domains. Jacob Buur [2] pointed out the functional thinking as a common theory that can be applied to all domains. This will enable methods, which are based on functional thinking to be used across the domains in mechatronic development. Some of the methods based on functional thinking are: life phase thinking, process descriptions of the product, state-transitions, function/means tree, and QFD. Quite soon in the development process, one needs to model and evaluate properties of the design. Whether the property modeling is performed based on sketching and/or computer simulation, the problem of a common mechatronic model appears, because an evaluation of a property is linked to a certain theory which will be domain specific. To assess several properties from different domains in one model, no adequate theory or tool or process has been proposed. We suggest the following thinking experiment: If two competing concepts are developed to finalized products, the consequences can be fully evaluated. Since this is seldom carried out for obvious reasons, it is necessary to show the relations and consequences by other means. We have previously described that a common conceptual model, which has details beyond describing functionality in the product, would violate the fundamental axioms. Therefore, we have to accept that not all the relations can be modeled, besides those few which can be described as the key relations. We should be willing to work with ill-defined problems across the domains and willing to generate alternatives and most of all to be able to identify what information is relevant to share with developers from other domains. We should acknowledge the 'collaboration' research field (the human aspect), and provide room (workshops) and methods, which will enable cross-domain discussions, and which will be graphically intriguing.

7 CONCLUSION

In this paper, the challenges, which seem to be most significant in the design of mechatronics are presented. They range from product specific challenges to company-level challenges. The proposed solutions in the literature only provide partial solutions to those challenges. A large part of the identified solutions appear to support analysis rather than synthesis. As a product concept progresses, effort must be spent to continuously update the information that goes into the analysis-oriented solutions to be able to use them. This effort compared to what can be gained by using a particular solution is seldom assessed, evaluated or investigated in the literature. The solutions which are not analytical in nature are the ones based on functional reasoning, which have the capability of being applied across domains. Unfortunately, these solutions are not well described in terms of how to apply them to an actual synthesis process of a mechatronic product. Even though functional reasoning should be capable of supporting the design process through all the design phases, the suggested solutions only support the initial steps in the conceptual phase.

A common design language would, as stated by many of the researchers in the study, facilitate a better collaboration between engineering disciplines. A common language, if possible to develop, would need to consist of 'x' number of product views to be modeled, ruling out the prospect of a unified representation. Furthermore, a common language should be evaluated based on: its capability to represent the desired views effectively, its potential to be understood by engineers from the various domains, and its effect on the efficiency of the development process. If a common language can be realized, it would also facilitate in creating variations of the product concepts in the conceptual phase. The case study illustrated this as being beneficial to reveal the consequences of selecting between alternative design concepts.

ACKNOWLEDGMENTS

We wish to thank Morten Linde and Jorn Werdelin, who are the co-founders of the watch company LindeWerdelin [36] for letting us use the development project as the case for this article. We would also like to thank Troels Pedersen from IPU and Henrik Bendsen from Oticon for their help in adding detailed information to build the case.

REFERENCES

[1] T. Tomiyama, V. D'Amelio, J. Urbanic, and W. ElMaraghy, "Complexity of Multi-Disciplinary Design," *Annals of the CIRP*, vol. 56, no. 1, pp. 185-188, 2007.

[2] Jacob Buur. (1990, Institute of Engineering Design, Technical University of Denmark) A Theoretical Approach to Mechatronics Design, PhD thesis, http://orbit.dtu.dk/get Resource?recordId=277032&objectId=1&versionId=1

[3] V. Slaminen and A. J. Verho, "Multi Disciplinary Design Problems in Mechatornics and Some Suggestions to its Methodical Solution in Conceptual Design Phase," in *International Conference on Engineering Design (ICED89)*, 1989.

[4] Mogens Myrop Andreasen and Tim C. McAloone, ""Joining Three Heads" - Experiences From Mechatronic Projects," in *Design For X Symposium*, 2001.

[5] Niklas Adamsson, "Model - Based Development of Mechatronic Systems - Reducing The Gap Between Competencies?," in *Tools and Methods of Competitive Engineering*, 2004.

[6] G Pahl and W. Beitz, *Engineering Desing: A Systematic Approach*. London: Springer, 1996.

[7] Stefan Wölkl and Kristina Shea, "A Computational Product Model for Conceptual Design Using SysML," in *ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2009*, San Diego, California,

2009.
[8] Jacob Buur, "Design Methods in Japan: Research, Education and Industrial Application from a European Viewpoint," *Journal of Engineering Design*, vol. 2, no. 2, pp. 91-103, 1991, ISSN 0954-4828.
[9] Robert L. Nagel, Robert B. Stone, Ryan S. Hutcheson, and Daniel A. McAdams, "Function Design Framework (FDF): Integrated Process and Functional Modeling for Complex Systems," in *ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2008*, 2008.
[10] Nam P. Suh, "Applications of Axiomatic Design to Engineering Collaboration and Negotiation," in *International Conference on Axiomatic Design (ICAD)*, 2006.
[11] Stefanie C. Braun and Udo Lindemann, "A Multilayer Approach For Early Cost Estimation of Mechatronical Products," in *International Conference on Engineering Design (ICED07)*, 2007.
[12] Matthias Kreimeyer, Stefanie Braun, Matthias Gürtler, and Udo Lindemann, "Relating Two Domain Via a Third - An Approach to Overcome Ambiguous Attributions Using Multiple Domain Matrices," in *ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2008*, New York, 2008.
[13] Mike Danilovic and Tyson R. Browning, "Managing complex product development projects with design structure matrices and domain mapping matrices," *International Journal of Project Management*, vol. 25, no. 3, pp. 300-314, 2007.
[14] G. Maarten Bonnema, "Insight, Innovation, and the Big Picture in System Design," *Systems Engineering*, 2010, DOI 10.1002/sys.20174.
[15] A. P. Sage and W. Rouse, *Handbook of Systems Engineering and Management*, 2nd ed., A. P. Sage and W. Rouse, Eds.: John Wiley & Sons, 2009, ISBN 978-0-470-08353-6.
[16] Krijn Woestenenk, Hans Tragter, G. Maarten Bonnema, Andrés A. Alvarez Cabrera, and Tetsuo Tomiyama, "Multi Domain Design: Integration and Reuse," in *ASME 2010 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2008*, Montreal, 2010.
[17] Sanford Friedenthal, Alan Moore, and Rick Steiner, *A Practical Guide to SysML: The Systems Modeling Language*.: Morgan Kaufmann, 2008, ISBN 978-0-12-374379-4.
[18] Johan R. Hauser and Don Clausing, "The House of Quality," *Harvard Business Review*, 1988.
[19] Rolf Isermann, *Mechatronic Systems Fundamentals*. London: Springer-Verlag, 2005, ISBN 1-85233-930-6.
[20] Association of German Engineers (VDI), "VDI-guideline 2206 - Design Methodology for Mechatronic Systems," Berlin, 2004.
[21] P. Daniel Borches and G Maarten Bonnema, "A3 Architecture Overviews - Focussing Architectural Knowledge to Support Evolution of Complex Systems," in *20th Annual INCOSE Symposium (IS2010)*, 2010.
[22] Object Management Group. (2010) OMG System Modeling Language Specification V1.2. [Online]. http://www.omg.org/spec/SysML/1.2/PDF/
[23] Jurgen Gausemeier, Martin Flath, and Stefan Mohringer, "Conceptual Design of Mechatronic Systems Supported by Semi-formal Specification," in *IEEE/ASME International Conference on Advanced Intelligent Mechatornics*, 2001.
[24] Jurgen Gausemeier et al., "Management of Cross Domain Model Consistency During the Development of Advanced Mechatronic Systems," in *International Conference on Engineering Design (ICED09)*, 2009.
[25] Aditya A. Shah, Aleksandr A. Kerzhner, Dirk Schaefer, and Christiaan J. J. Paredis, "Multi-View Modeling to Support Embedded Systems Engineering in SysML," *Graph Transformations and Model Dirven Engineering-Lecture Notes in Computer Science*, vol. 5765, pp. 580-601, 2010, DOI: 10.1007/978-3-642-17322-6_25.
[26] 81346. (2011) ISO/IEC 81346. [Online]. http://www.81346.com
[27] Modelica Association. (2010) Modelica Language Specification V3.2. [Online]. https://www.modelica.org/documents/ModelicaSpec32.pdf
[28] P. Daniel Borches and G. Maarten Bonnema, "System Evolution Barriers and How to Overcome Them!," in *8th Conference on Systems Engineering Research (CSEA2010)*, 2010.
[29] Tetsuo Tomiyama, Takashia Kiriyama, Hideaki Takeda, Deye Xue, and Hiroyuki Yoshikawa, "Metamodel: A Key to Intelligent CAD Systems," *Research in Engineering Design*, vol. 1, no. 1, pp. 19-34, 1989.
[30] Takeshi Kiriyama, Tetsuo Tomiyama, and Hiroyuki Yoshikawa, "A Model Integration Framework for Cooperative Design," *Computer Aided Cooperative Product Development*, vol. 492, pp. 126-139, 1991.
[31] Hiroyuki Yoshikawa, Tetsuo Tomiyama, Takashi Kiriyama, and Yasushi Umeda, "An Integrated Modelilng Environment Using the Metamodel," *Annals of CIRP-Manufacturing Technology*, vol. 43, no. 1, pp. 121-124, 1994.
[32] M.R. Cutkosky et al., "PACT: An Experiment in Integrating Concurrent Engineering Systems," *Computer*, vol. 26, no. 1, pp. 28-37, 1993.
[33] Mogens Myrup Andreasen, "Machine Design Methods Based on Systematic Aprroach," Lund University, 1980.

[34] Claus Thorp Hansen and Mogens Myrup Andreasen, "Two approaches to synthesis based on the domain theory," in *Engineering Design Synthesis*, Amaresh Chakrabarti, Ed. London: Springer-Verlag, 2002, pp. 93-108, ISBN 1-85233-492-4.

[35] N. H Mortensen, "Design Modeling in a Designer's Workbench," Technical University of Denmark, 2000.

[36] Linde Werdelin. (2011) [Online]. http://www.lindewerdelin.com

11.3 PAPER C

“Challenges in Designing Mechatronic Systems”

Published in the Journal of Mechanical Design, January 2013, Vol. 135

Jonas Mørkeberg Torry-Smith
Department of Mechanical Engineering,
Technical University of Denmark,
Copenhagen, Denmark
e-mail: jtor@mek.dtu.dk

Ahsan Qamar
Department of Machine Design,
School of Industrial Engineering
and Management,
KTH Royal Institute of Technology,
Stockholm 10044, Sweden
e-mail: ahsanq@kth.se

Sofiane Achiche
Department of Mechanical Engineering,
Design and Manufacturing Section,
École Polytechnique de Montréal,
University of Montreal,
Montreal H3T 1J4, Quebec, Canada
e-mail: sofiane.achiche@polymtl.ca

Jan Wikander
Department of Machine Design,
School of Industrial Engineering
and Management,
KTH Royal Institute of Technology,
Stockholm 10044, Sweden
e-mail: janwi@kth.se

Niels Henrik Mortensen
Department of Mechanical Engineering,
Technical University of Denmark,
Copenhagen, Denmark
e-mail: nhmo@mek.dtu.dk

Carl During
Micronic Mydata AB,
Box 3141, 18303 Täby, Sweden
e-mail: carl.during@micronic-mydata.com

Challenges in Designing Mechatronic Systems

Development of mechatronic products is traditionally carried out by several design experts from different design domains. Performing development of mechatronic products is thus greatly challenging. In order to tackle this, the critical challenges in mechatronics have to be well understood and well supported through applicable methods and tools. This paper aims at identifying the major challenges, by conducting a systematic and thorough survey of the most relevant research work in mechatronic design. Solutions proposed in literature are assessed and illustrated through a case study in order to investigate if the challenges can be handled appropriately by the methods, tools, and mindsets suggested by the mechatronic community. Using a real-world mechatronics case, the paper identifies the areas where further research is required, by showing a clear connection between the actual problems faced during the design task and the nature of the solutions currently available. From the results obtained from this research, one can conclude that although various attempts have been developed to support conceptual design of mechatronics, these attempts are still not sufficient to help in assessing the consequences of selecting between alternative conceptual solutions across multiple domains. We believe that a common language is essential in developing mechatronics, and should be evaluated based on: its capability to represent the desired views effectively, its potential to be understood by engineers from the various domains, and its effect on the efficiency of the development process. [DOI: 10.1115/1.4007929]

1 Introduction

The design of mechatronic products is a multidisciplinary activity and is performed to attain product-related advantages, which cannot be obtained by monodisciplinary efforts. Along with the benefits from having several engineering disciplines involved in the design activity, complexity of the task increases accordingly. Since a mechatronic product is composed of solutions from the areas of mechanics, electronics, and computer software, special attention has to be paid to dependencies in the product and between the design activities. A lack of sufficient attention to the dependencies causes integration problems and increased development cost [1].

The aim of this paper[1] is to gain a good understanding of the challenges related to the design of mechatronics (referred to as mechatronic challenges hereafter). Our intention is to help improve the development of solutions for mechatronic designers. A systematic and thorough literature review is carried out to determine the mechatronic challenges and their proposed solutions as presented by researchers. The remaining part of the paper is organized as follows: Section 2 presents the methods utilized to build up the literature review. Section 3 presents a discussion on selected literature and data analysis to pinpoint the mechatronic challenges. Section 4 evaluates current solution support and builds up an understanding of important challenges, which are evaluated to be not well

Contributed by the Mechanisms and Robotics Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received February 13, 2012; final manuscript received September 28, 2012; published online December 7, 2012. Assoc. Editor: Craig Lusk.

[1]This paper is an extension to the article published in the proceedings of the ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2011 [2]. The literature study has been expanded from three to five years which revealed an additional 10 articles, thus adding 200 references to be included in the data processing. Furthermore, structured searches in seven relevant journals have been added to the literature study to identify mechatronic challenges. As a result, additional researchers and solutions have been identified and included.

addressed. The case study in Sec. 5 is utilized to present real-world mechatronic design scenarios, and to argue about how well they are supported through current solutions. The paper concludes by a discussion in Sec. 6 and a conclusion in Sec. 7.

2 Method of Investigation

The objective of this paper is to identify the mechatronic challenges, assess their solutions, and then illustrate those challenges and solutions through a case study. In order to accomplish this, a literature study is carried out incorporating contributions from two sources. The first source consists of the research work carried out by researchers from the ASME IDETC/CIE mechatronics community. The second source is based on research work published in mechatronics-related journals plus the collective knowledge of the authors of this paper about important contributions within the research of mechatronic design. In addition to this, a workshop was set up to assess the completeness of the pool of identified researchers. For the first source, a filter function is needed to sort the vast amount of contributions in which mechatronic challenges are described. The filter function is based on the idea of extracting a large number of references from mechatronics-related articles from the ASME conference, and then selecting the most cited researchers. The proposed solutions to the stated challenges are obtained from sources 1 and 2, and from the knowledge of the authors regarding solutions available from the literature. A case study is used to illustrate the findings in terms of challenges and their solutions.

3 Literature Study

The goal of the first part of the literature study regarding the ASME community is to find the most reported and described challenges. This will be explained in detail in Secs. 3.1 and 3.2.

3.1 The Procedure for Gathering the Data. The five most recent ASME IDETC/CIE conferences are selected for the search (2007–2011). The process of finding the significant literature is based on identifying researchers who have published mechatronics-related articles, and those cited by the mechatronics community. The aim is therefore to find mechatronics-related articles and subsequently extract the references to see who are referenced the most in the community as a proof of relevance. First, articles dealing with the mechatronic design process have to be identified. This is done by using the keyword "mechatronics." If it is ambiguous whether or not the article would describe issues related to the mechatronic design process, the article is read to clarify the content. From the resulting 30 articles, 708 references are extracted.

3.2 Data Analysis. The 708 references extracted from the ASME conferences are analyzed by word-count software to reveal the names that appear the most. This quantitative evaluation is backed up by a qualitative scrutinizing of the reasons why the researchers are ranked as they are. Since it is common that authors cite their own previous work, a precondition is made that an author cannot appear more than once in the reference list of an article. The result of this evaluation is presented below in terms of a name and a numbered code. An example is "Wood (7/17/0)." The first number shows the number of different articles (out of 30 possible articles) in which the researcher has been cited. The second number shows how many times the researcher has been cited in total. The third number represents how many articles the researcher has published in the investigated conference proceedings (among the 30 investigated papers). In the given example, Wood has been cited 17 times in 7 different articles and did not write any of the articles in which he was cited. The obtained list from the analysis is as follows: Pahl (16/17/0), Beitz (16/17/0), Tomiyama (11/23/4), Gausemeier (9/26/3), Wood (7/17/0), Frank (6/13/2), McAdams (6/19/2), Paredis (6/11/3), Stone (6/28/2), Ulrich (6/9/0), Umeda (6/10/0), Yoshioka (6/7/0), Albers (5/11/2), Cabrera (5/7/3), Eppinger (5/9/0), Fenves (5/7/0), Hirtz (5/5/0), Pook (5/6/0), Schmidt (5/16/3), Steffen (5/7/0), and Suh (5/6/0). Researchers who are cited in less than four articles are omitted from the list, since it is assumed that the list of researchers presented above will cover the needed challenges. Furthermore, the number of researchers considered in the analysis has to be kept to a manageable level.

The presented search algorithm has limitations. It does not take into account if close colleagues are citing each other, or if the researcher is cited because his/her work is claimed not to be "sufficiently good" by others. Even though the impact of the research might not be directly reflected by the number of citations, the identified researchers are considered to have contributed significantly to the mechatronic community. Therefore their formulation and insight into the challenges faced by the design teams when developing mechatronic products are of importance to this study.

When investigating the researchers and their coauthors, certain research groups appear due to preferred research partners. In the following, the researchers from the list presented above are listed with their preferred research partners.

- Pahl group: Pahl and Beitz.
- Tomiyama group: Tomiyama, Umeda, Yoshioka, Cabrera.
- Gausemeier group: Gausemeier, Frank, Pook, Schmidt, Steffen.
- Wood group: Wood, McAdams, Stone, Hirtz.
- Ulrich group: Ulrich and Eppinger.

It can be assumed that researchers within a group tend to have similar views on mechatronic challenges. Hence, the grouping simplifies the data analysis. The group can consist of more members than stated above since the stated names are only from the ranked list.

3.3 Researchers Added From the Second Source. It is beneficial to extend the systematically generated list to other researchers, who, to our knowledge, have relevant insights into mechatronic challenges. These researchers come from the second source, where the articles are obtained by performing a general search for publications related to the design of mechatronics. This search targets relevant journals, the ICED conferences facilitated by the Design Society, along with the authors' knowledge about mechatronics research groups at an international level. The selected journals were: Journal of Mechanical Design, Research in Engineering Design, Systems Engineering, CIRP Annals—Manufacturing Engineering, Elsevier Mechatronics, IEEE/ASME Transactions on Mechatronics and Journal of Engineering Design. The reason for selecting these journals is that most research within *mechatronic design* methodology has been published in one of these venues. However, a number of other journals also publish mechatronic-related articles. Some examples are Journal of Computing and Information Science in Engineering, Engineering with Computers, Advanced Engineering Informatics, Robotica, and Artificial Intelligence in Engineering Design. Due to the multidisciplinary nature of mechatronics, research publications may end up in computer science, engineering, and design venues, and it was not intended for us to cover all publication venues as it would be unfeasible in a reasonable frame of time for a journal paper. This decision may introduce a small bias when preferring certain journals over others; however, we compensate by a deeper review of the considered papers in the corresponding journals. The authors took caution while selecting journals and the selected list is based on discussions with active researchers in mechatronic design at different international research groups.

The selected journals are searched with the keywords "mechatronic" and "design". Title, abstract and body text are all searched with the keywords. 135 (85 from the journals) relevant articles from these sources are analyzed and shortlisted based on their significance and relevance toward the design of mechatronic products. This provides a list of 46 articles. It is noted that about half of these articles are written by researchers already identified in the ASME list. Those researchers who were neither cited in the articles from source 1 nor published in ASME IDETC/CIE proceedings in the last 5 years include Buur [3], Salminen [4], Andreasen [5], and Adamsson [6]. The primary commonality among these researchers is their focus on the conceptual phase of the development life cycle, along with their emphasis on promoting collaboration between designers during the design activity.

The new extended list of researchers compiled from sources 1 and 2 was then discussed in a workshop with researchers belonging to the "Section of Engineering Design and Product Development" at the Technical University of Denmark, and the joined list was judged to be comprehensive.

3.4 Challenges Identified. It is now possible to extract the statements regarding challenges in mechatronic design. For this purpose, between three and five articles from each researcher or each research group are investigated. In Table 1, only one out of the three to five articles is referenced for the researcher/research group. Based on the extracted statements from each researcher, an affinity diagram method [7] is applied by which clustering of statements can be performed. A headline for each cluster is then formulated which should represent the statements clustered in it. In Table 1, these headlines for the challenges are listed. The highlighted rows in the table are used for illustration of points discussed in Sec. 4 and, thus will not be discussed in this section. In Table 1, the link to the researchers whose work complies with the stated challenges is marked with an "X." The stated challenges are causally linked. As an example, the "lack of common methodology" leads to a "lack of a common representation of a product concept." However, the causal chains will not be discussed further in this paper.

Table 1 cannot be assessed quantitatively since the pool of data, being the number of researchers investigated, is relatively small and because the filtering process may have distorted the picture of how many times a specific challenge is mentioned. The distortion occurs because it was chosen to group some of the researchers, which affected the number of times the challenges appeared.

In Table 1, some of the researchers identified in the data-processing of source 1 are left out. These are Pahl and Beitz, Ulrich (and Eppinger) and Suh. Their work is often cited due to their fundamental contribution to design theory, and even though they address mechatronic or complexity issues in their work, a large part of the mechatronic-specific challenges in Table 1 are unaddressed. Even though the Tomiyama group and the Wood group also have fundamental contributions to design theory, they are not omitted in Table 1 because they have contributions on mechatronic-specific challenges.

What have we gained by introducing the researchers from source 2? The conclusion can be drawn in two parts. First, the fact that a large part of the challenges is also stated by the researchers from source 2 supports the claim that the challenges stated by source 1 are truly generic and thus important to direct attention to in research. Second, we found that there are researchers from source 2 who contribute with challenges which are not reported by researchers from source 1. These researchers are Adamsson, Buur, Andreasen, and Salminen. Even though these reported challenges are not validated to the same extent as those described by both sources, they add to the understanding of the multidimensional challenges experienced by design teams developing mechatronic products.

The most commonly reported sets of challenges are primarily related to the way a product concept can be described and how information linked to the product concept can be shared across engineering disciplines. The commonly observed challenges are (the highlighted rows in Table 1 are not linked to this list): "A Lack of common understanding of the overall system," "A lack of a common language to represent a concept," "Different mental models of the system, the task and design-related phenomena," and "A lack of a common language to discuss freely." As stated by many of the researchers, the fundamental reason leading to the many challenges is the absence of a common mechatronic design methodology. This is again rooted in the fact that theories building upon different axioms cannot be joined to a common theory, as described by Tomiyama [1].

4 Solutions Proposed

This section will present a number of solutions to the mechatronic challenges, which are compiled through the literature study presented in Sec. 3. When there is sufficient documented evidence that a certain proposal addresses one or several challenges in the design of mechatronics, we consider it a solution. The solutions are listed in Table 2. The table shows the challenges which a given solution aims to support. The primary focus of a solution is represented by a black cell. The challenges are marked as either a "Y," a "P," or with no marking at all. A Y marks that a challenge is sufficiently addressed by a solution in the sense that it is possible to overcome the challenge by applying the proposed solution. A P indicates that the solution could aid in handling a given challenge, but does not fully address it. If a solution does not address a challenge at all, neither a P nor a Y is marked. The process of allocating the Y and the P was carried out by the authors of this paper (who are active researchers within the area of mechatronics). The selected articles were searched for documented examples, which would show the effects of applying a particular solution. If sufficient data are found within the searched articles (that a particular challenge is fully addressed by applying the proposed solution), a Y is marked. If an article provides sound arguments on only the presumed effects and benefits of applying the proposed solution, then the solution only qualifies as "partially addressed (P)." If an article provides no argumentation about handling a challenge, the solution does not qualify for either a Y or a P. A general overview of Table 2 shows that the mechatronics challenges are not sufficiently addressed by the proposed solutions. Specifically, solutions for challenges B, C, F, G, K, M, N, O, P, R, and S are either partially defined, or no solution is proposed. Among these challenges, there are challenges that relate to competences (K, M), to activities (G), and to organizational level (N, O). Although these are important challenges, they are not treated further in this paper. We restrict ourselves to focusing on product-related challenges. Challenges B, C, and F, and challenges P, R, and S are strongly connected to each other for the following reasons: Since there is a lack of a broadly accepted methodology (P) in mechatronics, a common language to represent the concepts can be difficult to accomplish. This creates a problem of finding the most suitable design through efforts across different domains. Along with difficulty in assessing consequences, the lack of methodology and the lack of a common language contribute to a higher complexity (S) in mechatronics. In addition to that, the lack of common language and inadequate information transfer between domains are strongly connected to challenge R (system engineer lacking detailed information on the system). Therefore, to gain a detailed insight on some of the core challenges in mechatronics, we will restrict ourselves to challenges B, C and F, which we believe are at the heart of mechatronic challenges. B, C, and F are marked in gray in Table 2. The other challenges are also important, but not treated further to limit the scope of this paper.

In the following, each solution is discussed and assessed about how well it supports challenges B, C and F.

(1) The first solution from Table 2 is about methods based on functional thinking. Buur [3], Nagel [9], van beek Tomiyama [18], and Suh [19] are a few examples of functional approaches. Nagel et al. [20] extended the functional approach by defining signal morphology and signal syntax to aid in assembly of functional models. The C&C-A

Table 1 Matrix relating mechatronic challenges to researchers stating them

Type	#	Challenges	Source 1: Tomiyama[a] [1]	Gausemeier[a] [8]	Wood[a,b] [9]	Paredis[a,c] [10]	Albers [11]	Cabrera [12]	Fenves [13]	Source 2: Adamsson [6]	Buur [3]	Salminen [4]	Andreasen [5]	Lindemann [14]	Browning [15]	Shea [16]	Bradley [17]
Product	A	Lack of a common understanding of the overall system design	X	X		X	X			X		X		X	X	X	X
	B	Difficulty in assessing the consequences of choosing between two alternatives	X	X		X	X				X	X			X		X
	C	Lack of a common language to represent a concept	X	X	X	X	X	X		X	X	X	X			X	X
	D	Modeling and controlling multiple relations in the product concept	X	X		X	X			X				X	X		
	E	Being in control of the multiple functional states of the product		X	X		X						X				
	F	Transfer of models and information between domains (expert groups)		X	X	X	X	X	X							X	X
Activity	G	Synchronizing development activities to attain concurrent engineering		X	X			X	X				X		X		
Mindset	H	Different tradition within the domains for how to conduct creative sessions										X					
	I	Reluctant to interact with engineers from other disciplines										X					
	J	Different mental models of the system, task and design-related phenomena	X	X		X		X		X	X	X	X		X		X
Competence	K	Lack of common language to discuss freely at creative meetings	X	X				X		X	X	X	X			X	X
	L	Education within disciplines do not call for integration in professional life									X				X		X
	M	The nature of design is different	X	X						X	X		X	X			
Organizational aspects	N	Product complexity affects the organization complexity	X	X											X		
	O	Knowledge transfer between domains is inadequate (even in cross-disciplinary teams)	X	X				X	X			X					X
Other aspects	P	Lack of a broadly accepted methodology	X	X	X	X					X	X	X	X			
	Q	Mechatronic ownership is lacking								X		X	X				
	R	System engineers are lacking detailed information of the system										X					
	S	Complexity as a generic problem	X	X	X	X	X		X	X				X	X		X

Researchers/Research Groups: Source 1 = Tomiyama through Fenves; Source 2 = Adamsson through Bradley.

[a]Research groups.
[b]Nagel is part of the Wood group, hence [9].
[c]Shah is part of the Paredis group, hence [10].

approach from Albers et al. [11] attempts to help designers understand and communicate the complex dependencies between function and form, and create system architecture through function and part database. Hehenberger et al. [21] described the hierarchical decomposition based on function models for mechatronic systems. Functional modeling is abstract in terms of the level at which the description of the product concept is performed. Therefore it can serve the purpose of a common modeling language (C) to an abstract level only. It is typically after the functional modeling that the development process becomes domain-specific. Functional thinking is only part of the complete picture of the design activities, and other factors, such as how system elements contribute to system properties, are not well supported by it (B). When performing model transformations (F), the focus is toward the means and not the functions, thus limiting the value of functional approaches.

(2) The second solution is about modeling relationships between elements from different mechatronic domains. Design structure matrix (DSM) and domain mapping

matrix (DMM) by Kreimeyer et al. [14,22], and Danilovic and Browning [15] are examples of modeling relationships between functions, components, physical structure, and resources during a mechatronic product development process. Bonnema [24] proposes a FunKey (function key drivers) architecting approach to model relationships between functions and key drivers of a product, with an aim to provide good insight to different stakeholders while designing. The main aim of these approaches is to support the managing of multiple relations and dependencies during design. However, analyzing the consequences of selecting between different alternatives becomes too cumbersome through these approaches in terms of effort and efficiency (B). Moreover, they are meant for establishing relations across different domains, and not for information transfer between design models (F). They are partially suitable for aiding a common language (C), since engineers from different domains can discuss dependencies based upon them.

(3) The third solution is about controlling integration between domains via requirements. Systems Engineering [25] and work by Woestenenk et al. [26] are examples of such solutions. However, requirements cannot be utilized for accessing consequences (B) of different design alternatives for a mechatronic system. Therefore, model-based system engineering [37] proposes the use of requirements management tools in addition to system-level modeling to control system design based on requirements (common modeling language (C)). This provides a better utilization of requirements through computer support. A model transformation between system-level models and domain-specific models (F) is, however, required to keep the design models consistent with each other.

(4) Different process models, specifying the activities to be performed during the design process are proposed by several researchers. These process models are usually an extension of a process model in one domain toward covering several domains. VDI2206 [28], Systems Engineering [25], and work by Isermann [27] and Salminen [4] are examples of such models. These models aim at synchronizing the workflow and activities, which the design team must perform during the development. However, these approaches state that dependencies should be handled, not how to actually manage them in relation to assessing consequences (B). System-level design plays an important role during the design of mechatronic systems, especially to support complexity management. Therefore, most process models urge for the creation of an architectural description of the system through a system-level design language. Different modeling languages can be used based on the product area. For instance UML has been popular for software design and SysML for systems engineering. Other examples are A3 architecture overviews and SFSL. Using languages such as SysML, a system-level description or an architectural description of the system can be made. SysML also allows the modeler to define product variants and competing concepts, so that they can be analyzed to choose the best candidate architecture. However, process models themselves do not solve the common language challenge (C). The same applies to model transformations (F), which can be made a part of the process models, but process models themselves do not aim at solving challenge F.

(5) In the aim of a common language and to solve the communication problems during the conceptual design phase, different solutions are proposed (solutions 5 and 6). The A3 architecture overviews [29] are an example of an informal description, which provides an overview of the complete system in terms of different system aspects, such as functional and physical. Representing system design concepts informally is useful for discussions among different domain experts. However, this does not address assessing consequences for each design domain while choosing different system alternatives (B). Moreover, such overviews have the same potential of becoming a common language as the functional thinking proposed by Buur [3]. Hence, there are other abstraction levels in design that cannot be supported by A3 overviews (C). Therefore, it can be said that representing concepts in A3 overviews can lead to gaps between domain-specific design activities and system-level design activities, and it is clear that model transformations cannot be utilized with A3 overviews to reduce this gap (F).

(6) An attempt toward a language more specifically related to mechatronics is the semiformal specification language (SFSL) by Gausemeier et al. [8], which aims at specifying a mechatronic concept in terms of a number of aspects, such as a behavior-aspect and a structural-aspect. Modeling languages that describe the system in terms of different views are also proposed such as SysML [30] and AM tool [31]. The opinions from researchers behind these modeling approaches contain a contradiction, especially in terms of their usefulness and effectiveness. For example Borches and Bonnema [38] document that formal modeling such as SysML does not necessarily resolve the communication problem between people from different design domains, nor does it produce models that are easy to understand. The fragmentation of proposals for a common modeling language by different groups of researchers indicates a need for further improvement in this area. Therefore it can be said that although a common design language is needed, the nature of such a language in terms of being formal or informal is still unknown, and there is still a need for developing support in this area.

(7) Model transformations are proposed as a possible solution to relate two design models. Shah et al. [10] show how a mapping between two design models can be used to build transformations between them. An example is the transformation between SysML and Modelica, which combines the descriptive capability of SysML with the analysis and simulation capability of Modelica. Formal models utilized during the conceptual design phase have advantages of supporting automated-model transformations to other design models. However, dependencies between mechatronic domains cannot be directly managed through model transformations as this requires explicit models of dependencies. The dependencies are usually only implicitly known. Hence, it is not always possible that a model contains a representation of all possible dependencies that arise while accessing consequences of different alternative design solutions (B). Moreover, model transformations (F) can be more effective if a proposal for a common design language (C) in mechatronics becomes successful. However, this is not an explicit goal of the model transformation community to develop such a language.

(8) Besides intradomain interfaces, interfaces can also be observed between domains, such as a shielding of an electronic sensor. An international standard exists (ISO/IEC 81346) that specifies how to define a physical interface. Furthermore, clearly defined interfaces are stated as being advantageous [25]. The interface description aims at specifying the physical interfaces based on a functional partitioning between the domains. Therefore, interface handling can only provide some of the information needed for assessing consequences. Hence, B is not covered. Clearly stated interfaces cannot be used as a common language (C), even though they can be used as a framework for discussions. Model transformation (F) is decoupled from interface specification, and is therefore not covered.

Table 2 Solutions proposed in literature against the challenges identified in Sec. 3.4. A black cell indicates the primary aim of the solution, and the gray columns show the most important challenges.

#	Solutions	A – "Lack of common understanding"	B – "Difficulty in assessing consequences"	C – "Lack of common language (product)"	D – "Modeling/controlling multiple relations "	E – "Controlling multiple functional states"	F – "Transfer of models"	G – "Synchronizing development activities"	H – "Conducting creative sessions"	I – "Reluctance to interact"	J – "Different mental models"	K – "Lack of common language	L – "Education within disciplines"	M – "Nature of design is different"	N – "Product/company complexity"	O – "Knowledge transfer between domains"	P – "Lack of common methodology"	Q – "Mechatronic ownership is lacking"	R – "Systems engineers lacking information"	S – "Complexity as generic problem"
1	Activities based on functional approaches and functional decomposition (Buur [3], Wood [9], Tomiyama[18], Suh [19]), applying functions, means patterns [20], C&C-A [11], state and event relations, hierarchical approach [21]	**Y**		P	Y	Y					P		Y							P
2	Relationship management e.g. DSM and DMM [22], [15], QFD [23], FunKey [24]		P		**Y**			P			P					P				**P**
3	Controlling design activities through requirements management (Systems engineering [25], Tomiyama [26])			**P**			P				P	P		P						
4	A process model containing activities for the development process. (Isermann [27], VDI2206 [28], Salminen [4], Systems Engineering [25])							**P**						P	P	P				P
5	Informal description consisting of a number of modeled/described aspects to specify systems, A3 overviews [29], Salminen [4], Buur [3]	Y	P	**P**	P						Y		P			P				P
6	Modeling languages to describe system as a whole, formally or semi-formally. SysML [30], SFSL by Gausemeier [8], AM tool [31]	Y	P	**P**	P						P									
7	Model transformation from a design model in one domain into a design model in another domain (Gausemeier [32], Paredis [10])						**P**					P		P		P				P
8	Formalized specification of interfaces. (ISO/IEC 81346 [33], Systems engineering [25])				**P**					P	P			P	P					P
9	Simulation of phenomena that incorporate elements from the different domains (e.g. Modelica [34], Bond Graphs [35], and integrated simulation [36])		**P**		P	P	P													
10	Setting up a systems integration group in the project (Adamsson [6], Andreasen [5])	Y			**P**			P	Y	Y	P	P				P		**Y**		P

(9) Computer aided modeling and simulation provides advantages of building and executing multidomain design models, in order to predict the product properties. Modelica is one example of a multidomain modeling and simulation language. Bond graph based approaches are also proposed such as Wu et al. [35], where topology of a design concept is captured through a function structure, and a system dynamics model is created though a CD graph**.** Another approach is an optimization process where domain-specific models are executed concurrently to perform a multidomain optimization on product properties. Albers et al. [36] described integration of structural and controller design models in such an optimization process. Although such approaches provide support for assessing consequences (B) to an extent, they cannot be treated as a common design language (C) for all domain experts. Moreover, they are only good for design modeling when the basic principles and the basic structure of the product have been determined. Current efforts within the Modelica community aim to standardize model transformations (F) between SysML and Modelica. However, this will only be useful if SysML is utilized.

(10) Adamsson [6] and Andreasen [5] proposed setting up a systems integration group. This group is primarily responsible for facilitating the information flow, and the collaboration between engineers from the different domains to increase performance of the overall system. However, challenges B, C, and F are only supported indirectly by anticipating that an integration group will facilitate closer integration between the domains.

5 Case Study

The purpose of presenting a case study in this paper is to illustrate the three selected mechatronic challenges (B, C, and F)

Fig. 1 (*a*) Instrument, (*b*) watch, (*c*) temperature unit, (*d*) heart-rate sensor, and (*e*) charger for instrument

highlighted in Sec. 4. This will allow us to relate the rather abstractly described challenges to a very concrete situation in a product development process. Additionally, the product case will help in assessing how well the proposed solutions would have helped the design team in their design task. Therefore, the case study is not used for verifying whether or not a challenge can be handled by the proposed solutions. Instead, it is used to bring in a real-world dimension, and create a context surrounding the challenges.

The aim of the project chosen as the case study was to develop a watch system based on a mechanical watch and an instrument, which can be attached to the watch (see Fig. 1). The instrument contains advanced functions used for alpine skiing. An additional two external units wirelessly transmit heart-rate information and temperature information to the instrument. In this case study, we focus on the external temperature unit showed in Fig. 2. It is noteworthy that one of the authors was involved as a development engineer in this specific project.

The case study was built up on the experience gained by participating in the development team. This was backed up by document-analysis and interviews with the project managers for the mechanical, electronics and software development. Due to the limitations of describing the development process as a whole, we deem it necessary to only select small fragments from the design process to illustrate the selected challenges. In the following, three scenarios from the case study are presented, which are directly related to what we consider as the most important challenges (B, C, and F). This is followed by a discussion on possible solutions from Table 2, and a conclusion on using those solutions to mitigate a particular challenge.

5.1 Assessing Consequences (Challenge B)

5.1.1 The Power Consumption Scenario. In the beginning of the project, it was assessed that the power consumption would be one of the key drivers for the project. The Radio Frequency (RF) chip for wireless transmission and a running processor are the primary sources for the power drainage. The main electronic components are illustrated in Fig. 3. Two basic approaches can be chosen: either to minimize the power consumption (thereby the user should change batteries), or to make the whole unit rechargeable. Within the scope of minimizing the power consumption, two main directions can be chosen: to cut the power manually or automatically when it is not in use, or to minimize the power usage by features in the electronics, and by clever programming. Solutions are spread over all the domains. Some solutions have a direct effect on the use pattern, hence the user experience. Some solutions require further technology clarifications. Other solutions require the consequences on the products' life phases (e.g., change of battery) to be assessed. The main challenge is that there are many conceptually different ways of solving the power issue, but how can we, in the best possible way, reason about the consequences of selecting one product concept above another? The problem of assessing the consequences when choosing between concepts is a general concern in product development. Yet, this concern increases when different domains are involved in the design process while investigating alternative design concepts.

5.1.2 Discussion on Solutions. In Table 2, four solutions have been identified, which potentially should embrace the challenge of assessing the consequences by selecting between two product concepts: a) Relationship management; b) Informal descriptions; c) Formal language description; d) Mechatronic concept description and simulation of phenomena. DSM, MDM, QFD as well as modeling languages such as SysML and the various simulation programs only provide a description of a single or few closely related properties or aspects. In the case study, a holistic approach is needed to consider the consequences of a product concept, which the mentioned mechatronic solutions cannot encompass. In the project, various concepts were sketched to reveal their potentials and drawbacks and to evaluate the life phases. The product concepts were then discussed in several meetings and the progression of reducing the needed power was continuously assessed.

Modeling languages exist ranging from the formal modeling languages such as SysML (including supporting integration frameworks, e.g., Refs. [39,40]) and AM Tool, over semiformal modeling languages such as Gausemeier's SFSL, to less restricted modeling such as the A3 overviews. Even the A3 overviews, proposed as an informal method, are not sufficient, since they do not address mechatronic-specific aspects such as the implications of different allocation of functions to the domains. An informal description different from the A3 overviews, seems to be the best way to mitigate the challenge since an informal description is flexible. The question is, however, is the informal description so flexible that it does not provide any mechatronic-specific support? The answer seems to be yes. In the presented case, the proposed solutions seem even less appropriate compared to the evaluation in Table 2.

5.2 Common Language to Represent a Concept (Challenge C)

5.2.1 "The Custom Made Gasket" Scenario. A request for changing the outer shape to make the unit appear lighter causes a change in the mechanical design (Fig. 4). The changed design makes less space for fitting the main gasket, which ensures the water resistant property of the unit. Instead of the previously used standard *O- ring*, a custom shaped gasket must be used unless the outline of the printed circuit board (PCB) is changed (Fig. 4). At this late stage of the electronic development, a change to the PCB

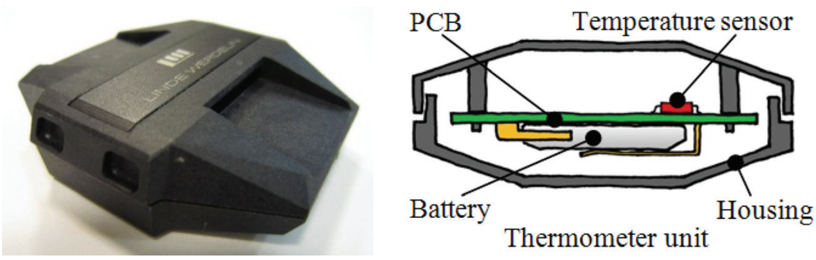


Fig. 2 Temperature unit

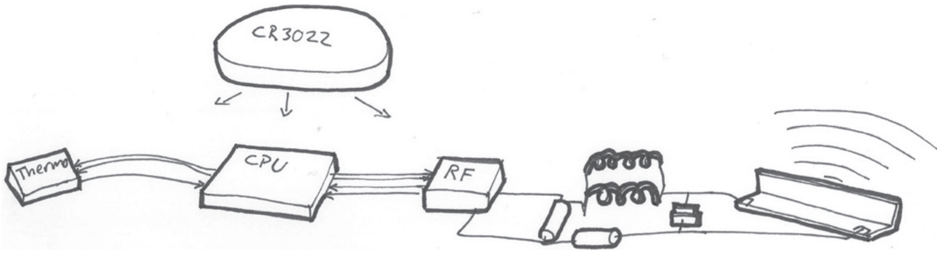


Fig. 3 Main electronic components

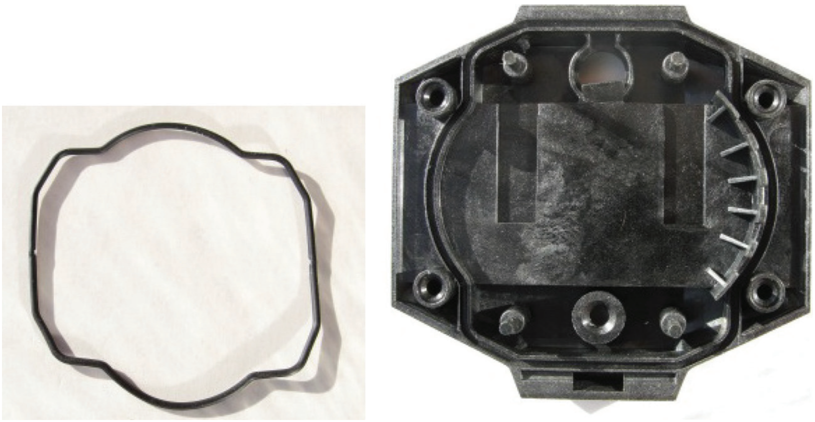


Fig. 4 The custom made gasket and the part in which it has to be inserted

would result in reorganizing the electronic components. In a High Frequency (HF) circuit, the relative placement of components affects the transmission quality, thereby increasing development cost, risk and time if the PCB layout were to be redesigned. Therefore a custom made gasket is evaluated as "the best fit" solution.

5.2.2 Discussion on Solutions. The situation as described above is a known characteristic of the design of mechatronics, where the best alternative among a few has to be chosen, such as changing the gasket or changing the PCB. However, there are consequences attached to each alternative for different design domains—for instance, the redesign cost of the PCB, the redesign cost of the gasket and the mechanical module, the packaging of the high-frequency electronics, and the success probability of the integration test. The dependencies between different domains during the design activity are major contributors toward these consequences, such as the relationship between the gasket and the size of the PCB. Moreover, the best solution has to be considered in terms of the overall system, and not just between domains. Considering Table 2, the common modeling language proposals such as SysML, SFSL, and A3 overviews can be considered to build a system view. The system view enables modeling and evaluation of alternatives in terms of the system as a whole. Moreover, DSM/DMM, and FunKey architecting are also proposals to identify relations between functions and user demands, and between functions and components. However, DSM/DMM and FunKey architecting serves the purpose of relationship management only. Building a holistic system view along with assessing certain characteristics of the system such as performance or cost is not supported. From Table 2, activities based on functional thinking, and controlling design through requirements are also proposed as solutions for a common language to describe the concept. However, functional thinking is proposed to describe only the functional view of the product, thereby leaving out the structural view which is essential to the gasket issue. In the case of requirements, they can be used for goal specifications (of the product to be), or result specifications (documenting the finalized product). However, the requirements cannot be used to represent a design concept.

Considering SysML, SFSL, AM tool, and A3 overviews, these languages provide different solutions toward representing the size constraint relation between the gasket and the PCB. This constraint modeling enables mechanical and electrical/electronic engineers to understand the effects of gasket size on the PCB. It also relates this constraint to the system model. However, the decision for whether to redesign the gasket, or redesign the PCB requires assessing the consequences of each alternative in relation to designer preferences. Such a decision requires models of dependencies which the current semantics of SysML, SFSL, or AM tool do not explicitly support. We believe that an informal and visual language, where designers from different domains can sketch their ideas to each other, and highlight the relationship of their concepts to each other, is a more effective way of managing dependencies such as between gasket and PCB. The sketching can be partially or fully supported by a calculation or a simulation engine (depending upon how open/restrictive the visualization is). References [41,42] are two examples of a sketch-based interface. A3-overviews target only a medium where the dependencies can be expressed; it does not address how the dependencies can be understood and managed. Formal models (such as a SysML model) are not necessarily useful in the conceptual design phase. One of the reasons for decreased usability is the rate at which models change. Another is the decreased effectiveness in communication caused by a lack of visual representation of the product structure [38].

5.3 Transfer of Models (Challenge F)

5.3.1 "The ESD Protection Issue" Scenario. Due to a requirement for better temperature sensing, a change of design is necessary. Discussing the proposed solution with the electronic engineers, it becomes apparent that the solution is prone to electrostatic discharges. Mitigations have to be made for the

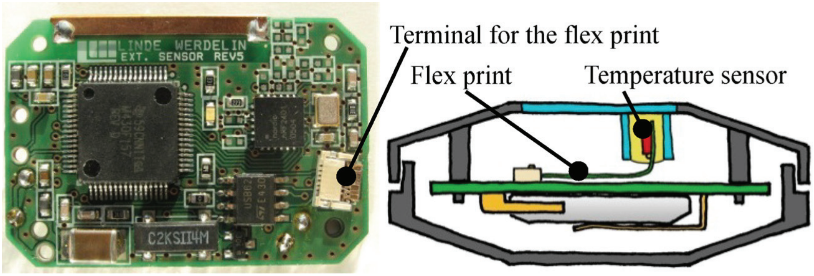


Fig. 5 The PCB and the positioning of the flex print and the flex print terminal

electronics not to be damaged in such a case. The proposed design is shown in Fig. 5.

For easier handling of the small thermo sensor, it is placed on a flex print which can be easily connected to the PCB instead of using five ordinary wires. Due to the stiffness of the flex print, the location and orientation of the terminal is important. The decision about this fitting is made by the electronic and mechanical engineers collaboratively. Figure 5 shows the PCB connector placement and the position of the flex print.

5.3.2 Discussion on Solutions. In this particular case, the orientation and location of the terminal on the PCB, and the placement of the connector on the flex print show a clear dependency between electronic and mechanical models. In order to reach a solution, both electronic and mechanical engineers had to have several discussions during a number of design iterations. From Table 2, three solutions have been identified which should aid in overcoming this challenge related to information transfer across domains: (a) Controlling the design through requirements management; (b) simulation of phenomena incorporating model elements from different domains; (c) integration of models through model transformations.

Requirement specifications play a key role in controlling the design, and hence it is proposed to utilize these specifications as a solution to ease the information transfer between domains. Traditionally, a specification has to direct the search for solutions. What is required (here) for information transfer are the details of the necessary parameters of a concept from each domain, and not the specification that directs a search for those concepts. Simulation of properties is also proposed as a possible solution to ease the information transfer between domains. However, in the above case, the mechanical and electronic engineers need information regarding the location of the terminal.

To satisfy the design constraints in both mechanical and electronic domains during an optimization run, integration supporting information transfer between design models in electronics and mechanics is required. Current tool support lacks such integration between tools [10]. Therefore, although simulation in the sense of algorithmic optimization can be built, the efforts and resources required to create it may prove too costly for an organization compared to a manual optimization performed by the involved engineers. Integration of models through a model transformation, such as Refs. [10,32], is proposed as a solution to aid in information transfer between domains. The location of the connector in the mechanical design model can be extracted and represented through a transformed model. In this case, it would be an electronic design model to support the electronic engineer during the design process. In the following paragraphs, we look more closely at the relationships between models to be able to evaluate challenge F in terms of performing a model transformation.

Different design models are related in terms of the system properties which they affect. Although two design models may both affect one system property, there is only a portion of each model that has substantial meaning in the other model. Tomiyama et al. [43] explain that two models can only be integrated with each other if the background theories (that these models are based on) are compatible. The compatibility between two background theories suggests that a concept in one theory can be related to a concept in another theory. For example, inertia has no meaning in electronic PCB design, but it has a meaning in controller design. If two background theories are compatible, then a model transformation can be applied to the corresponding models. Model transformation approaches provide a capacity to control which part of the source model is read and what is created in the target model by specifying metamodels and the transformation between them. Therefore, we conclude that model transformation has a potential in addressing challenge F. In the following, model transformation approaches will be discussed further, followed by concluding remarks on the limitations of a model transformation.

One approach for integration of models is to utilize a central product model where all the information is stored. The central product model can be utilized to understand and manage the relationships between different aspect models. The aspect models can also be generated from the central product model. Another approach for managing relationships between models is where an integration at the level of background theories is proposed to support integration between so-called "multiple aspect models" [44]. The approach is based on developing different aspect models based on different background theories, e.g., dynamic models, and geometric models. These aspect models can be integrated through a central metamodel, where the relationships between the concepts of the different background theories are specified. By specifying the concept relationships between the different background theories through metamodels, it is possible to manage the influence of one aspect model onto another. A similar approach is presented in the PACT experiment in Ref. [45], where an approach for integration among multiple aspects (agents) during design is discussed.

It is likely that a transformed model does not contain all the information that is required by a modeler, because it is not always possible to know at the earliest stage which properties affect each other and hence, should be in the model. This information might be known at a later stage. Therefore, if such properties are not explicitly supported by the metamodel of a domain, then a model transformation will not be useful straight away, and will require further efforts. Hence, challenge F is not fully addressed through model transformations.

In order to support the design process for mechatronic products, we propose model integration between domain-specific views such as a mechanical view and a system view built through a common system modeling language. This will provide an opportunity to find the best mechatronic design solution for a system. References [10,32] are examples of steps in this direction. However, since the nature of common modeling language is still an unknown, this area has good potential for further development.

6 Discussion

Most papers about mechatronic design end by stating that a common methodology and a common conceptual model is needed. This statement has been repeated for the last 20 yr. If it was possible, it would have been likely that such a method would have been found, or significant findings presented which would be a step toward it. Proposals of a mechatronic concept description always end up by constituting different needed views. Having "x" number of different views on a concept negates the idea of a common conceptual representation. In principle, this is not different from the acknowledgement that you need several different views of a system to be able to describe it, e.g., the proposal of the Domain Theory by Mogens Myrup Andreasen in the early 1980s [46][2]. Tomiyama [1] states that two theories cannot be joined if they are based on a different set of axioms. This is the reason why the so-called "common mechatronic concepts" always have to be presented by x number of views. For each type of property one has to model a separate view has to be created [48,49]. One or more of these views relate to function modeling, which is particularly interesting when trying to create a description spanning disciplines. Buur [3] states that function modeling can be used across the mechanical, electronic and software disciplines, which is further backed up by Tomiyama's statement on axioms [1]. This will enable methods that are based on function modeling to be used across the mechatronic domains. Some of the methods based on function modeling are: life phase thinking, process descriptions of the product, state-transitions, function/means tree, and QFD. Quite soon in the development process, one needs to model and evaluate properties of the design. Whether the property modeling is performed based on sketching and/or computer simulation, the problem of a common mechatronic model appears, because an evaluation of a property is linked to a certain theory which will be domain-specific. To assess several properties from different domains in one model, no adequate theory or tool or process has been proposed. We suggest the following thinking experiment: If two competing concepts are developed to finalized products, the consequences can be fully evaluated. Since this is seldom carried out for obvious reasons, it is necessary to show the relations and consequences by other means. We have previously described that a common conceptual model, which has details beyond describing functionality in the product, would violate the fundamental axioms. Therefore, we have to accept that not all the relations can be modeled, besides those few which can be described as the key relations. We should be willing to work with ill-defined problems across the domains and be willing to generate alternatives. Most of all, we have to be able to identify what information is relevant to share with developers from other domains. We should acknowledge the collaboration aspect of teamwork, and provide rooms (workshops) and methods, which will enable cross-domain discussions, and which will be graphically intriguing. While working with mechatronic issues the project team might direct all of their focus toward the technical mechatronic issues and thereby lose sight of the potential of collaboration methods and mindsets. In design-practice, these solutions (focusing on the collaboration issue) represent potential for obtaining better integration, and should be carefully considered along with other solutions to the mechatronic challenges.

7 Conclusion

In this paper, the challenges that seem to be most significant in the design of mechatronics are presented. They range from product-specific challenges to company-level challenges. The extended search for stated challenges and solutions revealed a larger number of scientific contributions within functional modeling approaches than originally revealed in the conference proceedings article [2]. Despite the extended search, the proposed solutions in the literature only provide partial solutions to those challenges. A large part of the identified solutions appear to support analysis activities rather than synthesis activities. As a product concept progresses, effort must be spent to continuously update the information that goes into the analysis-oriented solutions to be able to use them. This effort compared to what can be gained by using a particular solution is seldom assessed, evaluated or investigated in the literature. The solutions which are not analytical in nature are the ones based on functional reasoning, which have the capability of being applied across domains. Even though functional reasoning should be capable of supporting the design process through all the design phases, the suggested solutions only support the initial steps in the conceptual phase. It appears that introducing means to the functions gives rise to the product-related mechatronic challenges stated in Table 1. The reason is that a number of views are needed to model various properties of the product, which cannot be encapsulated by one methodology or one model. It is the need for considering these views concurrently that causes statements like "lack of overview of the system" or "difficult to assess consequences of choices."

A common design language would, as stated by many of the researchers in the study, facilitate a better collaboration between engineering disciplines. However, we do not believe that a common language based solely on functional modeling will be adequate in addressing the challenges. A common language, if possible to develop, would need to consist of x number of product views to be modeled, ruling out the prospect of a unified representation. Furthermore, a common language should be evaluated based on: its capability to represent the desired views effectively, its potential to be understood by engineers from the various domains, and its effect on the efficiency of the development process. If a common language can be realized, it would also facilitate in creating variations of the product concepts in the conceptual phase. The case study illustrated this as being beneficial to reveal the consequences of selecting between alternative design concepts.

Acknowledgment

This work is the result of the joint collaboration between KTH and DTU. The first and second authors have contributed equally in the development of this paper. We wish to thank Morten Linde and Jorn Werdelin, who are the cofounders of the watch company LindeWerdelin [50] for letting us use the development project as the case for this paper. We would also like to thank Troels Pedersen from IPU and Henrik Bendsen from Oticon for their help in adding detailed information to build the case.

References

[1] Tomiyama, T., Amelio, V. D., Urbanic, J., and ElMaraghy, W., 2007, "Complexity of Multi-Disciplinary Design," Ann. CIRP, **56**(1), pp. 185–188.
[2] Torry-Smith, J. M., Qamar, A., Achiche, S., Wikander, J., Mortensen, N. H., and During, C., 2011, "Mechatronic Design—Still a Considerable Challenge," ASME 2011 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, pp. 33–44.
[3] Buur, J., 1990, "A Theoretical Approach to Mechatronics Design," Ph.D. thesis, Technical University of Denmark, Denmark.
[4] Salminen, V., and Verho, A. J., 1989, "Multi-Disciplinary Design Problems in Mechatronics and Some Suggestions to Its Methodical Solution in Conceptual Design Phase," International Conference on Engineering Design (ICED89), Vol. 1, pp. 533–554.
[5] Andreasen, M. M., and McAloone, T. C., 2001, "'Joining Three Heads'—Experiences From Mechatronic Projects," 12th Design For X Symposium, pp. 151–156.
[6] Adamsson, N., 2004, "Model-Based Development of Mechatronic Systems—Reducing the Gap Between Competencies?," Tools Methods Competitive Eng., **1**(2), pp. 405–413.
[7] Buur, J., 1991, "Design Methods in Japan. Research Education and Industrial Application From a European Viewpoint," J. Eng. Des., **2**(2), pp. 91–103.
[8] Gausemeier, J., Frank, U., Donoth, J., and Kahl, S., 2009, "Specification Technique for the Description of Self-Optimizing Mechatronic Systems," Res. Eng. Des., **20**(4), pp. 201–223.
[9] Nagel, R. L., Stone, R. B., Hutcheson, R. S., McAdams, D. A., and Donndelinger, J. A., 2008, "Function Design Framework (FDF): Integrated Process and Function Modeling for Complex Systems," ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC/CIE 2008, pp. 273–286.
[10] Shah, A. A., Kerzhner, A. A., Schaefer, D., and Paredis, C. J. J., 2010, "Multi-View Modeling to Support Embedded Systems Engineering in SysML," Graph Transformations and Model Driven Engineering—Lecture Notes in Computer Science, Paper No. 5765/2010, pp. 580–601.

[2]The Domain Theory in Ref. [46] is also described in Ref. [47].

[11] Albers, A., Braun, A., Sadowski, E., Wynn, D. C., Wyatt, D. F., and Clarkson, P. J., 2011, "System Architecture Modeling in a Software Tool Based on the Contact and Channel Approach (C&C-A)," J. Mech. Des., **133**(10), p. 101006.

[12] Cabrera, A. A. A., Foeken, M. J., Tekin, O. A., Woestenenk, K., Erden, M. S., De Schutter, B., van Tooren, M. J. L., Babuška, R., van Houten, F. J. A. M., and Tomiyama, T., 2010, "Towards Automation of Control Software: A Review of Challenges in Mechatronic Design," Mechatronics, **20**(8), pp. 876–886.

[13] Fenves, S. J., 2001, "A Core Product Model for Representing Design Information," National Institute of Standards and Technology, Gaithersburg, MD, Technical Report No. NISTIR 6736.

[14] Kreimeyer, M., Braun, S., Gurtler, M., and Lindemann, U., 2008, "Relating Two Domains via a Third: An Approach to Overcome Ambiguous Attributions Using Multiple Domain Matrices," ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC/CIE 2008, pp. 297–306.

[15] Danilovic, M., and Browning, T. R., 2007, "Managing Complex Product Development Projects With Design Structure Matrices and Domain Mapping Matrices," Int. J. Proj. Manage., **25**(3), pp. 300–314.

[16] Wolkl, S., and Shea, K., 2009, "A Computational Product Model for Conceptual Design Using SysML," ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC/CIE 2009, pp. 635–645.

[17] Bradley, D., 2010, "Mechatronics—More Questions Than Answers," Mechatronics, **20**(8), pp. 827–841.

[18] van Beek, T. J., and Tomiyama, T., 2009, "Integrating Conventional System Views With Function-Behaviour-State Modelling," 19th CIRP Conference—Competitive Design, pp. 65–73.

[19] Suh, N. P., 2001, *Axiomatic Design: Advances and Applications*, Oxford University Press, New York.

[20] Nagel, R. L., Vucovich, J. P., Stone, R. B., and McAdams, D. A., 2008, "A Signal Grammar to Guide Functional Modeling of Electromechanical Products," J. Mech. Des., **130**(5), p. 051101.

[21] Hehenberger, P., Poltschak, F., Zeman, K., and Amrhein, W., 2010, "Hierarchical Design Models in the Mechatronic Product Development Process of Synchronous Machines," Mechatronics, **20**(8), pp. 864–875.

[22] Braun, S. C., and Lindemann, U., 2007, "A Multilayer Approach for Early Cost Estimation of Mechatronical Products," International Conference on Engineering Design (ICED07), pp. 187–188.

[23] Hauser, J. R., and Clausing, D., 1988, "The House of Quality," Harvard Business Review, pp. 11.

[24] Bonnema, G. M., 2011, "Insight, Innovation, and the Big Picture in System Design," Syst. Eng., **14**(3), pp. 223–238.

[25] Sage, A. P., and Rouse, W., 2009, *Handbook of Systems Engineering and Management*, John Wiley & Sons, Oxford.

[26] Woestenenk, K., Tragter, H., Bonnema, G. M., Cabrera, A. A. A., and Tomiyama, T., 2010, "Multi Domain Design: Integration and Reuse," ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2010), pp. 519–528.

[27] Isermann, R., 2005, *Mechatronic Systems Fundamentals*, Springer-Verlag, London.

[28] Association of German Engineers, 2004, *VDI 2206, Design Methodology for Mechatronic Systems, VDI Guidelines*, Beuth Verlag, Berlin.

[29] Borches, P. D., and Bonnema, G. M., 2010, "A3 Architecture Overviews—Focussing Architectural Knowledge to Support Evolution of Complex Systems," 20th Annual INCOSE Symposium (IS2010).

[30] Object Management Group, 2010, OMG Systems Modeling Language Specification V1.2, http://www.omg.org/spec/SysML/1.2/PDF/

[31] Cabrera, A. A. A., Woestenenk, K., and Tomiyama, T., 2011, "An Architecture Model to Support Cooperative Design for Mechatronic Products: A Control Design Case," Mechatronics, **21**(3), pp. 534–547.

[32] Gausemeier, J., Schafer, W., Greenyer, J., Kahl, S., Pook, S., and Rieke, J., 2009, "Management of Cross Domain Model Consistency During the Development of Advanced Mechatronic Systems," 17th International Conference on Engineering Design (ICED'09), pp. 1–12.

[33] 81346, 2012, ISO/IEC 81346 Standard, http://81346.com/english/

[34] Modelica Association, 2010, Modelica Language Specification Version 3.2, http://www.modelica.org/documents/ModelicaSpec32.pdf

[35] Wu, Z., Campbell, M. I., and Fernandez, B. R., 2008, "Bond Graph Based Automated Modeling for Computer-Aided Design of Dynamic Systems," J. Mech. Des., **130**(4), p. 041102.

[36] Albers, A., and Ottnad, J., 2010, "Integrated Structural and Controller Optimization in Dynamic Mechatronic Systems," J. Mech. Des., **132**(4), p. 041008.

[37] Friedenthal, S., Moore, A., and Steiner, R., 2008, *A Practical Guide to SysML, the Systems Modelling Language*, Morgan Kaufmann, Amsterdam.

[38] Borches, P. D., and Bonnema, G. M., 2010, "System Evolution Barriers and How to Overcome Them!," 8th Conference on Systems Engineering Research (CSER2010).

[39] Phoenix Integration, 2012, ModelCenter, http://www.phoenix-int.com/software/phx_modelcenter_10.php

[40] Comet Solutions, 2012, Comet Workspace, http://cometsolutions.com/products/workspace/

[41] Macomber, B., and Yang, M., 2011, "The Role of Sketch Finish and Style in User Responses to Early Stage Design Concepts," ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference IDETC/CIE 2011, pp. 567–576.

[42] Murugappan, S., and Ramani, K., 2009, "FEAsy: A Sketch-Based Interface Integrating Structural Analysis in Early Design," ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference IDETC/CIE 2009, pp. 743–752.

[43] Tomiyama, T., Kiriyama, T., Takeda, H., Xue, D., and Yoshikawa, H., 1989, "Metamodel: A Key to Intelligent CAD Systems," Res. Eng. Des., **1**(1), pp. 19–34.

[44] Yoshikawa, H., Tomiyama, T., Kiriyama, T., and Umeda, Y., 1994, "An Integrated Modeling Environment Using the Metamodel," Ann. CIRP-Manuf. Technol., **43**(1), pp. 121–124.

[45] Cutkosky, M. R., Engelmore, R. S., Fikes, R. E., Gennesereth, M. R., Gruber, T. R., Mark, W. S., Tenenbaum, J. M., and Weber, J. C., 1993, "PACT: An Expriement in Integrating Concurrent Engineering Systems," Computer, **26**(1), pp. 28–37.

[46] Andreasen, M. M., 1980, "Machine Design Methods Based on Systematic Approach," Ph.D. thesis, Lund University, Sweden.

[47] Hansen, C. T., and Andreasen, M. M., 2002, "Two Approaches to Synthesis Based on the Domain Theory," *Engineering Design Synthesis*, Amaresh Chakrabarti (ed)., Springer-Verlag, London.

[48] Mortensen, N. H., 2000, "Design Modeling in a Designer's Workbench," Ph.D. thesis, Technical University of Denmark, Denmark.

[49] Kiriyama, T., Tomiyama, T., and Yoshikawa, H., 1991, "A Model Integration Framework for Cooperative Design," Comput. Aided Cooperative Prod. Dev., **492**, pp. 126–139.

[50] Linde Werdelin, 2012, http://www.lindewerdelin.com/

11.4 PAPER D

"Classification of Product-Related Dependencies in Development of Mechatronic Products"

Submitted to the Research in Engineering Design journal, 2012

CLASSIFICATION OF PRODUCT-RELATED DEPENDENCIES IN DEVELOPMENT OF MECHATRONIC PRODUCTS

Jonas Mørkeberg Torry-Smith[1], Niels Henrik Mortensen[1], Sofiane Achiche[2]

[1]Department of mechanical engineering
Engineering Design and Product Development Section
Technical University of Denmark
2800, Kgs. Lyngby, Denmark
jtor@mek.dtu.dk, nhmo@mek.dtu.dk

[2]Department of mechanical engineering
Ecole Polytechnique de Montréal, Université de Montréal
C.P. 6079, succ. Centre-ville, Mtl, H3C 3A7, Qc, Canada
sofiane.achiche@polymtl.ca

Abstract

When designing mechatronic products 'complex dependencies' are often reported to be a major challenge. This paper focuses on managing dependencies between attributes of the product during the design process. The literature study shows that there is a gap in the literature with regards to the classification of product-related dependencies. Traditionally these dependencies have been described as appearing between the following product attributes: function, properties and structure. By analysing three mechatronic projects from industry we identified and classified 13 types of product-related dependencies. Each product-related dependency is described and illustrated using the practical examples from the industrial projects. The value of the classification is evaluated by applying it to an industrial development setting not used for the analysis. The evaluation shows that delays in the project schedule, loss of functionality and quality issues can be avoided if attention is directed toward the product-related dependencies in the development process.

Keywords: Mechatronics, Complexity, Dependency management, Classification.

1 Introduction

Companies involved in designing mechatronic products face the challenge of orchestrating a multi-disciplinary design effort. A key challenge is synchronizing the development activities performed by the involved engineering disciplines to obtain a concurrent engineering process (Torry-Smith et al. 2012; Gausemeier et al. 2009). Andreasen and McAloone (2001) show that in industry the onset of the mechanical, electronic and software development activities tends to be shifted causing the activities to be executed in a sequential manner rather than as a concurrent process. A concurrent process is a means to decrease the project lead-time and to harvest the synergy of a multi-disciplinary design. It requires synchronisation and control of dependencies in the design process. In this paper we focus on the dependencies in the mechatronic product, which are created as a consequence of the design process between the different engineering disciplines. The investigation and classification of the dependencies is the scope of this article, and they can be found between functions, properties and the structure of a product. The dependencies are consciously and unconsciously designed into the product concept as one is moving from the product idea to the finished product. The risk of failure for a design team increases if dependencies are not carefully considered, and often the level of control of the dependencies will be revealed at integration tests. If integration tests fail the team has to spend time fixing the problems, instead of spending time on the value-adding activities planned for the next stage of the development project.

We aim to classify these types of dependencies within mechatronic products to enable the design team to identify and manipulate them properly. This can contribute to a more transparent development process and at the same time reduce the risk of delays, loss of functionality, and quality issues by allowing the design team to discover undesirable dependencies as early as possible. An example of a dependency created in a product is the physical interface between an electronic sensor and the mechanical housing. Other dependencies can be found between functions and components in the product. These types of dependencies between attributes in the product, which is the focus in this article, will be called product-related dependencies, since other dependencies can be found in the design process, e.g. between design activities.

Dependency management is a well-established research area where dependency modelling via matrices such as Design Structure Matrices (Lindemann et al. 2009) and Domain Mapping Matrices (Danilovic and Browning 2007) are two branches. Within this research a further classification of dependencies between the attributes of products, which goes beyond the relationships between functions and structures, does not seem to exist. Indirect descriptions of product-related dependencies can, to a certain extent, be deduced from work on proposals for 'a mechatronic concept'. One example is from Gausemeier's work on SLFS (Gausemeier et al. 2009), which is a suggestion based on 8 modelled views, each representing central aspects of the concept. However, these contributions might highlight some product-related dependencies, but they do not propose a classification of product-related dependencies to be managed during the product development process.

The fundamental design theory *The Domain Theory* (Andreasen 1980; Hansen and Andreasen 2002) lay out a pattern for generic product-related dependencies based the attributes: functions, properties and structure. By using three mechatronic projects from industry we identify a number of mechatronic-specific dependencies, which can be classified, and grouped according to the generic dependencies obtained from the *Domain Theory*. The value of identifying and being able to manipulate the product-related dependencies in the development process is evaluated by deploying them in an industrial project-setting.

The structure of this article is as follows: Section 2 describes the background in terms of the research method, the theoretical foundation and the analysed industrial projects. In section 3 related work is reviewed. In section 4 the result of the classification of product-related dependencies is presented. An evaluation of the product-related dependencies applied in an industrial setting is in section 5. Our findings are summed up and concluded upon in section 6.

2 Background

In this article we focus on mechatronic products, which we define as products for which a coordinated effort between the following engineering disciplines takes place: mechanical engineering, electronics engineering, and software engineering.

2.1 Research method

Firstly, the theoretical foundation for viewing technical systems is described, which is used for deducting classes of dependencies, which would be expected to be found in industrial product development projects. A literature review of related work is conducted to reveal if previous attempts on classifications of dependencies have been performed. Having established the theoretical framework we use three selected projects from industry to make observations of dependencies, which are considered important seen from the point of view of the involved development engineers. Documentation from the projects is reviewed and interviews performed and all exposed dependencies are noted down. The dependencies are then grouped by use of an affinity diagram, from which the classes of dependencies are deducted. The results are evaluated in terms of *usability*, *applicability* and *usefulness* (Blessing and Chakrabarti 2009) by applying the found dependencies in an industrial development setting. The research method is depicted in Fig. 1.

Three projects from industry have been investigated (see Table 1): A stand-alone cooling module for vending machines, a continuous blood sugar measurement device for Intensive Care Units at hospitals and an advanced outdoor wrist watch system. The investigation is based on semi-structured interviews of the design engineers as well as analysing the product documentation. The product concept descriptions performed during the projects were extensively documented with sketches for the mechanical development, in which dependencies to the other engineering disciplines were described. The electronics was documented through user requirement specification documents. The software was documented in a conceptual manner and modelled by use of 'use cases' and flow diagrams known from SysML (Object Management Group 2010). The decomposition of the conceptual description to detailed description for key areas is also described in the software documentation. The product documentation was then compared with minutes from integration meetings to reveal the considerations behind the decisions.

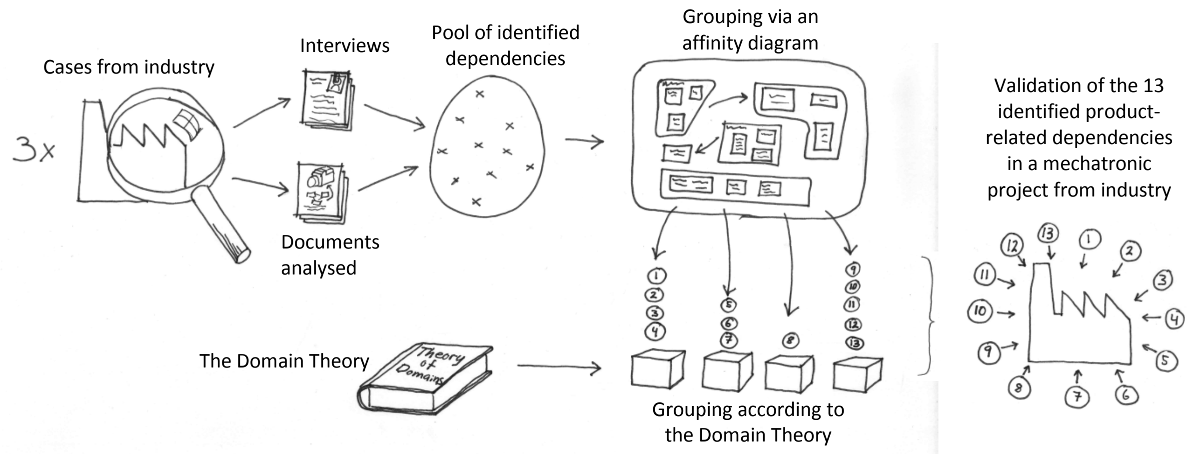


Fig. 1 Illustration of the research method used

2.2 Theoretical foundation

Most theories on design suggested by researchers describe a product in terms of product attributes. So does the *Domain Theory* by Andreasen (1980), which we will use for the definition of product-related dependencies. The theory has been chosen due the elaborate definitions and descriptions of terms used for describing the design object. In the framework proposed by Andreasen the functions and properties of a product constitute the behaviour of the product. The structure is what realizes the behaviour of the product. In mechanical engineering the physical structure realizes the behaviour, in electronics the electronics is realising the behaviour and within software engineering the software code is creating the behaviour of the products. These are expressions of technical means in a product. The division between functions, properties and means seems to be in concordance with the terms within software engineering. In the work by Peters and Pedrycz (2000) on software engineering a distinction is made between functional requirements, non-functional requirements and technical solutions. Similar terminology is used by Pressman (2005). In spite of intensive investigation of references within electronics it has not been possible to find publications on fundamental design theories establishing terms for the design object. The reason seems to be that design of electronics is application-specific and to a large extent based on adaption of standard designs. In electronics design emphasis seems to be on mathematical analysis and simulation of the behaviour of the system. So in the absence of design theory for electronics, the terms found in VDI2206 guideline (Association of German Engineers 2004) for mechatronic products are used for comparison of adequate terms for describing the design object. In the VDI 2206 guideline the terms used for describing the design object is a combined set of what is used in mechanical and software engineering. Two of the primary sources are the Engineering Design by Pahl et al. (2007) and the V-model originating from the software engineering discipline. Based on the reviewed literature sources the three terms (functions, properties and means) can be used to describe the product across the engineering disciplines. We will use these terms to define product-related dependencies and to establish the first classification of the dependencies. A product-related dependency is a dependency between two or more of the following attributes of a product: *function*, *property* and *means*. The nature of the dependencies we are looking for is related to the consequences of the decisions about the attributes made by the designer, meaning that a change of one attribute will affect other attributes in the product. E.g. a changed *means* might change some *properties* of the product. The dependencies can be observed as dependencies between attributes in one product concept.

In Fig. 2, the following abbreviations are used: *Fu* for *function*, *M* for *means* and *Pr* for *property*. Six dependencies can be created (shown in Fig. 2) as a result of a purely combinatorial exercise and will be described in the following text.

(1) **Fu-Fu:** A dependency between two *function*s is described by the link that is created when a *function* reacts to a stimulus created by another *function*.
(2) **M-M:** A dependency between two *means* in the product.
(3) **Fu-M:** A *function* is realised by a *means* and a *means* can be further de-composed into *sub-functions*, which creates the dependency between *functions* and *means*.
(4) **Pr-M:** *Properties* are realised by *means*, thereby creating dependencies between *means* and *properties*.
(5) **Pr-Fu:** There is no direct relation between a *function* and a *property*. Both are realised by *means* according to the Theory of Domains. A link can be established by combining the two relations Fu-M and the Pr-M. Therefore the Fu-Pr relation will not be described as a separate relation.
(6) **Pr-Pr:** There is no direct relation between a *property* and a *property*, and the argumentation is the same as the Pr-Fu relation. Both the Pr-Fu and the Pr-Pr dependencies are coupled to the design process in trade-offs settings and cannot be viewed as a dependency when viewing one product concept.

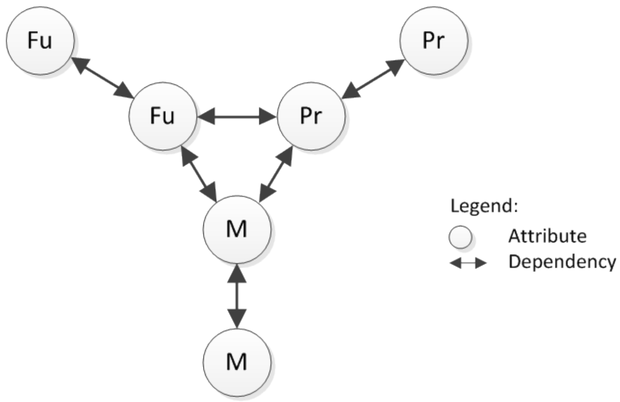


Fig. 2 Overview of product attributes and possible dependencies

It should be possible to identify dependencies 1 to 4 in the cases we have from industry. However, as presented above they are abstractly defined are not mechatronic specific. We will show that a further categorisation, which is more tangible and mechatronic-specific, will benefit the development of mechatronic products. This is presented in section 4.

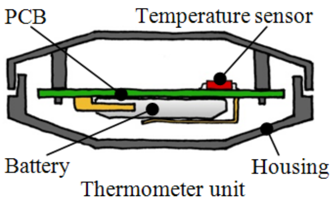
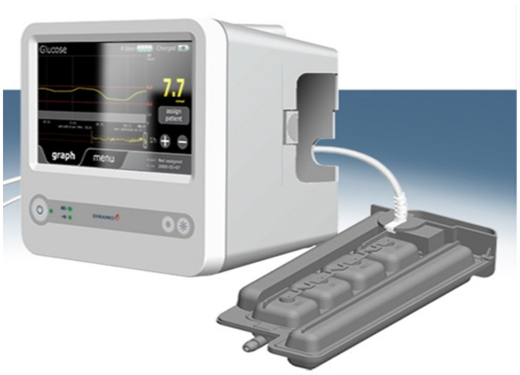
2.3 Projects cases used for analysis of dependencies

The three following cases from industry were investigated to reveal product-related dependencies:

(1) Cooling module for vending machines
(2) Thermometer unit for luxury outdoor sport watch system
(3) Continuous blood sugar measurement device for the Intensive Care Unit at hospitals

In Table 1 a brief description can be found of the analysed projects as well as the developed products.

Table 1 Overview of products used for analysis of product-related dependencies

The project	The product	Product description	Project description
Cooling module for vending machines		The cooling unit for vending machines contains electromechanical and mechanical cooling components, a cooling control unit and unit for communication with external systems. Also encapsulation was designed.	The project was initiated by a large Danish company. It was a high-volume product and focus was on cost and production performance as well as considering market variants.
Thermometer unit for luxury outdoor sport watch system	  	The watch system contains a mechanical watch, an attachable instrument with outdoor *functions* and external units. The analysis of the project is focused on the external unit, which wirelessly transmits the recorded surrounding temperature to the instrument.	The project was launched by two entrepreneurs, with the goal of inventing a conceptually different watch concept. It was aimed at the luxury segment and focus was on functionality and high performance above all.
Continuous blood sugar measurement device for the Intensive Care Unit at hospitals		The device detects light emission caused by a reaction between blood and chemical agents. The product is comprised of a unit located on the patients arm and a base unit located close to the patient. Both units contain mechanics, electronics and software.	The project was launched by a large Danish company because a market opportunity was spotted and they possessed competencies within micro-fluidic systems and chemical analysis. Focus was on functionality, safety and low cost production.

The projects lasted between 2 and 4 years and incorporated engineers from various technical disciplines. The number of development engineers involved in the projects was: 2-5 mechanical engineers, 1-3 electronics engineers and 2-4 software engineers. Other engineers with application-specific knowledge would also participate in the development. In the cooling module project the mechanical development was outsourced to a consultancy company, whereas the electronics and programming were performed in-house. In the watch project and in the blood sugar measuring device project the development activities were outsourced to consultancies companies. One company was in charge of the mechanical development and another company was in charge of the electronics and software development. During the development the teams had meetings approximately on a weekly basis or when needed.

3 Related work on product-related dependencies

In this section we review the mechatronics literature for a classification of product-related dependencies. The search for related work is structured based upon a previous study (Torry-Smith et al. 2012; Torry-Smith et al. 2011), carried out by the authors of this article. The study (Torry-Smith *et. al.* 2010) is built on data processing of more than 500 references, and reveals a number of research areas wherein suggested solutions on how to deal with mechatronic-specific challenges. The following research areas were identified:

1. Activities based on functional thinking (Buur 1990; Nagel et al. 2008; Tomiyama et al. 2007; Suh 2006)
2. Relationship management, DSM and DMM (Braun and Lindemann 2007; Danilovic and Browning 2007), QFD (Hauser and Clausing 1988; Bonnema 2011), FunKey (Bonnema 2010).
3. Controlling design activities through requirements management, Woestenenk et al. (2010) and Systems Engineering (Sage and Rouse 2009).
4. A process model containing activities for the development process (Isermann 2005; Association of German Engineers 2004; Salminen and Verho 1989), including the Systems Engineering process model (Sage and Rouse 2009).
5. Informal description consisting of a number of modelled/described aspects to specify a system, e.g. A3 overviews (Borches and Bonnema 2010; Salminen and Verho 1989; Buur 1990).
6. Modeling languages to describe systems as a whole, formally or semi-formally (Object Management Group 2010), SFSL by (Gausemeier et al. 2001).
7. Model transformation from a design model in one domain into a design model in another domain (Gausemeier et al. 2009), (Shah et al. 2010).
8. Formalized specification of interfaces, (ISO/IEC 81346 2012), Systems engineering (Sage and Rouse 2009).
9. Simulation of phenomena with cross-disciplinary elements, e.g. Dymola (Systems 2011) and (Modelica Association 2010).
10. Setting up a systems integration group in the project (Adamsson 2004; Andreasen and McAloone 2001).

The first area regarding ‘Activities based on functional thinking’ is widely applicable and very generic. Besides a description of the state-transition aspect this area does not provide a further classification of dependencies, which go beyond the study of fundamental design theories as those presented in section 2.2.

The area ‘Relationship management’ e.g. comprising DSM, DMM, QFD operate with a classification of product-related dependencies similar to the categories of the Theory of Domains. In DSM classes of the same kind are compared (e.g. Fu-Fu relationship) whereas different types of classes are compared in in DMM. In DMM the relationship between structure of the product and *functions* in a product can be mapped. The classification in ‘relationship management’, which is aimed at product-related dependencies, consists of three classes; namely: *function-function*, structure-structure and *function*-structure. Further detailing of dependencies besides that is not the aim of the methods.

‘Controlling design activities through requirement management’ can potentially encompass many different types of product-related dependencies, since many requirement categories are found in the general theory on requirement management. However, the literature which is specific to mechatronics does not provide a description of a classification of product-related dependencies.

Related work on process models covers System Engineering, the VDI2206 and similar models and description. Within this research area we might expect to find a classification of product-related dependencies, needed to be handled by the project teams to facilitate the integration process. Systems Engineering does describe activities related to functional analysis and how to break down the product into sub-modules to be handled by different teams. The functional dependencies are handled thoroughly whereas other dependencies created in that process are only briefly described. A classification within Systems engineering does not seem to have been established. The VDI2206 is a general description of the process incorporating the V model with the design object description found in Pahl et al. (2007). However, a classification of product-related dependencies is not stated in either of the references.

The research area 5 with the label 'Informal descriptions' provides a suggestion for a product classification based on what is depicted in Fig. 2, and thus, does not add to the classification already found. 'Semiformal and formal descriptions' such as Gausemeier's SFSL and the SysML, facilitates that product-related dependencies can be modelled in software applications. Gausemeier's SFSL provides eight views called *partial models*, which is used for describing the principle solution for a mechatronic concept.. Three of these views are product-related and are labelled *Functions*, Behaviour and Shape, and comprise modelled descriptions of the concept according to the labels. In addition, a view called 'active structure' is used to describe the control part of the product. In the active structure view data communication between *means* is modelled, which constitute an M-M dependency. No further classification is performed in SFSL. SysML models, which are often used to model the control view in a product, can express product-related dependencies. The question is whether the dependencies, which can be modelled are classified, or if the type of dependencies is expected to be defined by the designer from project to project? The latter scenario seems to be the case, thus not revealing a classification of product-related dependencies.

Related work on 'Model transformation and use of Metamodels' deals with dependencies that focus on exchanging parameters between computer models. From that point of view, functional modelling and the inherent dependencies between *functions* is or can be modelled but a classification of product-related dependencies cannot be found within that area.

'Formal interface descriptions' relates structure to structure. A further division into sub-categories does not seem to exist aimed at mechatronic products (ISO/IEC 81346 2010).

'Simulation of phenomena' will be a simulation of one or more *properties* revealing dependencies between attributes of the product. Examples are simulation tools such as Modelica, Dymolink and Bondgraphs, used for assessing control issues and dynamic performance. These simulation tools try to integrate electronics, software and mechanical considerations and include them in one simulation setting where parameters can be optimized. Due to the equation-based simulation a holistic approach of classifying product-related dependencies cannot be found within this area.

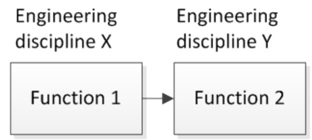
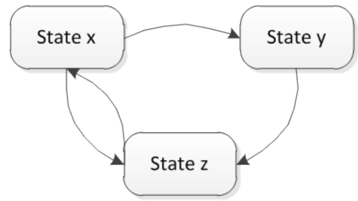
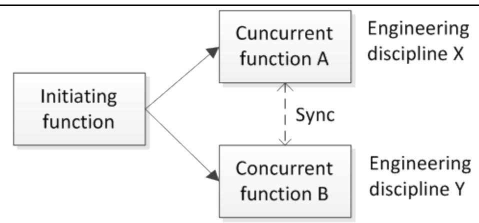
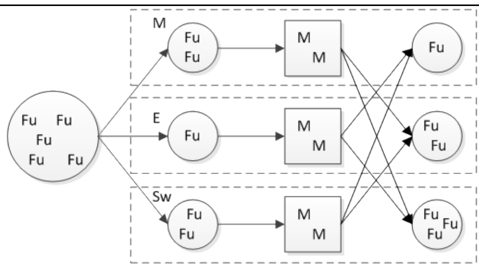
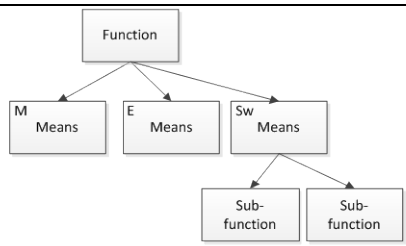
The last stated research area, aimed at supporting design of mechatronics, is deployment of 'systems integration groups'. These groups facilitate meetings between the engineering disciplines and focus on a holistic view of the system. If any, this should be the area where we could find suggestions for product-related dependencies, important to handle in the design process. However, this is not the case. Most descriptions of the systems integration group's role in mechatronic projects are carried out from a process or an organisational point of view. When viewing the role of the system group the descriptions of the performed activities of that group are held at an abstract level and do not describe or classify product-related dependencies.

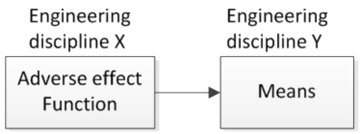
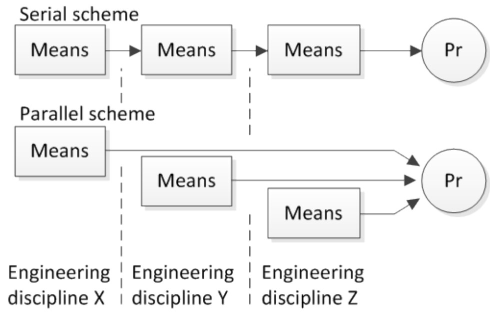
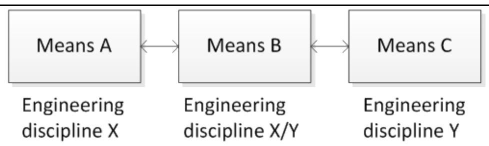
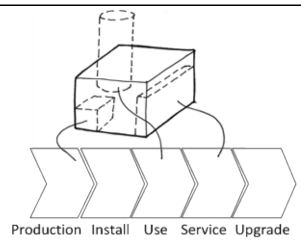
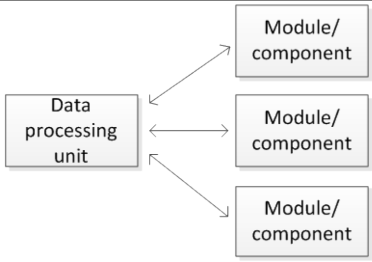
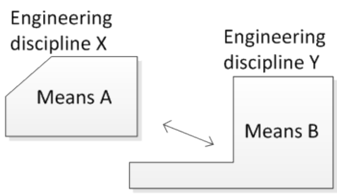
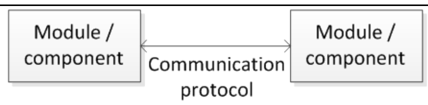
The attempt of suggesting a classification of product-related dependencies in a mechatronic context is not found in the literature that goes beyond the generic dependencies obtained from Fig. 2. The closest reference is the mindset behind SFSL by Gausemeier, in which four views are related to product-related dependencies, although heavily influenced by control engineering issues. Yet, new aspects of product-related dependencies are obtained from the literature study on design of mechatronics even though a classification could not be identified. The literature study reveals that we should expect to find the state-transition phenomenon as an aspect of the *function-function* dependency and that physical and communication interfaces can be expected to be observed in the cases as a part of the *means-means* dependency category.

4 Description of the 13 identified product-related dependencies

The analysis of the projects revealed a vast amount of product-related dependencies and these were grouped by using an affinity diagram. By completing the diagram, 13 categories of product-related dependencies were revealed. The 13 product-related dependencies are briefly described in Table 2, and subsequently described more thoroughly in what follows. In the description each category has been consolidated by a generic description of the nature of the product-related dependency.

Table 2 Summary of the identified dependencies

Domain Theory categories	Id #	Identified product-related dependencies	Description	Illustration of the dependency
Fu-Fu	1	Causal function	Interactions between *functions* when the functionality of the product is seen as a process flow	
	2	State/time function	Dynamic relations between *functions*, where *functions* are executed at specific time or at specific events.	
	3	Sync function	Dynamic relation between *functions* where *functions* reacts on stimuli and the timing of the stimuli is important	
	4	Response function	*Functions* react on stimuli from other *functions*. The size and type of the stimuli have to be matched between the *functions*.	
Fu-M	5	Fu-M disposition	Proposing *means* to functions in one domain will often have consequences in other domains in terms of supporting *functions*.	
	6	Cumulative Fu-M	The realisation of a *function* may require *means* from various disciplines.	

	7	Adverse effect	A *means* may have an adverse effect associated to it. The undesired adverse effect can be formulated as a function (e.g. 'create vibration').	
Pr-M	8	Property scheme	A single *property* of a product may have influencing factors allocated to various *means*. How these *means* contribute to the one *property* is important to clarify to optimise the products performance.	
M-M	9	Multi-disciplinary means	Some *means* have to satisfy boundary conditions, which are important to more than one engineering discipline.	
	10	Volume allocation	Physical *means* have to be located spatially in the product and the volume may have changing restrictions during the life phases of the product.	
	11	Liveliness	The flow of information between electronics and software must be designed without causing a system-lock, which requires a cross-disciplinary effort.	
	12	Physical interface	Physical interfaces between modules and components have stakeholders from electronics and mechanical engineering.	
	13	Communication interface	Digital components may have analogue communication incorporated as well as analogue components may have a digital port. To ensure seamless integration,	

			communication protocols must be evaluated and agreed upon.	

The identified product-related dependencies can be organized according to the categories presented in Fig. 2. This does not come as a surprise since Fig. 2 represents fundamental dependencies between the attributes of a product. Ordering the identified product-related dependencies according to Fig. 2 serves the purpose of organizing the dependencies and it clearly shows the link to the fundamental design theory. In Fig. 3 the identified dependencies have been circled in red. All but two dependencies have been identified. The Fu-Pr and the Pr-Pr dependency could not be identified. We have previously argued that there is no direct link between Fu-Pr and between Pr-Pr, which seems to be the case, also in the projects. As a curiosity the Fu-Pr and the Pr-Pr dependency were discussed in the project but it was in terms of comparing two concepts in a trade-off evaluation and the *means*, which realised the *functions* and *properties*, were central to the discussion highlighting the *means* as the mediator.

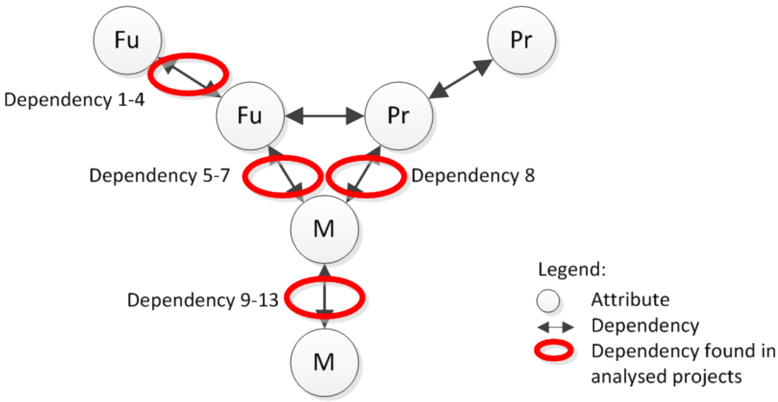


Fig. 3 Overview of the categories of dependencies (in red), which the 13 identified product-related dependencies fall into

4.1 The thirteen product-related dependencies

In the following each of the product-related dependencies is described and a concrete example is stated to highlight the relevance when designing mechatronic products. It is primarily described via the three investigated cases.

4.1.1. Dependencies between functions (Fu-Fu)

(1) **The Causal function dependency**. *Functions* might be sketched in tree structures, modelled in software or treated by discussions on 'what the product should do'. Determining the *functions* and their interactions will greatly influence the design task and careful attention should be paid to coordinate the effort between the engineering disciplines. In the analysed projects the *functions* are primarily handled via discussions of what we want the product to do. They could be formulated in the project as: Do we need to measure the temperature of the chemicals to make the calculation of the sugar level of the blood? Should we measure the temperature before or after the analysis of certain molecules in the chemical compound (see Fig. 4)? These decisions require a discussion between the engineers from the different disciplines directed at finding the causal relationship between the *functions* in the product. The generic aspect of the product-related dependency is illustrated in Fig. 5.

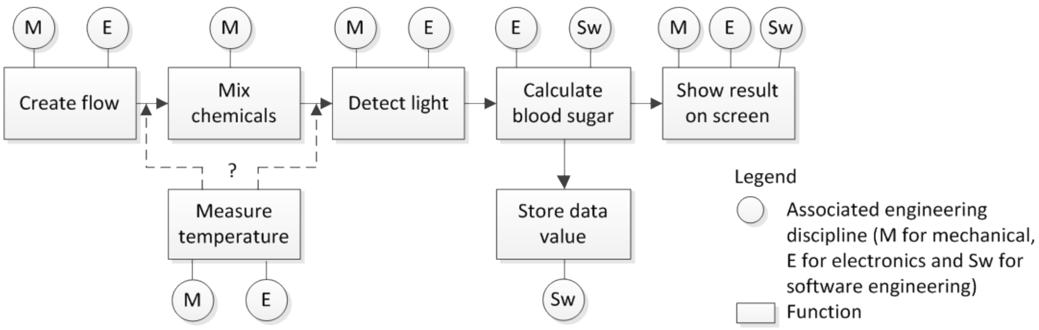


Fig. 4 The causal dependencies between *functions* illustrated for the development of the blood sugar measurement device

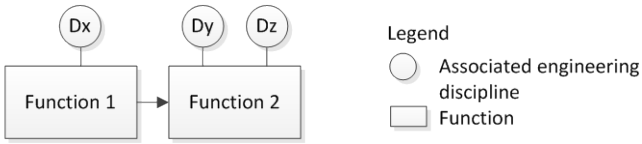
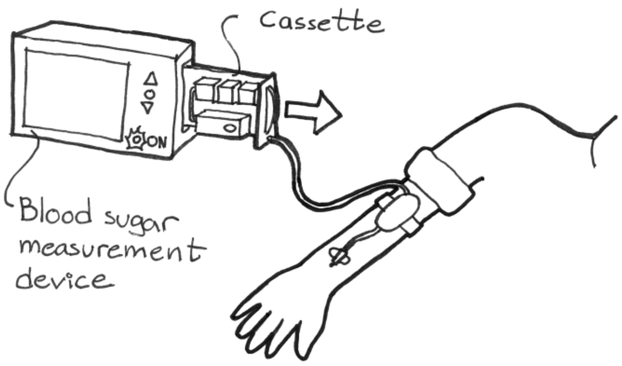


Fig. 5 The generic aspect the 'causal function dependency', where Dx, Dy and Dz are representing three different engineering disciplines

(2) **The State/time function dependency**. An important aspect for the functionality of the product is the sequence in which *functions* will be initiated or in what states the particular *functions* are active in. In the project about the blood sugar measuring device it can be discussed if it should be possible for the user to retract the cassette of chemicals while the device is running or if it is only possible to do so when the device is in Off-mode (illustrated in Fig. 6a). The analysis of the projects reveals that the dependencies related to *functions* viewed in a time perspective or in a state perspective have to be assessed and synthesised and coordinated by engineers from the different disciplines. The generic aspect of the 'state/time function dependency' is depicted in Fig. 6b.

a



b

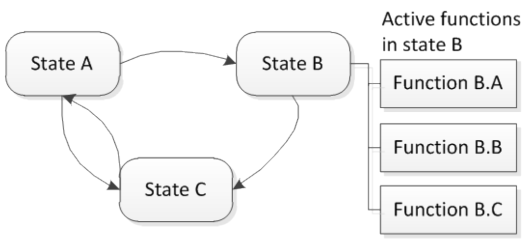


Fig. 6 (a) The scenario where the cassette is removed while the device is on and (b) the generic aspect of the 'state/time function dependency'

(3) **The Sync function dependency**. This product-related dependency is about synchronising certain *functions*. This dependency may not appear in all projects but if it does, special attention has to be directed to it by the development engineers since the effect of not seeing the dependency can be quite significant in the project. This type of product-related dependencies is typically discovered late in the design process and the amount of re-work will therefore be considerable. The dependency appears when two *functions* concurrently perform a task but where the system's state is only detected by

monitoring one of the *functions*. In this situation the system state might switch before both *functions* have been performed, causing unwanted states leading to various problems and malfunctions. The situation is likely to happen when a *function* in one domain (e.g. the mechanical) is initiated and is running concurrently to one or more *functions* in other domains. A concrete example would be a rotating table used in production facilities with fixtures, which has two mechanically determined positions while it is electronically controlled (see Fig. 7). A switch to determine the position might transmit a signal too early/late about the mechanical state of the table, if the synchronization is not thought out carefully. The inter-disciplinary exercise performed by the engineers is about discovering potential sync problems and deciding which *functions* should be higher ranked than others, i.e. decisions regarding 'master/slave' configuration of *functions*.

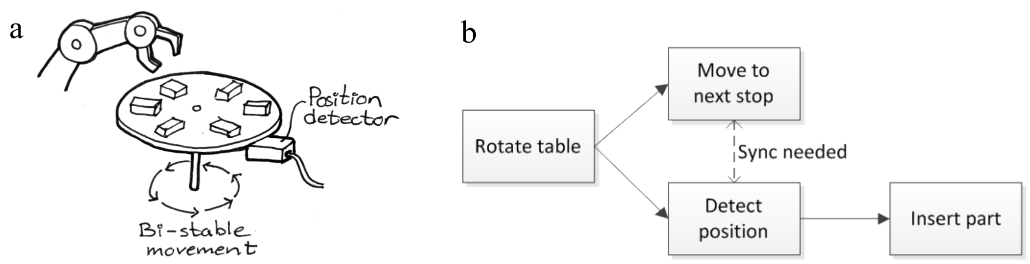


Fig. 7 The scenario of synchronising *functions* for the production round table (a). The generic aspect for the scenario (b)

(4) **The Response function dependency**. A product-related dependency which is of special importance to the control of a product is how *functions* react on a stimuli from other *functions*. For example, how much gain an amplifier should provide depending on the expected input (see Fig 8). If attention is not paid to this dependency the functionality of the product will fail. In the case of the amplifier, the size of the input to the amplifier could lead to an amplification which is out of range of the power supply or that the result of the amplification leads to forces which can damage the structure of the product. This functional view is aimed at ensuring that *functions* which have to be realised in the product are able to interact in the desired way and within defined limits. Considerable research efforts have been directed at providing methods and tools for handling these dependencies. Some of the tools are Simulink, Dymola, Modelica and SysML. Another example of what can happen if the input and output are not harmonised is a data overflow, which can cause errors in the products' behaviour. Such an overflow problem caused the European Ariane 5 rocket to crash in 1996 on its first launch (Siam 1996).

Fig 8 Example of how the amplifying *function* responds to the input (a). The generic aspect of the 'response function dependency'(b)

4.1.2 Dependencies between functions and means (Fu-M)

(5) **The Fu-M disposition dependency**. Dependencies appear because *functions* are transformed into *means*, which again are transformed into *sub-functions*. In design of mechatronics, managing these dependencies poses a challenge because *means* selected by engineers representing one discipline may require *sub-functions* to be established in one of the other engineering disciplines. Similarly, a *function* needed by one engineering discipline may require *means* to be established by other engineering disciplines. This is illustrated in Fig. 9b.

An example of this type of product-related dependency could be a situation where a battery is needed by the electronic engineers (to realise the *function* ‘provide power’). However, it will be the task of the mechanical engineer to make space for it, create structure to encapsulate it and maybe also to provide the possibility of changing batteries. This is a straight forward example and the dependencies are somewhat clear but in a case where a sensor is replaced by another type of sensor, the Electrostatic Discharge (ESD) performance of the new sensor might not have been evaluated. Therefore, a choice of the low cost sensor can lead to an added required functionality ‘protecting against ESD’. The mechanical engineers, who have to create a solution, did not expect this added functionality and the task of finding proper *means* and incorporating it into the design will prolong the development time; this example is illustrated in Fig. 9a. If we are not aware of the consequences of our decisions we will indirectly force design tasks to be done in other domains, which is a major unnecessary risk to introduce in a project. Since different engineering disciplines are involved in the design process it becomes difficult to see the consequences of one own’s decisions in the other engineering disciplines.

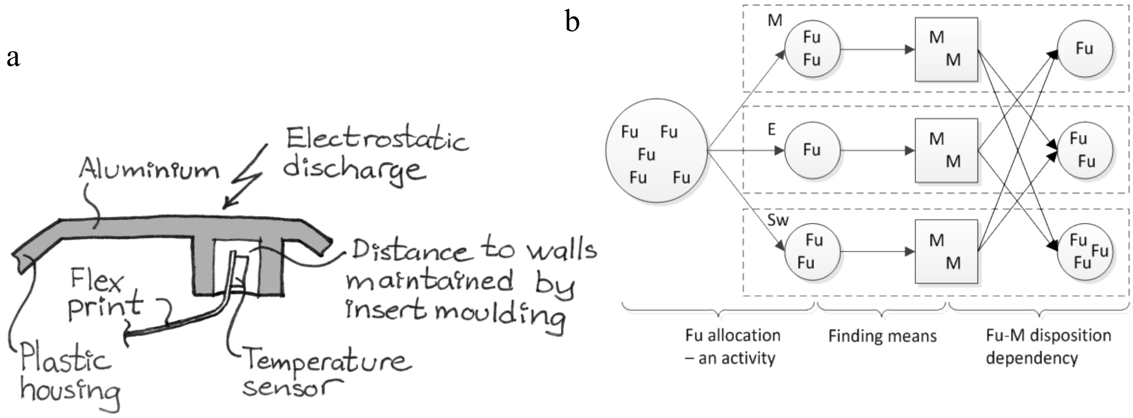


Fig. 9 Example of ESD protection of an electronic component (a). The generic aspect of the ‘Fu-M disposition dependency’ (b)

(6) **The Cumulative Fu-M dependency**. In mechatronic products, several *means* can contribute to realise a single *function*, and these *means* may be handled by different engineering disciplines making a joint effort necessary. To realise the *function* ‘transmitting the wireless signal’ in the watch project, various *means* had to be established as illustrated in Fig. 10a. It had to be done by a coordinated effort by the mechanical, the electronics and the software engineers. Space had to be created to accommodate the antenna, and considerations regarding other metal objects in the product had to be made to avoid interference with the electromagnetic waves. The size and shape of the antenna had to be discussed between the mechanical engineers and the electronic engineers due to space restrictions. The electronic engineers and the software engineers worked closely together in finding and controlling the components for ensuring a sufficient performance of the transmission. For this task the engineers have to work together to realise the *functions*. This is challenging since many studies show disintegration between the engineering disciplines rather than integration (Gausemeier et al. 2008).

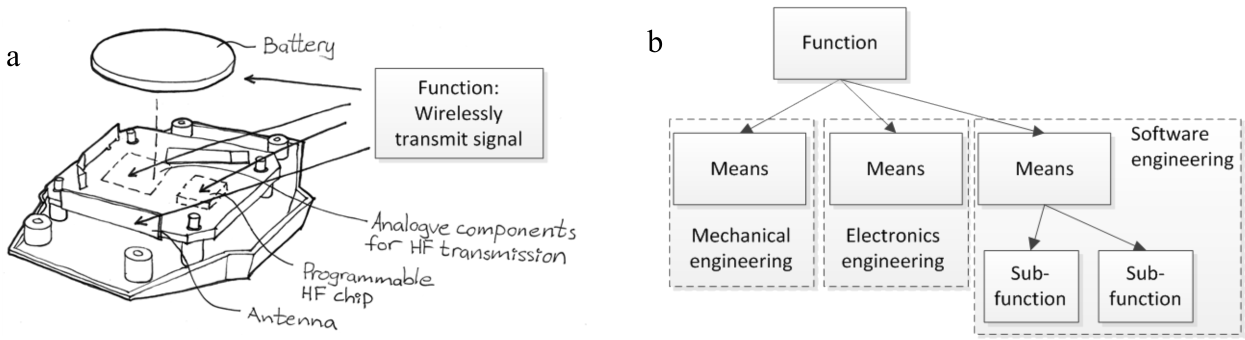


Fig. 10 The *function* ‘wirelessly transmit signal’ is realised by different *means* solved by the different engineering disciplines (a). The generic aspect of the ‘cumulative Fu-M dependency’ (b)

(7) **Adverse effect dependency**. When finding *means* to realise a *function* we as designers may create solutions with adverse effects such as vibrations. The focus for an engineer is the intended functionality and not the undesired adverse effects, causing them potentially to be overlooked. Adverse effects produced by *means* in a product have the same nature as *functions* and can be described as such, e.g. ‘generate noise’ or ‘create vibrations’. These adverse effects have the potential of interfering with *means* in the product. An example could be some electromagnetic waves which interfere with the LCD screen or heat generated which cause an electronic component to become unstable (see Fig. 11). It is naïve to think that all adverse effects can be acknowledged in advance, but some can be detected in due time and precautions made, which will minimize delays in the design process.

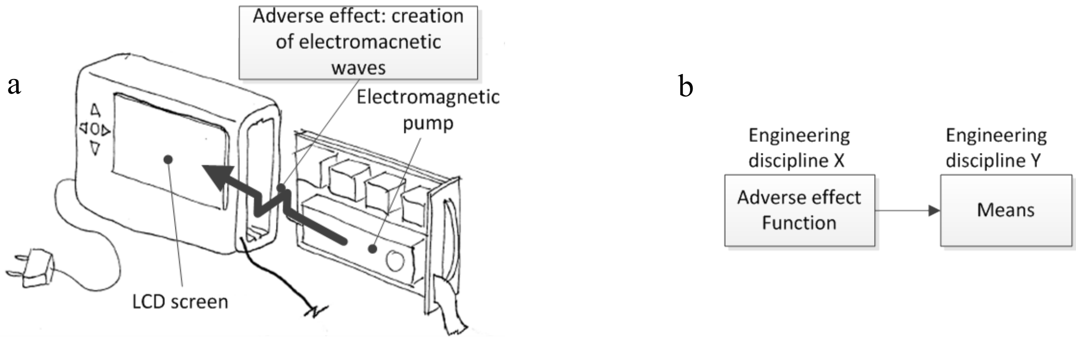


Fig. 11 A components emits electromagnetic waves which have adverse effect on the LCD screen (a). The generic aspect of the ‘Adverse effect dependency’ (b)

4.1.3 Dependencies between properties and means (Pr-M)

(8) **The Property scheme dependency**. The realisation of *properties* will rely on the specific *means* created by the engineers from the different disciplines involved. Some *properties* may be realised only by a single engineering discipline such as the *property* ‘mechanical advantage’ for mechanisms. Other *properties* have to be realised by *means* from more than one discipline. How the *means* contribute to the realisation of a *property* is the topic for this category of product-related dependencies. The *means* will contribute to a given *property* based on a certain scheme, which will be serial, parallel or a combination of the two. This is different from realisation of *functions* where all contributing *means* will have to be established to provide the desired functionality, which will be serial in its nature. As an example of a *property* where the *means* are contributing in a serial manner is the accuracy of the blood sugar measuring device (Fig. 12). Here the steadiness of the flow of the chemical compound, the steadiness of the temperature, the tolerance of the size of the tubes etc. are influencing the accuracy in a proportional manner and thereby having a serial configuration. An example of a parallel configuration is the physical robustness of the thermometer in the watch system. If the mechanical

robustness of the PCB and the soldered components are increased twofold it will not increase the overall robustness equally if some of the other components contributing to the robustness are 'the weakest link in the chain'. Achieving desired *properties* will often be a joint effort between teams, which might represent different engineering disciplines. Change of *properties* will echo downstream resulting in changed *means* and changed functional surfaces. It is therefore an important task to clarify the *means* that contribute to the *properties* and how they contribute. In cases where the contributing factors are influencing the desired *property* in a more complex pattern, i.e. a combination of serial and parallel with different coefficients for the linearity, mathematical models might be used. The serial and the parallel contribution scheme is illustrated in Fig. 13.

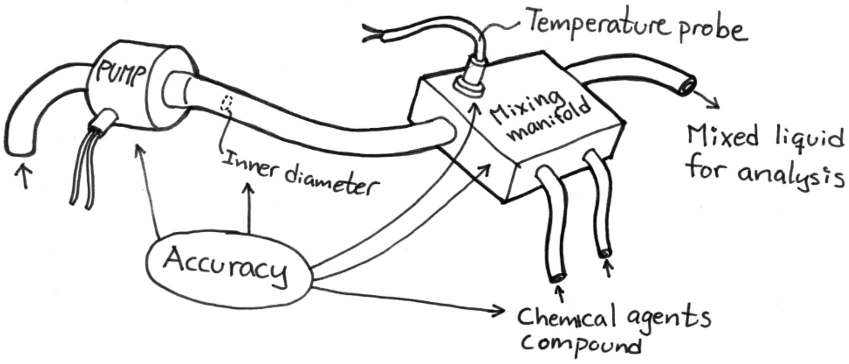


Fig. 12 The accuracy of measuring the blood sugar has different contributing elements which contribute to the accuracy *property* in a serial scheme

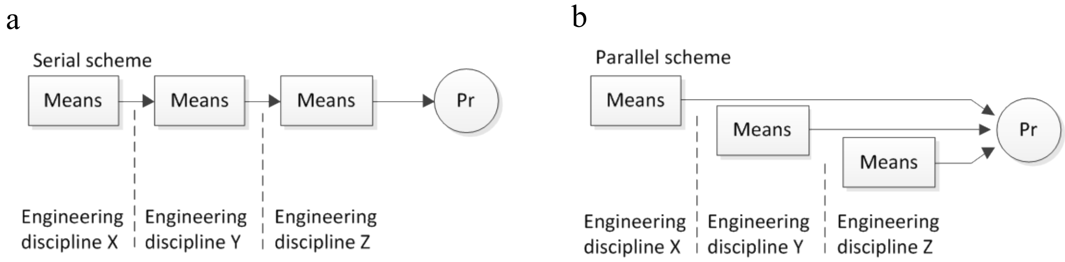


Fig. 13 The generic aspect of the 'property distribution scheme'. (a) depicts the serial scheme, whereas (b) depicts the parallel scheme

4.1.4 Dependencies between means (M-M)

(9) **The Multi-disciplinary means dependency**. When realising *functions* in a mechatronic product some components will have stakeholders from more than one engineering discipline. This poses two challenges. Firstly, it requires a high degree of integration between the engineers. Secondly, data regarding the component quite often has to be manually synchronised between computer tools used within the separate engineering disciplines. Even though considerable efforts have been put into research regarding effective model transformations and use of meta-models, they are not fully operational and the utilisation of the methods is not adopted by the industry yet for various reasons (Torry-Smith et al. 2012). Most common multi-disciplinary *means* in mechatronic products are electromechanical components and programmable digital components. A DC motor is an example of a multi-disciplinary *means* (illustrated in Fig. 14). The mechanical engineers are involved in fastening the motor, damping vibrations, connecting the shaft to other component and possibly considering how to change it in a repair situation. The electronic engineers need the electronic *properties* for controlling it and being able to provide it with the needed current related to control algorithms. The product-related dependency is the relation between a multi-disciplinary *means* and the *means* located in either of the

two engineering disciplines involved.

Fig. 14 (a) The dc motor has stakeholders within mechanical and electronics engineering discipline. (b) The generic aspect of the 'Multi-disciplinary *means* dependency'

(10) **The Volume allocation dependency**. The analysis of the projects shows that the volume allocation aspect plays a significant role when looking at the relationships between *means*. This product-related dependency is important to the mechanical engineer and the electronics engineer, but plays an insignificant role to the software engineer due to the physical nature of the dependency. We see two types of product-related dependencies related to volume allocation. One of them is related to where the components will be located in the product, which may be called 'volume of control'. The other type is volume considerations related to planned activities in the product's life phases, which may be called 'volume of activity'. An example of 'activity of control' is the battery or the antenna in the watch project, which had to be accommodated to the available space in the device. An example of 'volume of activity' is from the blood sugar measurement device project. During the project it turned out to be advantageous to test the printed circuit board in-line in the production after it had been mounted in the housing part. For the electronics to be tested it should be possible to lower a probe from above to touch connection pads on the PCB (see Fig. 15). Therefore space restrictions on the shape of the housing and on the parting lines between the two housing parts became restricted. Technically and theoretically it is possible to model the different volume allocations in a CAD tool as enveloped volumes. This would, however, only be made in the mechanical CAD system since electronic CAD systems are in 2D. Thus the effect of such modelling is limited.

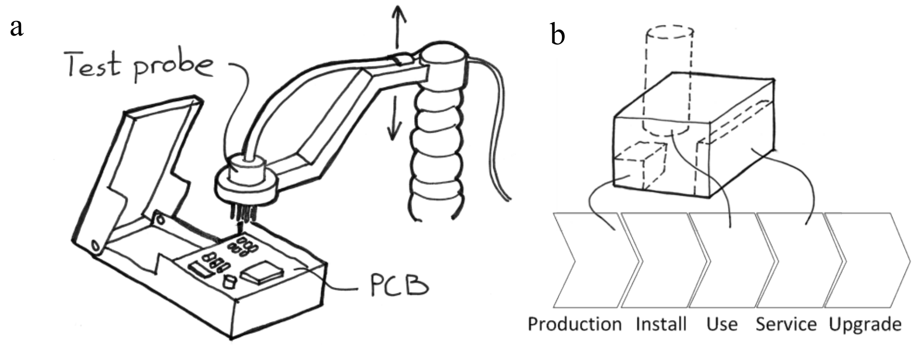


Fig. 15 Inline test of PCB in the production of the blood sugar measurement device (a). The generic aspect of the 'volume allocation dependency' (b)

(11) **The Liveliness dependency**. The dependency covers the considerations about being able to process control signals at any time to ensure the liveliness in the data processing. The dependency appears between *means* due to allocating processing resources. A central example is the monitoring of the different inputs to the CPU provided by the electronic components. In case of the blood sugar measurement device, one consideration is the temperature sensing of the mixed chemicals (Fig. 16). Conceptual considerations are: Should it be based on 'access by request' or should the information flow create 'a flag' in the processing unit letting it know that new information has arrived. It is important to ensure that all signals are processed and that the systems do not freeze. This is of primary concern for Real-Time Systems, which are often found in embedded software systems of mechatronic products. Here processing is linked to events, which have to happen within a time-

window. If the system fails then the control system will most likely fail, causing the overall functionality of the mechatronic product to fail.

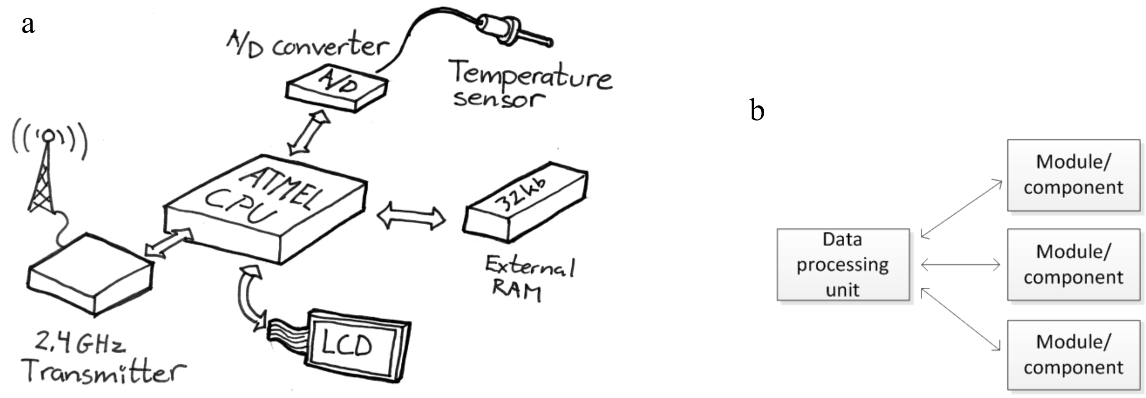


Fig. 16 Some of the communication to and from the CPU (a). The generic aspect of the 'liveliness dependency (b)

(12) The Physical interface dependency. The mechatronic projects reveal that physical interfaces are of major concern to the involved engineers involved in the project. The mechanical engineers and electronics engineers are the prime stakeholders since physical interfaces play an insignificant role to the software engineers. There are physical interfaces in a product structure, which are exclusively handled by the mechanical engineers, e.g. between two mechanical modules. Similarly, there are physical interfaces solely within the electronics domain (e.g. a wire connection via a socket between two PCBs).Therefore; the physical interfaces important to the integration of engineering disciplines are the physical interfaces between mechanical components and electrical components. A sketched consideration of physical interfaces from the blood sugar measurement device is depicted in Fig. 17.

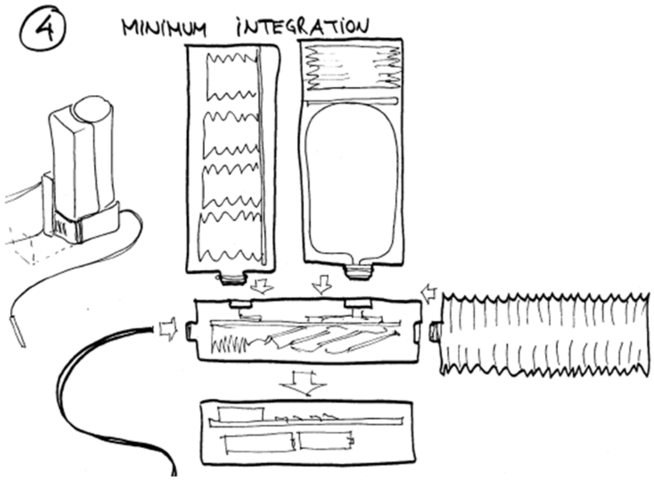


Fig. 17 A sketch of some of the proposed interfaces for the blood sugar measurement device

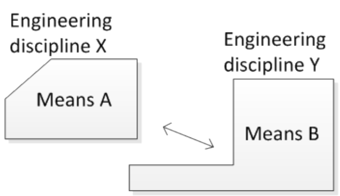


Fig. 18 Depiction of the generic aspect of the 'Physical interface dependency'

(13) **The Communication interface dependency**. Signal processing is a major consideration within design of electronics and software. Thus communication interfaces becomes an important product-related dependency between electronics and software *means*. The communication interface can rely on standardised data protocol or the interface can be custom designed to fit the design task. In Fig. 19 an example of communication protocols is shown. An example of an existing data protocol is the RS232 which is used between the thermometer component and the CPU. The main reason for the use of communication protocols is standardization to increase modularity and to reduce complexity and thereby the size of the development task. What we can observe from the analysed projects is that communication protocols are widely used to specify communication between software and electronic components. Most communication protocols are digital. However, analogue protocols such as the radio on the FM band also exist.

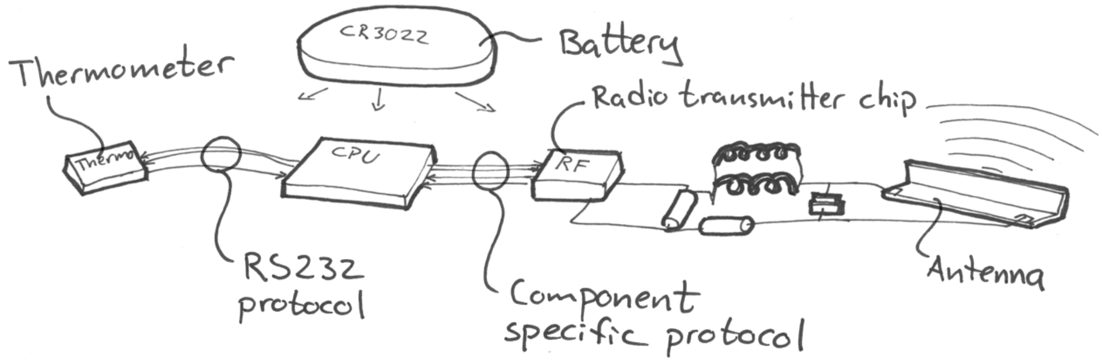


Fig. 19 The communication protocols used in the thermometer unit in the watch project

Fig. 20 Depiction of the generic aspect of the 'Communication interface dependency'

A fair question to ask about the identified product-related dependencies is: Did we particularly look for product-related dependencies which would fit the categories obtained from fundamental design theories? No, it is not the case. The reason why it is possible to group the product-related dependencies according to the categories from fundamental design theory is that *function*, *properties* and *means* do cover all the categories with the ability to comprise any given product-related dependency. However, the generic dependencies do provide an overview of the identified dependencies from the cases, which is what they were intended for.

5 Validation of the identified product-related dependencies

Having identified the product-related dependencies, it is desirable to estimate the value of using them in design practice. For that purpose the guidelines proposed by Blessing and Chakrabati [book 2009] will be used. They recommend three aspects to be evaluated:

- *Usability: the ease with which the method can be used for the intended task;*
- *Applicability: whether it has the intended direct effect on a design process; and*
- *Usefulness: whether the direct effect leads to an improvement in a high-level success factor, taking into account possible adverse effects."*

An additional industrial mechatronic development project was selected to serve as the test arena. The project was about developing an actuated hand for arthritis patients, which can be fitted inside their own palm and provide an enhanced grip. The device can be attached and removed as the user wishes. The aim is to make the device appear discrete when worn at home or in public places. See the rendering in Fig. 21, which served as the visualisation of the product at the beginning of the project.

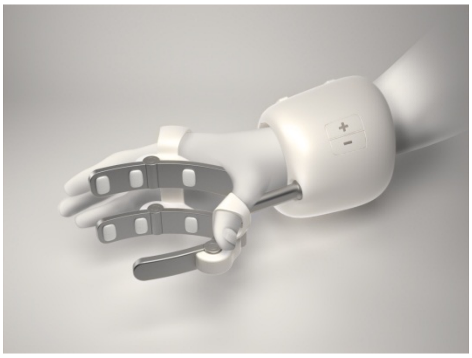


Fig. 21 The vision for the product (computer rendering)

The project was selected because it fitted the needs for evaluation of the product-related dependencies. Both mechanical, electronics and software engineering were represented in the project, and the relatively small size of the project made it possible to monitor all conceptual decisions and infuse the mind-set about the product-related dependencies to all participants. Three mechanical engineers and one electronic/software engineer participated in the development. An agreement with a manufacturing company of linear actuators was made requiring them to develop the actuator in close collaboration with the design team. Besides that, designers, project managers and usability engineers represented the core development team.

The dependency mind-set was introduced in the conceptual phase where the overall functionality had been determined. Solutions had been proposed to illustrate the realisation of the functionality, but well aware that a large part of the *means* would change during the development and that the next layer of functionality should be determined in the process to come. A suggestion of the MMI was proposed but not tested by users yet. Fig. 22 is a sketch of the concept (document from the project) at the time where the dependency mindset was introduced. Fig. 23 shows the functional model half a year later, which is fully operational. We were therefore able to evaluate the value of the product-related dependencies from the conceptual phase and through a detailed design phase aiming at a fully operational functional model (Fig. 23).

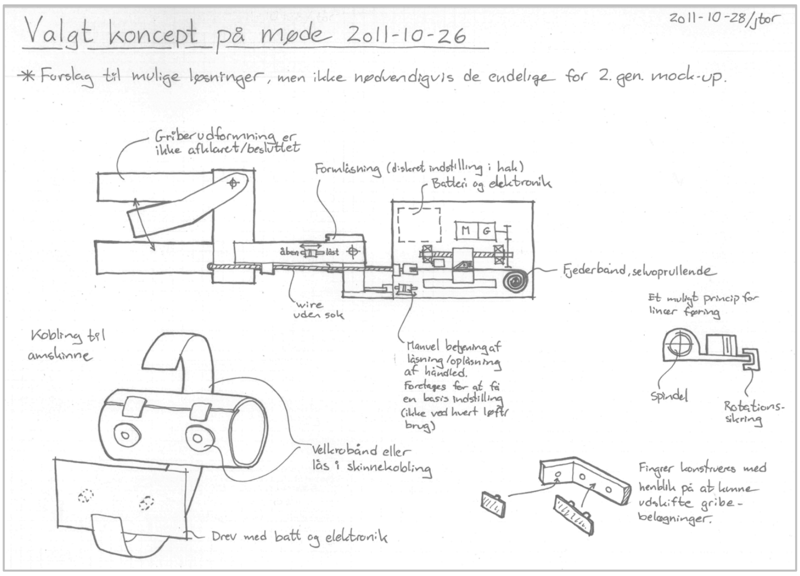


Fig. 22 Sketch of the product concept

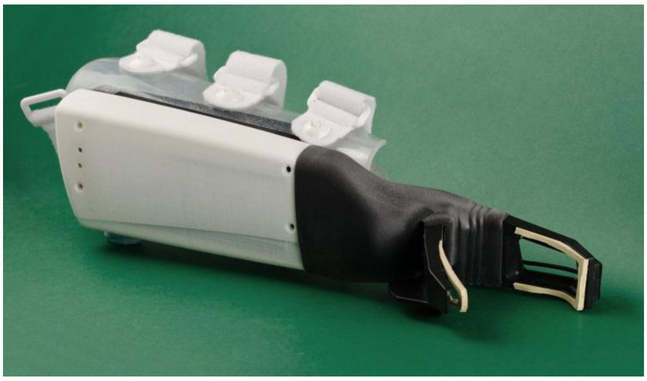


Fig. 23 The functional model of the product concept (fully operational)

The product-related dependencies were infused in the project group as questions aiming at revealing these 13 classifications of dependencies. The questions were asked at integration meetings and also asked to each of the engineers individually in separate sessions. For example, the product-related dependency 'Fu-M disposition' was transformed into the three questions below, which formed the starting point for the discussion. The questions are by intention overlapping in order to ask about the same product-related dependency from different angles and at different abstraction levels.

(1) Try to formulate to the other engineering disciplines what components you plan to utilize to solve the needed functionality in the product.
(2) Identify the lowest level of *functions* and their *means* (use the mental picture of a FU/M-Tree) and ask what new *functions* the lowest level of *means* would require in the other domains.
(3) Ask the other engineering disciplines what considerations they have about the effect of you choosing the *means* you plan to.

The character of the questions was slightly more abstract than the other conceptual considerations, which were carried out in the project. However they were seamlessly integrated in the discussion on integration meetings. In the individual sessions with the engineers the wordings of the questions matched the technical terms used by the engineers, leading to a constructive discussion on product-related dependencies.

When the product-related dependencies were revealed, which called for a design change or a decision at the point in time, these were carried out at once to ensure the quality of the product. The advantage of using a real industrial project was that the project setting was not simulated. The disadvantage was that we could not leave some of the dependencies unattended to monitor consequences of not handling the product-related dependencies. However, it was not difficult to assess the consequences on not reacting on the information due to the involved persons accumulated design experience of what can go wrong and being able to estimate consequences of bad decisions and bad designs.

Fifty-four (54) product-related dependencies were found as a result of the sessions where product-related dependencies were revealed and discussed. These would have had an impact on the project in terms of delays due to rework, lack of functionality, degraded performance of the product, quality issues of end product. It is impossible to evaluate or estimate if the dependencies would have been found in due time if it not had been for the questions asked. We will therefore argue that the sooner a dependency is spotted in a project the better the chances are to make the right decisions based upon the information thereby minimizing the risk of delays and degraded performance. Table 3 presents an overview of the product-related dependencies revealed by deploying the mindset via the questions. One example of the revealed dependencies from the table will be presented later in this section. We have refrained from further analysis of the number of occurrences of the product-related dependencies seen in Table 3 for simplistic reasons.

Table 3 Number of revealed dependencies, which would have had a significant impact had they not been addressed in the industrial project

Domain Theory categories	**Id #**	**Identified product-related dependencies**	**Number of revealed product-related dependencies**
Fu-Fu	1	Causal function	6
	2	State/time function	10
	3	Sync function	1
	4	Response function	2
Fu-M	5	Fu-M disposition	8
	6	Cumulative Fu-M	3
	7	Adverse effects	7
Pr-M	8	Property scheme	4
M-M	9	Multi-disciplinary means	3
	10	Volume allocation	3
	11	Liveliness	1
	12	Physical interface	3
	13	Communication interface	3

An example of the revealed dependency ‘Fu-M disposition’ is presented in the following: The grippers are moved by a custom made linear actuator driven by a brushless motor. When the user wants to grip an object a button is pressed and the grippers close to exert a predefined force on the object. If the user experiences that the force is not adequate, the button can be pressed once more and the actuator will be activated and driven to a new position where more force is exerted on the object. The exerted force is detected via the current in the motor and this concept seemed straight forward. However, by asking the questions about the needed *functions* on the next lower level as a consequence of choosing the *means*, it was revealed that the elasticity of the system plays an important role to the concept. If the gripped object is hard and the system has only little elasticity the increase in force by a predefined amount cannot be controlled. The reason being that the motor in a stiff system might only have to make two turns to increase the force two-fold. Such a small angular movement cannot be controlled accurately and the increase in force would fluctuate greatly, which would not be acceptable for the user. If we did not have identified this dependency between the mechanical system, the electronics and the control software, one of the following scenarios would have been likely to happen. A spring system would have to be incorporated in the already compact design where almost all space is taken up by other components, requiring a redesign. In a compact design such a design change would propagate through the design and affect a number the components and requiring the team to redo design fits and optimizations. If the time or cost constraints on the project would prevent the team from choosing the previous scenario the team might have been forced to abort this functionality of the product. Both scenarios of not having detected the dependency in time would have a significant impact on the project. This is just one dependency out of many which was revealed by asking questions about the 13 types of product-related dependencies. The dependencies represents potential delays in time schedule and degraded performance depending on the ingenuity of the engineers when they have to solve the problems. The later in the process you discover problems, which calls for changes in the design, the more likely it is to have cascading effects when the problems have to be solved, causing even more re-work. Today's market is extremely competitive and companies need every resource possible to focus on value-adding activities. Therefore it is of paramount importance to discover the dependencies before they become obstacles on the way to push the performance of the product that far that it can become a market success.

The *usability* of the 13 dependencies was addressed by formulating questions covering the product-related dependencies. They were introduced to the team on integration meetings and in separate sessions, where dependencies were revealed, by which the *applicability* criteria was justified. The *usefulness* was evaluated based estimations on what the consequences would have been if the dependencies would not have been identified. This was necessary because the mindset was applied on an industrial project and the team would have act on identified dependencies. The *usefulness* was based on estimations by professional design engineers, and thus principally prone to bias due to the evaluation process. This was the trade-off by testing it in an industrial setting and the possible bias seemed as a small sacrifice in comparison.

6 Conclusion

In this article we have focused on the dependency-aspects in designing mechatronic products, which is a challenging factor in the design process. Dependencies can contribute to make the design process being perceived as complex. However, it seems that if dependencies are consciously controlled and manipulated through the design process the perceived complexity of the task will be reduced. We have investigated a group of dependencies, which is related to the product-concept, and hence, called product-related dependencies. The *Domain Theory* showed generic groups of these dependencies, which would apply to all kind of products. The aim in the article was to reveal the type of product-related dependencies, which was of importance to the engineers when designing mechatronic products. For that, three cases from industry were investigated to reveal dependencies. Thirteen product-related dependencies had been identified being important to the design process. A combinatorial exercise between the attributes: *function*, *property* and *means*, also revealed two extra dependencies in *property-property* and *property-functions*. However, a direct relation between these attributes strives against the theory since the relation must be mediated through a *means*. This acknowledgment was supported by the findings from analysing the three project cases, where these dependencies could not be observed. Many discussions in the mechatronic case projects were about optimising *properties* and discussing what functionality the product should possess. The dependencies were discussed as *property-means-property* and *property-means-function* in terms of comparing product concepts. However, this is a different scope compared to the classification we are looking at. The thirteen classified dependencies all represent a direct dependency and can be viewed when analysing one product concept alone.

Our investigation shows that at least 13 product-related dependencies play a significant role in the mechatronic design process. The product-related dependencies have been identified, classified and characterised and they are in concordance with the fundamental design theory: *The Domain Theory*. It is assumed that the dependencies, if modelled and discussed adequately between engineers from the different engineering disciplines, will facilitate a better integration and positively affect the quality of the product and reduce the amount of rework in a project. The validation part of this research showed that positive effects could be obtained by applying this mind-set about the 13 dependencies in an industrial product development setting. Further work should be aimed at consolidating the classification of dependencies in terms of completeness and consistency by targeting a larger variety of mechatronic projects.

References

Adamsson N (2004) Model-based Development of Mechatronic Systems - Reducing the Gap Between Competencies? In: Tools and Methods of Competitive Engineering, vol 2. pp 405-413

Andreasen MM (1980) Machine Design Methods Based on Systematic Approach. PhD. thesis, Lund University, Sweden

Andreasen MM, McAloone TC (2001) "Joining Three Heads" - Experiences From Mechatronic Projects. In: 12th Design For X Symposium, Neukirchen. pp 151-156

Association of German Engineers (2004) VDI 2206, Design Methodology for Mechatronic Systems. VDI Guidelines. Beuth Verlag, Berlin

Blessing LTM, Chakrabarti A (2009) DRM, a Design Research Methodology. Springer, London

Bonnema GM (2011) Insight, Innovation, and the Big Picture in System Design. Systems Eng 14 (3):223-238. doi:http://dx.doi.org/10.1002/sys.20174

Borches PD, Bonnema GM (2010) A3 Architecture Overviews - Focussing Architectural Knowledge to Support Evolution of Complex Systems. Paper presented at the 20th Annual INCOSE Symposium (IS2010), Chicago, USA,

Braun SC, Lindemann U (2007) A Multilayer Approach for Early Cost Estimation of Mechatronical Products. In: International Conference on Engineering Design (ICED07). pp 187-188

Buur J (1990) A Theoretical Approach to Mechatronics Design. PhD. thesis, Technical University of Denmark, Denmark

Danilovic M, Browning TR (2007) Managing Complex Product Development Projects With Design Structure Matrices and Domain Mapping Matrices. International Journal of Project Management 25 (3):300-314

Gausemeier J, Flath M, Mohringer S (2001) Conceptual Design of Mechatronic Systems Supported by Semi-formal Specification. In: IEEE/ASME International Conference on Advanced Intelligent Mechatornics, Como, Italy. pp 888-892

Gausemeier J, Frank U, Donoth J, Kahl S (2009) Specification Technique for the Description of Self-optimizing Mechatronic Systems. Res Eng Des 20 (4):201–223

Gausemeier J, Low CY, Steffen D, Deyterl S (2008) Specifying the Principle Solution in Mechatronic Development Enterprises. In: 2nd Annual IEEE Systems Conference, Montreal, Canada. p 1

Hansen CT, Andreasen MM (2002) Two Approaches to Synthesis Based on the Domain Theory. In: Chakrabarti A (ed) Engineering Design Synthesis.

Hauser JR, Clausing D (1988) The House of Quality. Harvard Business Review:11

Isermann R (2005) Mechatronic Systems Fundamentals. Springer-Verlag, London

ISO/IEC 81346 (2012) ISO/IEC 81346 Standard. http://81346.com/english/.

Lindemann U, Maurer M, Braun TDI (2009) Structural complexity management : an approach for the field of product design. Springer, Berlin

Modelica Association (2010) Modelica Language Specification Version 3.2. http://www.modelica.org/documents/ModelicaSpec32.pdf.

Nagel RL, Stone RB, Hutcheson RS, McAdams DA, Donndelinger JA (2008) Function Design Framework (FDF): Integrated Process and Function Modeling for Complex Systems. In: ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2008, vol 43284. ASME, pp 273-286

Object Management Group (2010) OMG Systems Modeling Language Specification V1.2. http://www.omg.org/spec/SysML/1.2/PDF/.

Pahl G, Beitz W, Feldhusen J, Grote K (2007) Engineering Design-A Systematic Approach. Second edn. Springer-Verlag, London

Peters JF, Pedrycz W (2000) Software engineering : an engineering approach. John Wiley & Sons, New York ; Chichester

Pressman RS (2005) Software Engineering : A Practitioner's Approach. International ed., 6th ed. edn. McGraw-Hill Higher Education, Boston, Mass. ; London

Sage AP, Rouse W (2009) Handbook of Systems Engineering and Management. Second edn. John Wiley & Sons, Oxford

Salminen V, Verho AJ (1989) Multi-disciplinary Design Problems in Mechatronics and Some Suggestions to Its Methodical Solution in Conceptual Design Phase. In: International Conference on Engineering Design (ICED89). pp 533-554

Shah AA, Kerzhner AA, Schaefer D, Paredis CJJ (2010) Multi-View Modeling to Support Embedded Systems Engineering in SysML. Graph Transformations and Model Driven Engineering- Lecture Notes in Computer Science 5765/2010:580-601. doi:10.1007/978-3-642-17322

Siam (1996) Inquiry Board Traces Ariane 5 Failure to Overflow Error. Siam News. http://www.cs.clemson.edu/~steve/Spiro/arianesiam.htm.

Suh NP (2006) Applications of Axiomatic Design to Engineering Collaborations and Negotiation. In: 4th International Conference on Axiomatic Design (ICAD).

Systems D (2011) Dymola - Multi Engineering Modeling and Simulation. http://www.3ds.com/products/catia/portfolio/dymola,

Tomiyama T, Amelio VD, Urbanic J, ElMaraghy W (2007) Complexity of Multi-Disciplinary Design. Annals of the CIRP 56 (1):185-188

Torry-Smith J, Qamar A, Achiche S, Wikander J, Mortensen NH, During C (2012) Challenges in Designing Mechatronic Systems. Journal of Mechanical Design

Torry-Smith JM, Qamar A, Achiche S, Wikander J, Mortensen NH, During C (2011) Mechatronic Design - Still a Considerable Challenge. In: ASME 2011 Design Engineering Technical Conferences & Computers & Information in Engineering Conference, Washington, USA, vol 54860. pp 33-44

Woestenenk K, Tragter H, Bonnema GM, Cabrera AAA, Tomiyama T (2010) Multi Domain Design: Integration and Reuse. In: ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2010) Montreal, Canada, vol 44137. ASME, pp 519-528

11.5 Paper E

"The Mechatronic Integration Concept"

Submitted to the International Journal of Industrial Engineering: Theory, Applications and Practice, 2012

THE MECHATRONIC INTEGRATION CONCEPT

Jonas Mørkeberg Torry-Smith[1], Niels Henrik Mortensen[1], Sofiane Achiche[2], Ole Ploug[3]

[1]Department of mechanical engineering
Engineering Design and Product Development Section
Technical University of Denmark
2800, Kgs. Lyngby, Denmark
jtor@mek.dtu.dk, nhmo@mek.dtu.dk

[2]Department of mechanical engineering
Ecole Polytechnique de Montréal, Université de Montréal
C.P. 6079, succ. Centre-ville, Mtl, H3C 3A7, Qc, Canada
sofiane.achiche@polymtl.ca

[3]R&D Manager, Svinget 5, 6230 Rødekro, Denmark
Ole_ploug@hotmail.com

Design of mechatronics is greatly challenging due to its multi-disciplinary nature. To be able to stay competitive as a company we need to utilize the potential synergy between the engineering disciplines involved in the project, and we need to be able to address the dependencies created in the product as a consequence of the collaboration between the involved engineering disciplines. This paper proposes a way to identify, model and clarify dependencies by use of a *"Mechatronic Integration Concept"*. Literature on the topic of modeling dependencies lack proposals for how to model the dependencies explicitly in a mechatronic system, which is visually intriguing and can be a base for cross-domain discussions. The usefulness of the *Mechatronic Integration Concept* has been tested in an industrial development project showing positive results of shortening the lead-time, minimizing rework and increasing the performance of the product.

Significance: Many companies involved in development of mechatronics have difficulties in handling the multi-disciplinarity of the design task. This paper suggests how to identify, model and clarify dependencies created in the product.

Key words: Dependency modeling, mechatronic concept, design of mechatronics

(Received: Accepted:)

1. INTRODUCTION

Development of mechatronics is a challenging task to undertake. Nonetheless, the necessity of combining solutions from the areas of mechanical, electronics and software engineering is often what drives companies to accept a higher level of complexity in the development process. Having accepted a higher level of complexity as a consequence of the multi-disciplinary design efforts, companies strive to fight back this exact complexity for a better overview of the design task. Some of the challenges companies encounter include: undesirable and unexpected dependencies are discovered too late in the design process (D'Amelio and Tomiyama 2007), product *properties* are dispersed onto different modules handled by different engineering disciplines without proper control and tracking, and the nature of the development within the involved engineering disciplines is different causing synchronization problems between the domains in terms of deliverables. Often the desired level of integration between engineers from different engineering disciplines is not achieved which contributes to the aforementioned challenges. The research work presented in this paper aims at providing a means for modeling and representing a mechatronic concept in which dependencies in the product between domains can be modeled and clarified and hence mitigate the above-cited challenges.

We assume that the process of modeling an 'integration concept' along with modeling important dependencies in the mechatronic design will drag the attention of the designers toward integration issues. To be able to identify and clarify dependencies, a shared understanding of the design is needed. To gain full advantage of a shared understanding the knowledge must be made *explicit*. Nonaka (1994) describes the transition from *tacit knowledge* to *explicit knowledge* as a necessary action in obtaining new insight for the team of persons collaborating across boundaries. Many good suggestions from researchers can be found on how to describe a mechatronic concept. Some suggestions aim at describing the concept seen from a holistic point of view (Buur 1990), whereas other suggestions are aimed at specific applications such as control

engineering (Gausemeier et al. 2009a). In this paper, we are interested in how a mechatronic concept should be modeled when focusing on the integration issue between the domains. The research question we are trying to answer is: how can we elucidate and clarify dependencies in a mechatronic concept? The term 'dependency' is addressing relations in the product concept which forces restrictions on the design between the engineering disciplines. Facilitating integration between domains will have the advantage of shorter lead time due to reducing rework and optimized performance and increased quality of the end product (Andreasen and Hein 1987).
The outline of the paper is as follows: first the research method is described in section 2 followed by a description of related work in section 3; the content of the *Mechatronic Integration Concept* is explained in section 4 and the results of testing the concept in an industrial setting are in section 5; discussion and conclusion are found in section 6 and 7 respectively.

2. RESEARCH METHOD

Three mechatronic development projects aimed at the consumer product segment were thoroughly investigated and documented; we observed and recorded trends of how the mechatronic product concepts were modeled to facilitate collaboration between the engineers from different domains. The second step was to verify to what extent the insights from the investigated projects were in concordance with the theory for mechatronic design. A detailed description of types of dependencies central to mechatronics projects was obtained from recent work carried out by the authors (Torry-Smith et al. 2012a). The concept description and the overview of dependencies were then combined into a proposal for how to clarify and model dependencies in mechatronic concepts. Related work was reported on to position the research to the literature. The proposal was then tested in a context of an industrial project and the results evaluated in terms of its usefulness and impact on the design activities.

3. RELATED WORK

The aim of the section is to report on prior work, in which dependencies are modeled explicitly in mechatronic design. The search for related work is performed by focusing on two aspects of design of mechatronics. The first aspect is how to describe a mechatronic product concept and the second aspect is how to describe and model dependencies found in mechatronics projects.
Many researchers proposed different frameworks for mechatronics design. Buur (1990) proposes a theoretical framework for design of mechatronics building on the Domain Theory by Andreasen (1980). Within this framework he suggests to model interface 'organs' defining the boundary between the domains, i.e. dependencies. However, further definition of these dependencies is not stated. Other modeling suggestions based on functional reasoning include Contact and Channel Approach by Albers et al. (2011) and FDF by Nagel et al. (2008). Though dependencies can be modeled using these models they are limited to describing the flow of energy, material and signal between functions or functional carriers.
Gausemeier et al. (2009a) focus on control engineering when proposing a definition of a mechatronic concept. A number of views are suggested to be modeled and one of these views is called 'active structure' and relates to the control issue describing exchanged signals and energy between components. Even though interfaces (Mohringer and Gausemeier 2002) and consistency management is described (Gausemeier et al. 2007) this research is on how one can link holistic model with domain-specific model but it is not providing an overview of what to link. The work by Gausemeier is related to research on transformation models and formal description languages. Alternative formal modeling languages include SysML (Object Management Group 2010), UML (Group 2011) and IDEF (Integration Definition Methods 2012) and AM-tool (Cabrera et al. 2011). Common for the formal languages is that they provide the possibility to model aspects of a mechatronic concept as well as dependencies due to the flexibility of the semantics provided. Though, no systematic reporting on types of dependencies is found. Research work on transformation models presents the possibility to have one meta-model which shares parameters with domain-specific modeling to maintain consistency across domains (Gausemeier et al. 2009b; Wynn et al. 2009; Shah et al. 2010) . This provides a framework for managing dependencies but doesn't define any dependencies to model. Instead of using a meta-model for consistency checking Hehenberger (Hehenberger et al. 2010) suggests to perform automatic consistency checking when design parameters are changed. It provides some advantages but it requires the concept to be describes via a formal model (e.g. a SysML model). This work is not aimed at providing overviews of dependencies but to suggest a method for automatically handling and checking inconsistencies (which can be interpreted as dependencies).
An alternative to formal modeling and use of meta-models is the use of informal modeling of mechatronic concepts such as A3 architecture overviews (Borches and Bonnema 2010) and sketching techniques by e.g. Buur (Buur 1990). These methods provide flexibility in describing both the concept and the dependencies. While the flexibility is the strength of these

methods it is also the weakness. Only general descriptions of the content of the models are provided, thereby not touching upon descriptions of dependencies and how to model them in the concept.

Design Structure Matrix (DSM) is directly aimed at modeling dependencies in products (Felgen et al. 2005). DSM is characterized by the ability of comparing two-of-a-kind, e.g. describing which components is physically interfacing other components. Domain Mapping Matrices (DMM) (Danilovic and Browning 2007) is also aimed at describing dependencies but are capable of comparing two different entities, which could be e.g. relating functions to components. The analyzed dependencies related to the product are limited to the relations between functions and components only. A further distinction between types of dependencies is not found within the methodology. The advantage of using DSM is the possibility to apply algorithms to rearrange rows and columns. Even though applicable algorithms are far more limited when using DMM, matrix representation of the dependencies is a fundamental prerequisite. Due to the focus on matrix representation, graphical representations or other visual models of the concept are lacking.

The goal of this review was to find related work on how to explicitly model dependencies in mechatronic concepts with the aim of facilitating a better integration between the domains. The tools and methods presented above are primarily aimed at either describing a mechatronic concept or modeling dependencies explicitly. To the authors' knowledge, a combination of the two aspects and a more elaborate description of dependencies to look for do not seem to be covered in prior work.

4. THE DESCRIPTION OF THE MECHATRONIC INTEGRATION CONCEPT

From investigating three mechatronic projects from Danish industry we observe that three views are used for cross-disciplinary integration meetings: (1) A common conceptual descriptions based on a functional description; (2) a view dealing with spatial relations between components, physical forces and other physical effects; and (3) a view dealing with signal processing and data processing. The views are labeled the M/E/Sw-view the M/E-view and the E/Sw-view respectively. Having established the views the literature is consulted. It appears that these views are in concordance with mechatronics theory presented by Jansen (2007) and Tomiyama et al. (2007). Tomiyama states that a view bridging two domains can only be possible if the two domains share the same axioms. To exemplify this statement the M/E view is possible to model because they share axioms when viewing the system with regard to spatial relations or physical effects such as forces. Software does not have axioms tied to spatial relations and, hence, cannot be modeled in that view. Table 1 is showing the content to be modeled for each of the three views obtained from investigating the three cases. Since the three views are central to create cross-domain discussions we propose that they be a part of the *Mechatronic Integration Concept*.

M/E/Sw view – Functional description	**M/E view – Physical structure and spatial arrangement**	**E/Sw view – Data structure and signal processing**
Aspects to cover in the M/E/SW view • Task analysis for life phases • Functions and function carriers • Sequence of the functions	Aspects to cover in the M/E view • Spatial configuration • Connectivity between components • Force and physical effects	Aspects to cover in the E/SW view • Data and signal flow • Data structure (architecture) • Timing and sequencing
Suggestion for models to describe the view • Life phase scenarios • Task flow diagram with description of the technical process performed at each step including sensors and actuators involved • Function/Means tree • Finite State Machine diagram	Suggestion for models to describe the view • Spatial drawing or sketch of products outer shape and MMI elements • Spatial drawing or sketch of the main components/means • Overview of force distribution in product or critical loads on structure. • Interface diagram containing main components and their interfaces	Suggestion for models to describe the view • Use case diagram • Data Flow Diagram showing main components and the data/signals transferred • Data structure diagram defining the architecture • Critical executable blocks modeled with pseudo code

Tabel 1: The content of the M/E/Sw-view, the M/E-view and the E/Sw view

In addition to the three views presented in Table 1, we propose to include an overview of important dependencies identified in the product concept, so that the *Mechatronic Integration Concept* will be composed of these four descriptions. In the following, a description of generic dependencies is presented. From previous work, we have classified a number of dependencies, which can be grouped according to *Systems Theory* (Hubka and Eder 1988). By doing so the following groups emerge (Torry-Smith et al. 2012a):

- *A Function-Function dependency:* A dependency between two *functions* is described by the link that is created when a *function* reacts to a stimulus created by another *function*.
- *A Means-Means dependency:* A dependency between two *means* in the product.
- *A Function-Means dependency:* A *function* is realized by a *means* and a *means* can be further de-composed into *sub-functions*, which creates the dependency between *functions* and *means*.
- *A Property-Means dependency: Properties* are realized by *means*, thereby creating dependencies between *means* and *properties*.

These dependencies are all related to one product concept, meaning that the dependencies can be revealed when viewing one product concept alone. An example of a dependency between *means* is the physical interfaces between two components. The relations *Function-Property* and *Property-Property* cannot be found as direct relations in products because the relation will have to go through a *means*. An evaluation of a *property* will include an evaluation of the *means* to which the *property* belongs. This causes the *means* to be included in the relation thereby making it impossible to observe a direct relation between two *properties* or between a *property* and a *function*.

In previous work by the authors (Torry-Smith et al. 2012b) the generic categories presented above can be further classified into 13 groups of dependencies specifically directed at mechatronic products. The classification into the 13 groups has the advantage of offering a guideline to what to look for and hence making the identification of central dependencies more tangible. The identified groups of dependencies are described in Table 2.

Categories	Id #	Name of dependency	Description of the dependency
Fu-Fu	1	Causal function	The dependency between *functions* when the functionality of the product is considered as a process flow
	2	State/time function	Dynamic dependencies between *functions*, in which the sequence and the timing is important.
	3	Sync function	Ensuring that the states of the product are synchronized in all domains.
	4	Response function	*Functions* react on stimuli from other *functions*. The size and type of the stimuli have to be matched between the *functions*.
Fu-M	5	Fu-M disposition	Proposing *means* to *functions* in one domain will often have consequences in other domains in terms of supporting functionality.
	6	Cumulative Fu-M	The realization of a *function* may require *means* from various disciplines.
	7	Adverse effect	A *means* may have an adverse effect associated to it. The undesired adverse effect can be formulated as a function (e.g. 'create vibration').
Pr-M	8	Property scheme	The realization of a *property* may be distributed on several components designed by different engineering disciplines.
M-M	9	Multi-disciplinary means	Some *means* have to satisfy boundary conditions (e.g. requirements), which are important to more than one engineering discipline.
	10	Volume allocation	Physical *means* have to be located spatially in the product and the volume may have changing restrictions during the life phases.
	11	Liveliness	The flow of information between electronics and software must be designed without causing a system-lock.
	12	Physical interface	Physical interfaces between modules and components have stakeholders from electronics and mechanical engineering.
	13	Communication interface	Communication between components whether they are analogue or digital is a dependency between electronics and software engineering.

Table 2: Overview of the classification of dependencies into 13 groups specific to mechatronic development

A description comprising the three views (M/E/Sw-view, M/E-view and E/Sw-view) and the overview of dependencies constitute the integration concept. A modeled example of the integration concept is illustrated in the case study in section 5. By combining the three modeled views and the overview of product-related dependencies we are able to make a unified description which can bridge the domains. The modeling and the focus on dependencies will be the integration catalyst revealing potential challenges before they become problems as well as promoting synergistic solution-finding between the domains.

5. INDUSTRY APPLICATION - ‘THE STRONG HAND’ CASE

The usefulness and the obtained results of applying the *Mechatronic Integration Concept* in an industrial project are reported on in this section. First the project is briefly described followed by a presentation of how we modeled the *Mechatronic Integration Concept* comprising the identified dependencies. Then three examples are illustrated of how dependencies were treated in the project and the effects of being able to model and clarify them. The modeling of one of these dependencies is shown for that particular example.
The project selected for the testing was about developing an actuated hand for patients with severe arthritis. The mechatronic hand can be fitted inside their own palm to help provide an enhanced grip. The device can be attached and removed as the user wishes. The aim is to make the device appear discrete when worn at home or in public places. Figure 1a served as visualization of the product at the beginning of the project.

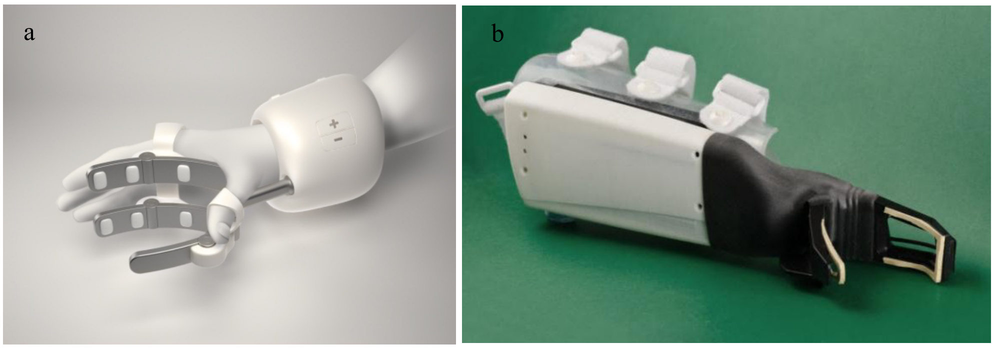


Figure 1: a) Computer rendering of the product concept b) The functional model of the product concept (fully operational)

The project set-up was a joint venture between several companies comprising engineers representing the mechanical, the electronics and the software domain. The project team also included user experience experts, industrial designers and a board of practitioners (arthritis specialists) in addition to the project management group.
The *Mechatronic Integration Concept* was deployed in the conceptual phase where the overall functionality had been determined and solutions in terms of suggestions for technology building blocks have been proposed. Only a rough modeling of the concept has been performed in CAD showing an outline of the subassemblies. A suggestion of the *Man Machine Interface* was proposed but not tested by users so far. Figure 2 is a sketch of the concept (document from the project) illustrating the clarification level. Figure 1b shows the functional model three months later which is fully operational. The project being half-way through the concept development phase creates a purposeful option to test the usefulness of being able to model the *Mechatronic Integration Concept*.

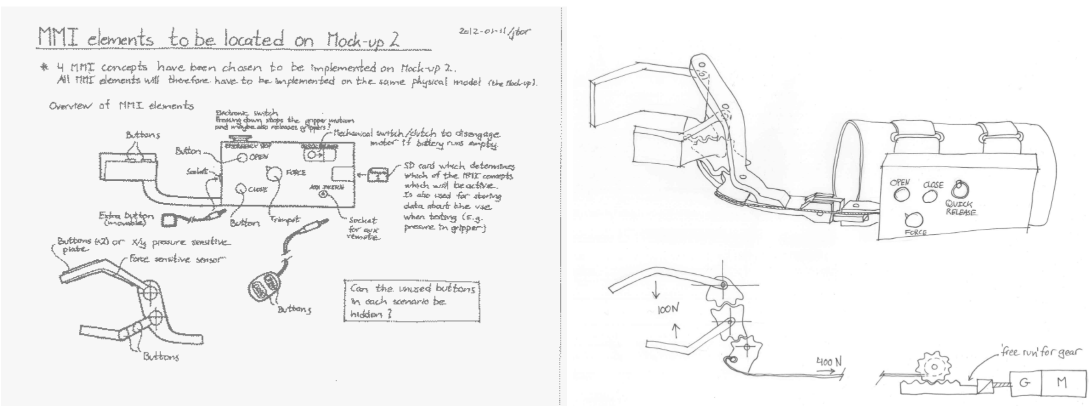


Figure 2: Sketches of the product concept

By using the classification of dependencies from Table 2 we are able to identify 54 central dependencies in the product concept, which have to be addressed in the project. To facilitate the clarification and further handling of the dependencies the *Mechatronic Integration Concept* is modeled (see Figure 3). The functionality of the product is modeled via a task flow analysis in which the technical process including the involved sensors and actuators are described for each step. This description is broken down into two functional descriptions: 1) A functional description where *functions* are related to the principle solutions and 2) A functional overview describing which functionality is active depending on the state of the product. Based on the functional description the M/E-view and the E/Sw-view are created. The M/E-view contains descriptions of the spatial arrangements of the modules and components in the product as well as central physical effects considerations, which are force calculations for the electro-mechanical transmission. In the E/Sw-view we choose to model the data and signal flow in a Data Flow Diagram. In addition we model the data architecture by defining the hierarchy and interaction between main modules of the software.

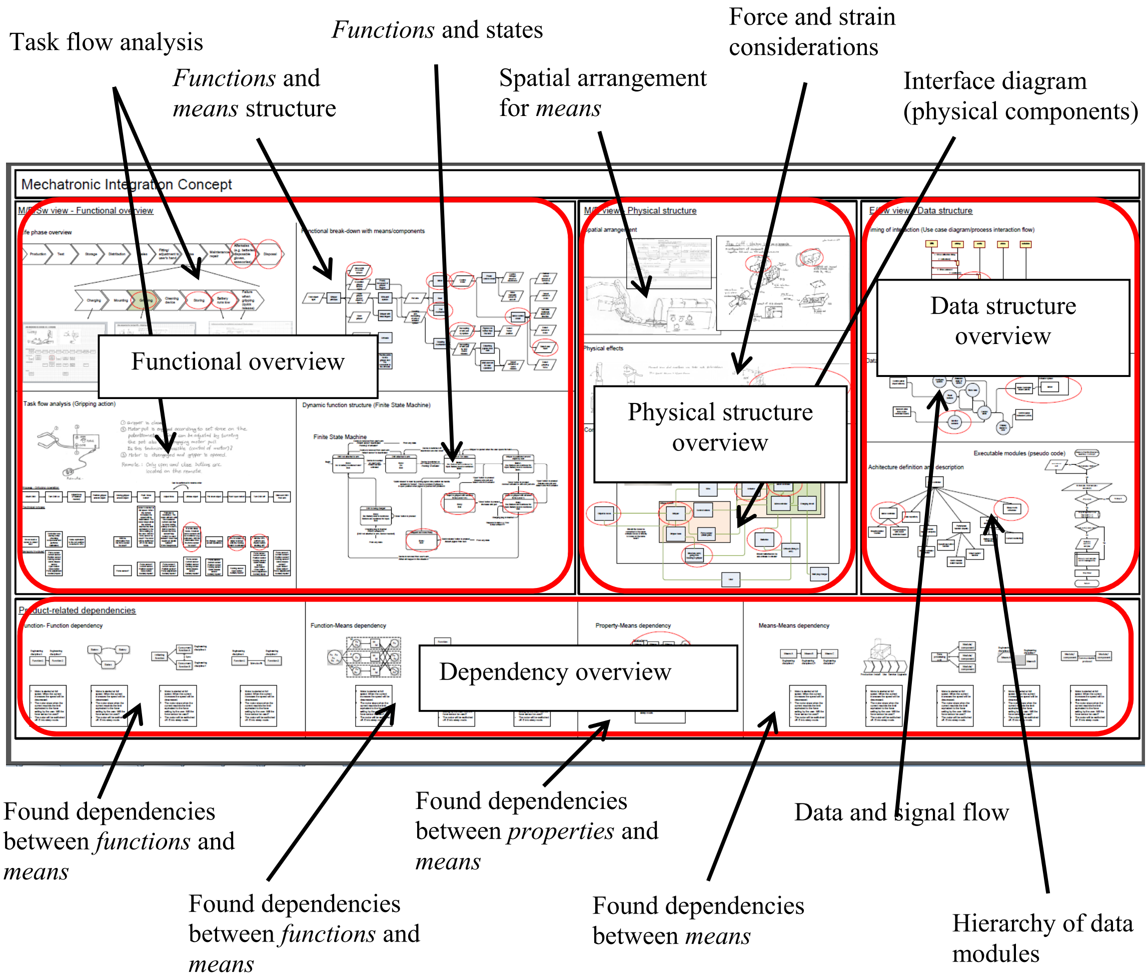


Figure 3: The content of the *Mechatronic Integration Concept*

The found dependencies are grouped according to the 13 categories, and the four main categories from Table 2. There are also dependencies within each engineering discipline such as an interface between two mechanical parts. However, these are not modeled since the scope is the cross-disciplinary dependencies. The dependencies may be identified as short statements, but it is likely that they are not complete or fully clarified. As each dependency reaches across at least two engineering disciplines, the presence of representatives for each discipline is needed to achieve a clarification. In this process we transcend from tacit knowledge to explicit knowledge as stated by Nonaka (1994). Hence, it allows us to check if assumptions of the effect of design decision across engineering disciplines are correct.
We use the poster as presented in Figure 3 to facilitate the discussion among representatives from each domain and to keep track of the created explicit knowledge. Due to the scope and length limitations of this paper we cannot describe all of the found dependencies. Instead we have selected three dependencies to exemplify the potential of clarifying the product-related dependencies. They have been selected based on the criteria to be fairly explicable while still showing the complexity of the dependencies. In addition, an example is given in the section labeled II for how to model a dependency.

I. The first dependency was identified when investigating principle solutions to the functionality of "holding on to an object after gripping". The role of the motor was revealed as a (multi-disciplinary) dependency between the electronics and the mechanical domain. The solution seemed fairly straight forward. When the grip is tightened the current in the motor will rise and when the current has reached a limit, the current is maintained at this level to ensure the firm grip on the object.

When this was discussed with the electronic engineer, the solution was not as straight forward as expected. A steady current cannot be maintained in the coil of the motor. The current will have to either increase or decrease if the motor is not turning. It could be done, but it would require advanced electronics which would 'trick' the motor to believe it was supplied with a steady current. Considering the dependency after having revealed the aspects, it was chosen to find a different solution to secure a firm grip of the object. If this dependency had not been discovered in time various scenarios could have happened: 1) Late changes of the principle solution which would require adaption and re-work of the product to accommodate the change, 2) Incorporating the more advanced and more expensive solution in the electronics domain, which again might require extra space causing adaptions of the design in the mechanical domain. In addition to these undesirable situation either of the scenarios would inflict re-work causing higher development cost and an increased development time.

II. The next example to illustrate the potential of working with dependencies is one that is linked to the battery life time. The *battery life time* is a *property* identified to have contributing elements in the mechanical, electronics and software domain. Due to the link between the *means* contributing to the *property*, the dependency is of the type: Property scheme dependency (according to Table 2). The device is powered by a battery located on the device and *battery life time* is a central concern. Firstly, the obvious components affecting the *battery life time* is identified: battery size (capacity and technology); power consumption of motor and power for the electronics. When these are broken down further and new aspects are discovered, the picture is far from simple. The stiffness of the structure and the mechanical advantage of the system play an important role. The type of battery technology (e.g. polymer-Ion) is linked to the capacity and the capacity can vary over time as a consequence of how the charging is performed and monitored. In the software domain sleep modes can be introduced but should be carefully designed to allow for monitoring of user inputs in-between sleep modes. These are just some of the *means* which influence the *battery life time*. The modeling process begins by identifying and highlighting the contributing elements. This is simply done by circling the elements in all three views which will or may influence the *property* (see Figure 4). One by one the influence of the elements is discussed between the involved stakeholders and thereby clarified at a meeting where representatives from each domain are present. The task for the involved engineers is to figure out how the target specification can be met and what *means* to optimize to use the allocated resources most ideal not to inflict the development time negatively.

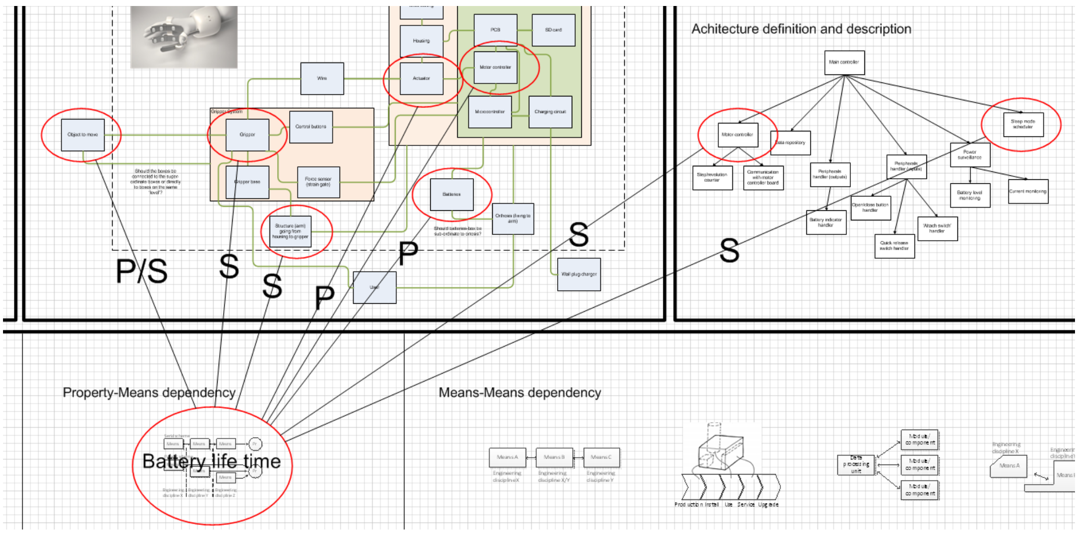


Figure 4: Modeling the dependency linked to the *property* '*battery life time'* (Section of the *Mechatronic Integration Concept*)

Means can contribute in either a serial or in a parallel manner. The physical size of the battery will have the character of contributing in a serial manner, meaning that the size of the battery is almost proportional with the capacity, which will extend the *battery life time* proportionally. Sleep modes will have an effect on the drain of the battery but the effect is not as straight forward as the battery capacity and will e.g. be influenced by the ratio between the device gripping and the device being inactive. During the clarification process the type of contribution scheme (parallel or serial) is marked in the views as can be seen in Figure 4). Having modeled the dependency in detail, decisions on how to treat it further in the project can be made on a sound basis. If the relation between the contributing factors is not clarified, the risk of sub-optimization is immanent and resources will be wasted as well as unnecessary design changes being made in the name of 'optimization' (perform the optimization blindly). Resources allocated to a project are always limited and therefore it is very important that we use them in the best possible way.

III. The last example is about how to adjust the gripping force after the user has gripped an object. As described in the first example the electronics and software measures the current in the DC motor. This measurement is an indirect measurement of the gripping force. Prior to gripping, the user can adjust the gripping force by turning a knob on the device. If the user experience the grip is not firm enough after gripping the user can increase it. If the force has to be increased the motor is activated until a higher threshold limit for the current has been reached. Adjusting the force up or down after gripping represents a dependency since solutions from every domain is required to realize the functionality (Fu-M cumulative dependency). The effort has to be coordinated and the dependency has to be understood by all involved domains. When discussing this functionality of the device, a concern arose: increasing the force would most likely be feasible but decreasing the force would be a problem. The reason being that the current will not reflect (be proportional to) the gripping force when decreasing the force. Due to the gripping in action there is a pull in the actuator. Thereby the actuator can reduce the gripping force by a very small reverse current since the tension in the system is helping the movements of the system decreasing the gripping force. After having revealed the dependency the team could choose between two options: i) remove the functionality of reducing the gripping force after gripping, ii) solve the functionality by use of other means. One possibility, which is discussed among the engineers is to count the pulses to the motor when decreasing the force and then based on experiments assess how many steps the motor should reverse to obtain a certain decrease in gripping force. The decision in the project was to set the functionality on hold, wait for the functional model in order to perform tests to see if 'counting steps' is sufficient to control the decrease in gripping force. By revealing the dependencies at an early stage, the team or the project manager is able to make the decision up front of removing the functionality or allocate resources to find an alternative solution. The result is improved project planning, better use of resources and enhanced monitoring of the predicted performance of the product.

The *Mechatronic Integration Concept* aided in clarifying the dependencies. The application in the industrial setting showed that is possible to model the dependencies explicitly and that the *Mechatronic Integration Concept* can facilitate a cross-disciplinary discussion. The result from applying the concept in the project was a potential cut in the lead-time, increased efficiency of resources used thereby pointing in the direction of being able to help reduce costs in a development project.

6. DISCUSSION

The discussion section will be used for reflections on our experience by using the *Mechatronic Integration Concept*. One could argue that you will never be sure whether all relevant dependencies have been identified. It is true. Yet, as in the creative solution-finding phase of a project you are not certain if all relevant solutions have been found. To mitigate this situation we utilize structured methods in addition to the creative sessions. Similarly we support the likeliness of finding relevant dependencies by providing a classification of types of dependencies and a way to model them in *Mechatronic Integration Concept.* All identified dependencies might not have to be modeled in detail as illustrated in Figure 4. As in every project the ability to be agile in managing the encountered challenges is key to success. The *Mechatronic Integration Concept* provides the basis to make the discussion of each dependency tangible. Having obtained an overview of the dependencies which have to be managed, it can be decided which of the dependencies should be modeled in detail. The reasons could be many. Some of them could be if a dependency is perceived as being complex or dependencies which are linked to high uncertainty and great risk of having an impact on the product's performance. Once the dependencies have been identified and clarified to the extent that the team understands the different aspects of it, the team has to decide how to manage the dependency. The three basic choices are: either to remove the dependency by a re-design or to manipulate the dependency via a re-design to reach a situation where the integration is manageable. Finally there is the option of accepting the dependency 'as is' and then monitor it when continuing the design process.

7. CONCLUSION

The main originality in this article is the introduction of the *Mechatronic Integration Concept* in which central dependencies can be modeled explicitly. The mechatronic concept ensures a common and shared understanding of the product concept with its inherent dependencies. The structured cross-boundary clarification of the dependencies enables the team to resolve integration issues early on in the project, which has many positive effects on the product development process. When applying the *Mechatronic Integration Concept* in an industrial project we observe strong indications of effects comprising: reduced lead-time and better utilization of resources due to avoidance of re-work as well as increased performance of the product. The benefit for companies in the long run from having the dependencies under control is that it enables them to run a concurrent process in which the mechanics, electronics and the software are aligned. The advantages of a concurrent process include even shorter lead-times and an increased potential for innovative solutions due to the achieved integration.

8. REFERENCES

1. Albers A, Braun A, Sadowski E, Wynn DC, Wyatt DF, Clarkson PJ (2011) System Architecture Modeling in a Software Tool Based on the Contact and Channel Approach (C&C-A). Journal of Mechanical Design 133 (10):101006-101008

2. Andreasen MM (1980) Machine Design Methods Based on Systematic Approach. PhD. thesis, Lund University, Sweden

3. Andreasen MM, Hein L (1987) Integrated product development. IFS, Bedford

4. Borches PD, Bonnema GM (2010) A3 Architecture Overviews - Focussing Architectural Knowledge to Support Evolution of Complex Systems. Paper presented at the 20th Annual INCOSE Symposium (IS2010), Chicago, USA,

5. Buur J (1990) A Theoretical Approach to Mechatronics Design. PhD. thesis, Technical University of Denmark, Denmark

6. Cabrera AAA, Woestenenk K, Tomiyama T (2011) An Architecture Model to Support Cooperative Design for Mechatronic Products: A Control Design Case. Mechatronics 21 (3):534-547

7. D'Amelio V, Tomiyama T Predicting the unpredictable problems in mechatronics design. In: International Conference on Engineering Design (ICED'07), 2007. p DS42_P_153

8. Danilovic M, Browning TR (2007) Managing Complex Product Development Projects With Design Structure Matrices and Domain Mapping Matrices. International Journal of Project Management 25 (3):300-314

9. Felgen L, Deubzer F, Lindemann U Complexity Management During The Analysis of Mechatronic Systems. In: International Conference on Engineering Design (ICED'05), 2005. p 409

10. Gausemeier J, Frank U, Donoth J, Kahl S (2009a) Specification Technique for the Description of Self-optimizing Mechatronic Systems. Res Eng Des 20 (4):201–223

11. Gausemeier J, Giese H, Schäfer W, Axenath B, Frank U, Henkler S, Pook S, Tichy M (2007) Towards the design of self-optimizing mechatronic systems: Consistency between domain-spanning and domain-specific models.

12. Gausemeier J, Schafer W, Greenyer J, Kahl S, Pook S, Rieke J Management of Cross Domain Model Consistency During the Development of Advanced Mechatronic Systems. In: International Conference on Engineering Design (ICED'09), 2009b. pp 1-12

13. Group OM (2011) OMG Unified Modeling Language (UML) Specification V2.4.1. http://www.omg.org/spec/UML/2.4.1/.

14. Hehenberger P, Egyed A, Zeman K Consistency checking of mechatronic design models. In: Proceedings of the 2010 ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2010, Montreal, Quebec, Canada, 2010.

15. Hubka V, Eder WE (1988) Theory of Technical Systems: A Total Concept Theory for Engineering Design. Springer Verlag,

16. Integration Definition Methods (2012) IDEF. Integration Definition Methods. www.idef.com.

17. Jansen S (2007) Eine Methodik zur modellbasierten Partitionierung mechatronischer Systeme. Ruhr-Universität Bochum, Achen

18. Mohringer S, Gausemeier J An Interface Specification for Principle Solutions Supporting the Cross-Domain Design of Mechatronic Systems. In: PROCEEDINGS OF THE INTERNATIONAL DESIGN CONFERENCE, 2002. vol 7. pp 533-538

19. Nagel RL, Vucovich JP, Stone RB, McAdams DA (2008) A Signal Grammar to Guide Functional Modeling of Electromechanical Products. Journal of Mechanical Design 130 (5):051101-051110

20. Nonaka I (1994) A Dynamic Theory of Organizational Knowledge Creation. Organ Sci 5 (1):14-37

21. Object Management Group (2010) OMG Systems Modeling Language Specification V1.2. http://www.omg.org/spec/SysML/1.2/PDF/.

22. Shah AA, Kerzhner AA, Schaefer D, Paredis CJJ (2010) Multi-View Modeling to Support Embedded Systems Engineering in SysML. Graph Transformations and Model Driven Engineering- Lecture Notes in Computer Science 5765/2010:580-601. doi:10.1007/978-3-642-17322

23. Tomiyama T, Amelio VD, Urbanic J, ElMaraghy W (2007) Complexity of Multi-Disciplinary Design. Annals of the CIRP 56 (1):185-188

24. Torry-Smith JM, Mortensen NH, Achiche S (2012a) Classification of product-related dependencies in development of mechatronic products. Res Eng Des:(under review, submitted 2012)

25. Torry-Smith JM, Qamar A, Achiche S, Wikander J, Mortensen NH, During C (2012b) Challenges in Designing Mechatronic Systems. Journal of Mechanical Design

26. Wynn DC, Nair SMT, Clarkson PJ The P3 platform: an approach and software for developing diagrammatic model-based methods in design research. In: International Conference on Engineering Design (ICED'09), Stanford, USA, 2009.

DTU Mechanical Engineering
Section of Engineering Design and Product Development
Technical University of Denmark

Produktionstorvet, Bld. 426
DK- 2800 Kgs. Lyngby
Denmark
Phone (+45) 4525 6263
Fax (+45) 4593 1577
ISBN: 978-87-7475-366-7
www.mek.dtu.dk

DCAMM
Danish Center for Applied Mathematics and Mechanics

Nils Koppels Allé, Bld. 404
DK-2800 Kgs. Lyngby
Denmark
Phone (+45) 4525 4250
Fax (+45) 4593 1475
www.dcamm.dk
ISSN: 0903-1685