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 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

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.
- 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, Robotics, 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,” “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 a P or a Y. 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 many 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
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. Hegenberger 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

<table>
<thead>
<tr>
<th>Type</th>
<th>Challenges</th>
<th>Researchers/Research Groups</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Product</td>
<td>A Lack of a common understanding of the overall system design</td>
<td>X X X X X X X X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>B Difficulty in assessing the consequences of choosing between two alternatives</td>
<td>X X X X X X X X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C Lack of a common language to represent a concept</td>
<td>X X X X X X X X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>D Modeling and controlling multiple relations in the product concept</td>
<td>X X X X X X X X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>E Being in control of the multiple functional states of the product</td>
<td>X X X X X X X X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>F Transfer of models and information between domains (expert groups)</td>
<td>X X X X X X X X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity</td>
<td>G Synchronizing development activities to attain concurrent engineering</td>
<td>X X X X X X X X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mindset</td>
<td>H Different tradition within the domains for how to conduct creative sessions</td>
<td>X X X X X X X X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>I Reluctant to interact with engineers from other disciplines</td>
<td>X X X X X X X X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>J Different mental models of the system, task and design-related phenomena</td>
<td>X X X X X X X X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Competence</td>
<td>K Lack of common language to discuss freely at creative meetings</td>
<td>X X X X X X X X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>L Education within disciplines do not call for integration in professional life</td>
<td>X X X X X X X X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>M The nature of design is different</td>
<td>X X X X X X X X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organizational aspects</td>
<td>N Product complexity affects the organization complexity</td>
<td>X X X X X X X X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>O Knowledge transfer between domains is inadequate (even in cross-disciplinary teams)</td>
<td>X X X X X X X X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other aspects</td>
<td>P Lack of a broadly accepted methodology</td>
<td>X X X X X X X X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Q Mechatronic ownership is lacking</td>
<td>X X X X X X X X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>R System engineers are lacking detailed information of the system</td>
<td>X X X X X X X X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>S Complexity as a generic problem</td>
<td>X X X X X X X X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Research groups.
*Nagel is part of the Wood group, hence [9].
*Shah is part of the Paredis group, hence [10].
(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, Borchers 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 automatic 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 (B). Hence, it 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.
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.

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)

<table>
<thead>
<tr>
<th>#</th>
<th>Solutions</th>
<th>Challenges</th>
</tr>
</thead>
</table>
| 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] | A - "Lack of common understanding"  
B - "Difficultly in assessing consequences"  
C - "Lack of common language (product)"  
D - "Modeling/controlling multiple relations"  
E - "Controlling multiple functional states"  
F - "Synchronizing development activities"  
G - "Knowledge transfer between domains"  
H - "Lack of common language"  
I - "Different mental models"  
J - "Lack of common language (interface)"  
K - "Lack of common methodology"  
L - "Lack of common mental tools"  
M - "Lack of common information"  
N - "Lack of common design process"  
O - "Lack of common design practice"  
P - "Lack of common design tools"  
Q - "Mechatronic engineers lacking information"  
S - "Complexity as generic problem" |
| 2  | Relationship management e.g. DSM and DMM [22], [15], QFD [23], FunKey [24] |  |
| 3  | Controlling design activities through requirements management (Systems engineering [25], Tomiyama [26]) |  |
| 4  | A process model containing activities for the development process (Isermann [27], VDI2206 [28], Salminen [4], Systems Engineering [25]) |  |
| 5  | Informal description consisting of a number of modeled/described aspects to specify systems, A3 overviews [29], Salminen [4], Buur [3] |  |
| 6  | Modeling languages to describe system as a whole, formally or semi-formally, SysML [30], SFSL by Gaussianer [8], AM tool [31] |  |
| 7  | Model transformation from a design model in one domain into a design model in another domain (Gaussianer [32], Paredis [10]) |  |
| 8  | Formalized specification of interfaces. (ISO/IEC 81346 [33], Systems engineering [25]) |  |
| 9  | Simulation of phenomena that incorporate elements from the different domains (e.g. Modelica [34], Bond Graphs [35], and integrated simulation [36]) |  |
| 10 | Setting up a systems integration group in the project (Adamsson [5], Andreasen [5]) |  |

(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 through 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.
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-rings, 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
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
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 overcoming 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 metamodells 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 metamodells, 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]. 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 to have been 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 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 Wendelin, who are the founders of the watch company Linde-Wendelin [50] for letting us use the development project as the case for this paper. We would also like to thank Troels Pederesen from IPU and Henrik Bendesen from Oticon for their help in adding detailed information to build the case.

References
