INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED11
15-18AUGUST2011,TECHNICAL UNIVERSITY OF DENMARK

A FRAMEWORK FOR COMPARING DESIGN MODELLING APPROACHES ACROSS DISCIPLINES

Boris Eisenbart[1], Kilian Gericke[1] and Luciënne Blessing[1]
(1) Université du Luxembourg

ABSTRACT

Design models are an important means for the representation of design information in product development processes. Designers use design models to visualise and communicate their ideas to other members of a design team, the project manager or a costumer. Communication between experts from different disciplines using design models is often hindered by different terminology and different ways of modelling. Potential consequences are design flaws, which may lead to time-consuming iterations and – if undetected – to problems during production or use.

In order to enhance the communication through design models across disciplines, and to address the mentioned problems, an understanding of similarities as well as differences between modelling approaches needs to be established. For that purpose, this paper discusses different design states which represent a distinct level of available information in the development process. They are derived through comparing different design models proposed in literature and provide the basis of a framework for a detailed comparison of modelling approaches across disciplines. Further, first insights into different ways of modelling are discussed.

Keywords: design states, design models, design methodologies, interdisciplinary design

1 INTRODUCTION

1.1 Motivation

Industrial product development experiences a growing pressure due to global competition, shorter product life-cycles, higher quality standards, and increasing product functionality demanded by consumers. The resulting product complexity requires the collaboration of experts from various disciplines within the product development process [1, 2]. In order to optimally co-ordinate individual design activities in complex (interdisciplinary) product development projects and to ensure every designer to be working towards a common goal, Valkenburg [3] stresses the importance of establishing what she refers to as „a shared understanding“ of both design problem and potential solution. “*Not understanding the overall system is a source of uncertainty and errors in the design*“ [4]. As a consequence, communication between members of the design team and thus across different disciplines, is an essential part of successful interdisciplinary product development [1, 2, 5]. Designers need to elaborate and clarify to each other, for example,

- the constraints, the required functionality (overall and sub-functions), and architecture of the product or system to be developed,
- the working principles, and properties of sub-systems, as well as
- the interfaces between different sub-systems, and the interfaces of the overall system with its environment.

Apart from verbal exchange, communication is mainly facilitated through design models as a means of abstract representation and visualisation of information [3, 6]. Various authors – irrespective of disciplines (e.g. [1, 2, 7, 8]) – hence stress the importance of design modelling in the progress of product development projects. Buur and Andreasen [9] for example, studying interdisciplinary design teams in mechatronic product development, emphasise

> “*...the success of a mechatronic design project does not only depend on the specialized skills of the designers but perhaps even more on their abilities to communicate and visualize their ideas to the rest of the project group.*”

Most modelling approaches are essentially mono-disciplinary, as they are typically proposed in discipline-specific methodologies. It is because of this mono-disciplinary nature, that these are not sufficiently able to make discipline-specific information accessible to team members across disciplines. As a consequence, the establishment of a shared understanding is often hindered by

- differences in terminology ([5, 10]),
- different ways of modelling (which may lead to misunderstandings), as well as
- lack of knowledge about other disciplines, lack of knowledge about what (information) is requested, respectively needed or is available, respectively *can* be requested, as well as
- additional effort e.g. to explain design models to designers from other disciplines, in order to make the modelled information accessible to them.

The potential design flaws resulting from these misunderstandings may require iterations in the design process [11], lead to project failure, and – if undetected – to problems during production or usage of the product. Design modelling across disciplines needs to be enhanced, so as to overcome these problems and support the exchange of information [1, 7]. For that purpose, it is essential to establish a basic understanding of similarities as well as differences between specific design modelling approaches in collaborating disciplines, by comparing these systematically.

1.2 Research Focus

Mechatronic product development is common in industry. Despite extensive support in each discipline involved in mechatronics, there is still an insufficient theoretical basis for the symbiosis between them. The aim of this paper is to present a framework for comparing design modelling approaches across disciplines. This framework has been developed based on the results of a literature study on discipline-specific product development approaches proposed by various international authors from different disciplines. The literature review was conducted for the analysis and categorisation of the proposed design models within the individual product development approaches and focuses on literature from disciplines frequently collaborating in the development of mechatronic products (or multi-technology products in general), like mechanical engineering, electrical engineering, and software design.

In addition, building design is addressed as a means to apprehend inspiration from another field of designing, which is not solely focused on the development of *technical* products. While the involved sub-disciplines in the development of multi-technology products typically operate in parallel [12, 13], in building design different phases of the design process usually involve different people: concept development is usually carried out by architects, while civil engineers and building services engineers, following up on one another, are more focused on detail design [14]. Since individual designers operate sequentially, exchange of all the available information is essential at handover, requiring intensive use of design models. Looking at the design models and modelling approaches in building design, thus, may suggest potential solutions to the communication problems, in the development of multi-technology products.

2 DESIGN MODELLING

In his theory on models and modelling activities Stachowiak [15] describes models, to be “an excerpt of reality”, which are always referring to an object of whom they represent certain aspects without being identical to it. Further, models serve a user, which he refers to as “subject”, for a specific purpose over a period of time. Stachowiak and Buur and Andreasen [9] emphasise that design models are never able to grasp the entirety of available information about the object (i.e. the product) being developed. A design model can merely address a limited number of properties, representing different perspectives and thus playing a different role in the development process. As different levels of information about the addressed problem and the potential solution require different models to represent this information, different modelling approaches are applied in the progress of product development [3]. Only the combination of design models provides a – more or less – complete picture of the currently available information about the product being developed.

Design models are not limited to formal mathematical models used in e.g. finite elements analysis or control theory, which typically occur in a later phase of the design process. Graphical design models like conceptual sketches, diagrams, schemes, technical drawings, and three dimensional models as well as bills of material are of much higher importance to the design engineer [7]. Figure 1 shows examples of design models from the disciplines collaborating in the development of e.g. mechatronic products.

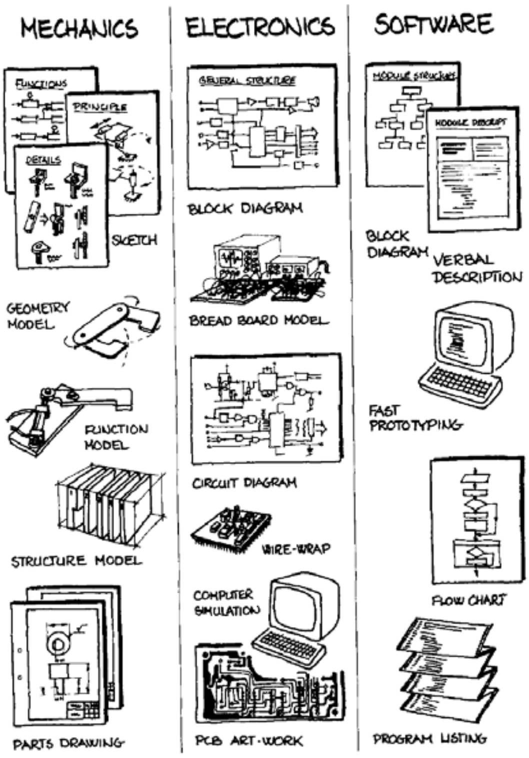

Figure 1: examples of design models in mechatronic product development [9]

Buur and Andreasen [9] derived a modelling morphology for design models in mechatronic product development, based on the definitions of central terms in modelling theory by Hubka [16]. The morphology is divided into the *modelling activity* and the *design model* itself. The modelling activity addresses the aspects of: the object, the modelled properties, the purpose of modelling, and the user of the model. The design model uses a code, e.g. human language, symbols, drafting standards, etc. and a medium, for instance paper, pictures, etc., while differing in detail and level of abstraction. Design models can address more than one of the aspects in each of the categories, as e.g. prototypes not only address dimensions but also represent functionality, usage etc. [17]. This detailed classification enables a systematic comparison of individual design models.

3 DESIGN STATES

A *design state* is defined, according to Dym [18], as the incorporation of all the information about a design as it evolves. Apart from supporting communication, design models are important means for capture and storage of information generated in the progress of product development: new information is typically stored in a new or updated design model. Roth [19] even describes the development of a product as progressing from one design model to another, as systematic product development approaches often propose the sequential creation of various documents, partly based on one another, in the progress from problem to solution. The process of product development can thus be captured by analysing the successively proposed design models. Joining Roth’s and Dym’s individual perspectives, the design process moves through a succession of design states, corresponding to a succession of particular (sets of) design models, storing the gained information. Within one design state, the corresponding design models address similar information, using different modelling approaches and (potentially) serving different purposes.

Blessing [20] identified several design states which are common in mechanical engineering. She discusses different strategies for product development – product- and problem oriented – based on the design states addressed and their specific chronological order. Problem-oriented approaches show a step back from addressing the product idea to a detailed elaboration of the problem space through various steps of abstraction. Product-oriented design approaches tend to move directly from a specific product idea to an overall product solution, therefore, addressing fewer design states than problem oriented ones. This is illustrated in Figure 2.

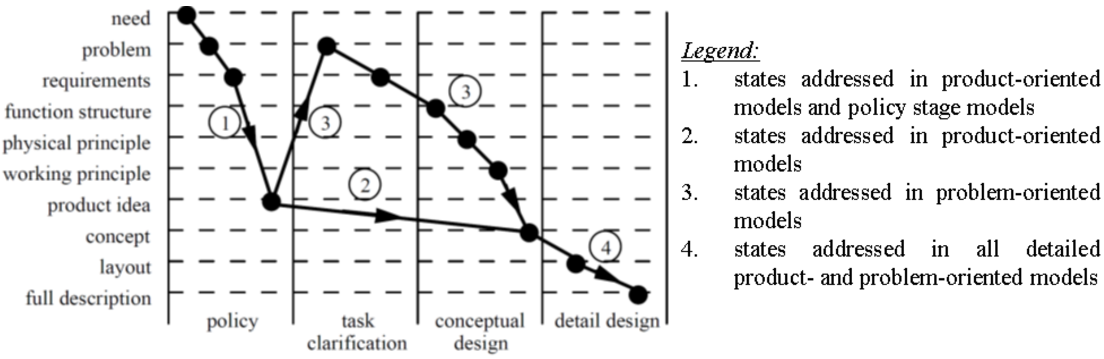

Figure 2: Different design strategies after [20]

Identification of the *stimuli* of product development [21] and proposition of the development task typically mark the end of the *policy* or *product planning* stage, which leads to the elaboration of the design problem, resulting in a problem definition and a set of requirements which finalise the *task clarification* stage. In the conceptual design stage the required functionality is transferred into a design concept, which is detailed and refined within the *detail design stage*, resulting in a full description of the product, so as to enable production [22].

As a design task is widely seen as an ill-structured problem, the development process cannot easily move from the problem to a solution [23]. In the progress of exploring the potential solution(s) to the given design problem, certain features and constraints of that potential solution typically lead to a (reoccurring) redefinition of the problem space. This is usually referred to as *co-evolution*: the stepwise increase of knowledge about the addressed problem parallel to the developed solution, i.e. the product [23]. This repetitive moving back and forth between problem and solution is not represented in Figure 2, nor are the various iterations, which typically occur in the design process.

The “creative leap” from a description of the problem over to its potential solution is also manifested in the proposed design models in literature. While problem definitions mainly consist of textual descriptions of the needs, requirements or constraints, design models addressing the solution are rather focused on specifying and visualising characteristics as well as evaluating, analysing and simulating properties of a potential product solution. As this is a major difference between modelling approaches, design states are regrouped prior to a more detailed analysis into *problem states* and *product states*.

- Within a *problem states* a given problem, a task (as well as requirements and constraints) are addressed, *without* proposition of a potential solution
- Within a *product states* information about the potential solution is addressed.

4 A FRAMEWORK FOR COMPARING MODELLING APPROACHES

4.1 Development

As discussed above, design models within one design state address similar information on a similar level of detail and abstraction, thus enabling comparison of the particular way the information is modelled. Based on Blessing’s elaborations [20] it can be assumed that there are also common design states in other disciplines, as well as *generic* ones across disciplines. Within these generic design states, the corresponding modelling approaches can be compared across disciplines throughout the entire development process. Identification of design states is facilitated through systematically categorising the design models proposed in methodologies according to Buur and Andreasen’s modelling morphology: *object* (e.g. function structure, feature list), *property* (e.g. specification of overall and sub-functions) and *purpose* (e.g. specify, structure, etc.) of individual design models are used to identify the corresponding design states. Regarding the used code, symbols etc., the categorisation of individual design models addressing the same generic state, may then be used to analyse the differences in the applied modelling approaches across disciplines. The review of the individual systematic product development approaches in literature, thus aims at:

- collecting the proposed design models,
- categorising individual design models, so as to identify discipline-specific as well as common (generic) design states,
- thus enabling a detailed analysis and comparison of discipline-specific modelling approaches addressing similar design information (the same generic design state, respectively).

Literature analysis

Wynn and Clarkson [24] distinguish between *analytical*, *abstract* and *procedural* approaches to product development. Analytical approaches do not address the entire process of product development, but rather focus on specific steps or individual activities in the overall process. Abstract approaches typically describe product development on a high level of abstraction which allows application in a wide range of situations. They are less focused on specific disciplines and do not provide specific guidance in the product development process.

Procedural approaches on the other hand are more focused on the development of a product in a specific discipline and address the concrete steps and activities within the different phases in great detail. They are typically represented in *design methodologies* and provide a systematic approach consisting of phases and distinct design activities typically including the proposition of corresponding design models. The combination of process and proposed design models enables a detailed analysis of the design states.

In electrical engineering, design methodologies, thoroughly capturing the development process, are rather scarce. Electrical engineers state, that design methodologies in electrical engineering until recently have rarely been an issue. With the strong increase of digital electronics and expensive wafer-technology, wherein the systematic planning of the different manufacturing processes (e.g. etching, sputtering etc.) has an essential influence on the physical layout of the product, the need for a systematic, methodical approach towards product development has arisen. Current popular approaches are the Very Large Scale Integration (VLSI) approach, based on Walker and Thomas' Y-model [25] as well as Electrical Design Automation (EDA), which is strongly supported by computer tools.

Design methodologies in building design covering the entire process tend to originate from project management [26, 14]. Literature from *architecture* typically only addresses the development process up to the point detail design begins and the concept is handed over to civil and building services engineers.

The VDI guideline 2221 [13] introduces a process model based on mechanical engineering, but also addresses electrical engineering and software design, which is why it can be regarded as a rather interdisciplinary approach. Zeiler and Savanovic [27] and Cross [28] both propose approaches combining design methods from various disciplines (Cross mainly from architecture and mechanical engineering; Zeiler and Savanovic from German, Dutch and Anglo-American design literature and systems engineering) which can also be regarded as interdisciplinary approaches.

Identification of design states

Table 1 illustrates individual design methodologies, which propose design models corresponding to the individual design states, divided between problem and product states. Design states addressed in each discipline – *generic* design states – are highlighted in bold. Individual methodologies often address additional design states, which do not occur in other methodologies in their own discipline or in that of others. Those can therefore not be regarded as discipline-specific or as generic, and therefore are not presented here. The design states of *context analysis* and *project proposal* cannot be classified as either product or problem state, as they address a problem as well as a potential solution.

Table 1: design states addressed in methodologies across diciplines

design state		**mechanical engineering**									**electrical engineering**				**software design**						**building design**							**mechatronics/ interdisciplinary approaches**				
problem state	**product state**	**Pahl and Beitz [21]**	**Ulrich and Eppinger [29]**	**Ehrlenspiel [30]**	**Hubka [31]**	**French [32]**	**Archer [33]**	**Pugh [34]**	**Roozenburg and Eekels [35]**	**Tjalve [36][1]**	**Walker and Thomas [25]**	**EDA – Dewey [37]**	**EDA – Handbook [38]**	**VLSI [39]**	**Royce (Waterfall) [8]**	**Boehm [40]**	**IGAB, V-Modell [41]**	**Kruchten – RUP [42]**	**Scrum [43]**	**Cooper [44]**	**Darke [45]**	**Rittel [46]**	**Alexander [47]**	**Jones [48]**	**Lawson [49]**	**Engel [14]**	**Dalziel (RIBA) [26]**	**Zeiler and Savanovic [27]**	**Cross [28]**	**VDI 2221 [13]**	**VDI 2206 [12]**	**Salminen [50]**
problem statement					x				x									x		+				x	x	x		+				
context analysis		x	x	x				x												+						x	x					
need		x	x		x		x	x	x		x	x	x					x	x						x	+	x	+	x	+		
	product idea	x		x					x				x									o	o	x	x							+
product proposal		x	x	x				x							+		x			+					x	x	x		x	+	+	
design objective specification			x				x									o				+				x		x		+				
requirements specification		x	x	x	x	x	x	x	x	x	x	x	x	x	+	o	x	x	x	+	o		o			x	x	+	x	x	x	x
	product functionality	x	x	x	x	x	x	x		x		x	x	x			x	x	x	+				x		x	x	+	x	x	x	+
	principle solutions	x	x	x	x		x			x				x								o		x	x	x		+	x	x	x	x
	working structure	x	x	x	x		x			x		x	x					x						x	x	x		+	x	x	x	x
	conceptualisation	x	x	x	x	x	x	x	x	x	x	x	x		+	o	x	x	x	+			o	x	x	x	x	+	x	x	x	+
	preliminary layout	x	x	x	x	x	x	x	x	x	x	x	x	x	x	o	x	x						x	x	x	x	+	x	x	x	
	layout	x	x	x	x	x	x	x	x	x	x	x	x	x		o	x	x	+	+			o			x	x	+	x	x	x	
	production documents	x	x	x	x	x	x	x		x				x												x	x	+	x	x	x	

"x" – Example of design model given or content specified in publication

"+" – Specific design model mentioned, but neither example given, nor content specified

"o" – Design state mentioned, but no specific corresponding design model proposed/ mentioned

---

[1] Tjalve's product development approach addresses industrial design for mechanical products

The following design states can be found across disciplines. The succession of individual design states presented below varies in some methodologies, up to the state of requirements specification. The presented order represents a tendency for the succession proposed in the individual design methodologies.

- *Problem statement:*
  A short, (usually textual) description of the design problem or development task.
- *Context analysis:*
  A problem statement or design task typically leads to the analysis of the product context (e.g. competitor analysis), in order to elaborate general requirements (needs) and constraints of the product and formulation of objectives for the development process.
- *Need:*
  Although not every design methodology explicitly addresses this design state or proposes the creation of corresponding design models, they typically recognise basic stakeholder or market needs to be driving a development process.
- *Product idea:*
  An initial, more or less detailed, idea of a potential new product or an idea for the introduction of new technologies, functionalities or looks etc. for an existing product.
- *Product proposal:*
  A product proposal typically addresses an idea for a new product, its main functionalities, initial requirements, as well as a target budget and potential time consumption of the design project.
- *Design objective specification:*
  Description of the design target, typically including (values for the) overall objectives like “cost-reduction”, “quality improvement”, etc.
- *Requirements specification:*
  Documentation of required functionality, important influences, constraints and dependencies, as they result from the specific demands, needs and wishes of the stakeholders, the market, etc.
- *Product functionality:*
  Detailed documentation of required main- and auxiliary functionality. Typically this includes the analysis of the overall function which is broken down into (basic) sub-functions.
- *Principle solution:*
  These are working principle elements (physical effects, functional blocks, etc.) which are suitable for fulfilling a specific sub-function of the product.
- *Working structure:*
  A working combination of principle solutions to fulfil the required sub-functions.
- *Conceptualisation:*
  The principle solutions specified in the working structure are integrated into an overall solution concept, consisting of various components, respectively modules or functional elements.
- *Preliminary layout:*
  The components or modules, respectively the solution elements, of the overall solution are detailed and integrated into one overall layout.
- *Layout:*
  The individual components which build up the overall layout are detailed to completion.
- *Production documents:*
  Finalisation of design models which specify all the information required for production.

Table 2 gives examples of typical discipline-specific design models representing these generic design states.

Regarding the generation of production documents, a speciality can be found in software design: While in other disciplines, the development process and the production process are typically sequential – requiring the creation of specific documents to enable manufacture of the product – in software design the product (i.e. the program code) is produced in the course of the actual design process. For that reason, there are typically no production documents proposed in literature from software design. “Final documentation” proposed in other methodologies (e.g. VDI guideline 2221 [13]) in software development, usually refers to the product manual.

*Table 2: **examples** of discipline-specific design models representing **generic** design states*

Problem states	Product states	Mechanical engineering	Electrical engineering	Software design	Building design
Need		text, (hierarchical) documentation	*	product backlog	initial design brief
Requirements specification		requirements list	requirements list	feature list, release backlog	design brief
	Product functionality	function structure	functional blocks hierarchy	use case definition	network planning
	Working structure	working structure	basic solution elements structure	initial system architecture	design proposal(s)
	Conceptualisation	concept sketches	module structure	system architecture	lead model
	Preliminary layout	preliminary layout	logic plan, circuit diagram	integration of system components	layout drawings
	Layout	dimensional layout	physical layout drawings	program code	technical design drawings

* *no corresponding design model proposed in analysed design methodologies*

4.2 Applying the framework

In order to illustrate how a similar design state is represented by discipline-specific design models, this section discusses examples for the discipline-specific modelling of *product functionality*, illustrated in Figure 3. A very common example of function modelling in mechanical engineering and mechatronics is a *function structure*, presented as block diagrams (**a**) or hierarchical trees (**b**). Function structures often differ in the level of detail and types of the proposed basic sub-functions in the different design methodologies. A hierarchical tree illustrates the overall functionality (circle A in **b**) which is decomposed continuously into sub- and auxiliary functions (circles B, C, E and F). The VLSI approach [39] in electrical engineering proposes a variant of such a hierarchical tree, wherein decomposition is repeated until basic sub-functions (circles D, G, H and I) are found, which can directly be fulfilled by re-usable electrical functional devices (e.g. converters, switches, logical elements etc.).

In software design Kruchten [42] proposes the creation of *use case schematics*, which indicate a flow of events, to be enabled by the program, to specify and elaborate the required software functionality. An actor starts a request, which the program has to process and link with other requests (**c**). Use case schematics share certain aspects of a flow chart including *actors* to request the functionality i.e. software processes. Scrum [43] on the other hand simply distinguishes between main- and sub-functions to be fulfilled by the software, which are represented textually and divided into different sub-documents.

In building design the elaboration of the required functionality can be a detailed analysis as presented in **d**. It can be regarded as a form of product functionality specification as it illustrates the material flow (respectively, the flow of people) between functional elements (in this example individual rooms are allocated to specific processes: storage, WC, etc.) in a factory building, in a structured way – which an essential characteristic of a function structure. The thickness of the connecting lines indicates the quantity of the flow between spatial areas, which can be used to arrange them optimally in a building [14]. Table 3 shows the categorisation and comparison of the exemplary design models.

Table 3: categorisation of examplary design models

design model	object	property	purpose	user	code	medium
Function structure	function structure	overall-and sub-functions, energy and signal flow	structure and elaborate information	project team, project manager, the designer himself	blocks, arrows indicating energy, signal flow	various options (e.g. paper, computer screen, etc.)
Actors and use case schematics	required system behaviour	behaviour; actors and procedure	visualise		symbols, arrows, blocks	
Functional network planning	functional elements, interfaces	material flow	visualise, elaborate		blocks, quantified connectors	
Function tree	product functionality hierarchy	overall and auxiliary functions	structure, visualise		tree hierarchy	

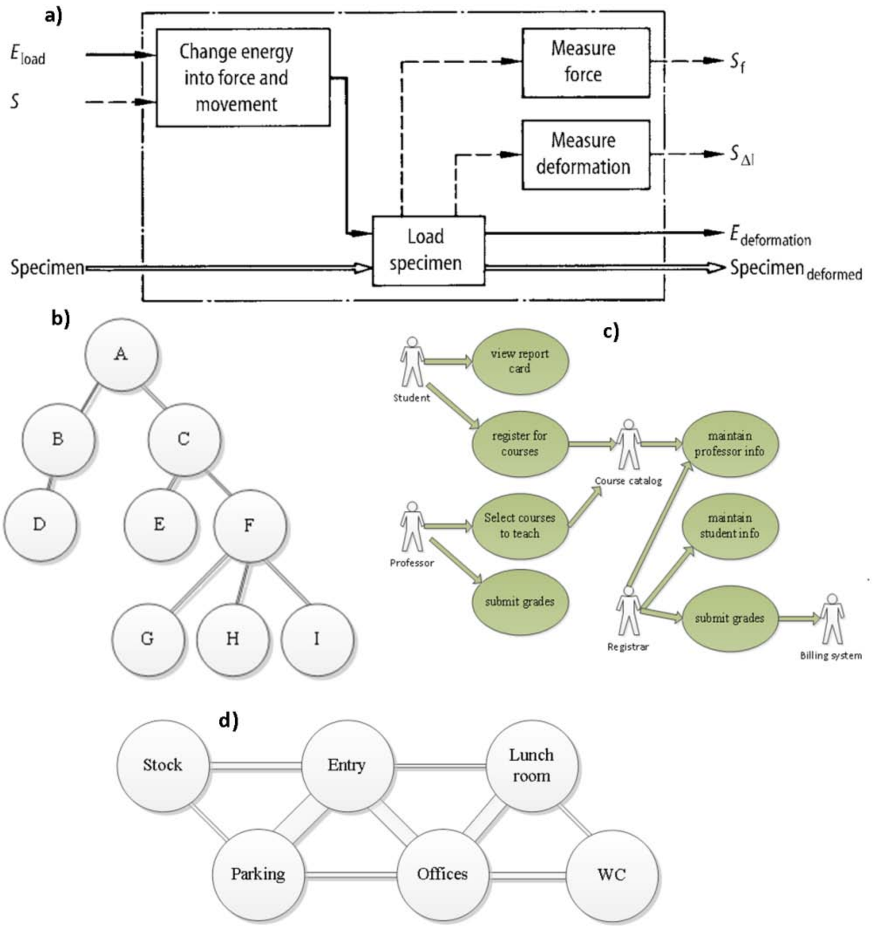

Figure 3: examples of design models addressing product functionality a) function structure [21] b) hierarchy of functional blocks [39] c) example of use case schematics after [42] d) functional network planning [14]

This small example illustrates four different ways of modelling functionality in different disciplines and enables an insight into the different discipline-specific perspectives onto modelling design. According to interviewed designers, the problems in understanding design models across disciplines originate mainly from the used code and symbols, discriminating crucially from each other due to the different perspectives dominating in each discipline. Although the generated design models share the purpose of modelling the required functionality, their focus is rather different. For instance, function structure and function tree both illustrate decomposition into sub- or auxiliary functions, however, signals or energy flows are only represented in a function structure.

Design models in software design typically address a higher level of abstraction (e.g. system architecture or product structure) than similar design models in other disciplines, as the actual product (code) is already generated while it is being designed, thus decreasing the need for product representing models. For obvious reason, in software design and design of electronics, there is no flow of materials, only of energy and signals. Nonetheless, signals cannot exist on their own, but in some variation of energy and are channelled through materials (or waves) which may provide an approach for integrating the different discipline-specific perspectives.

5 DISCUSSION

Application of the developed framework on the proposed discipline-specific design models in various methodologies led to the identification of discipline-specific as well as generic design states, enabling comparison of individual modelling approaches within individual design states across disciplines.

The conducted literature study implicates that there are additional design states represented by design models meant to support project management (e.g. product proposal). As these design models need to give a comprehensive overview of the current state of the design project, they have to combine information from every discipline collaborating in this particular project. A detailed analysis of these design states might provide further insights into the problems that occur in communication between experts from different disciplines.

Various design methodologies across disciplines propose decomposition into individual sub-functions and (discipline-specific) modules (discussed above) connected via interfaces, i.e. the generation of a product structure – which is an essential concept of systems engineering. These similarities, in addition to the analogies of material, signal or energy flow (respectively energy or signal flow through materials) across disciplines, may proof useful in order to develop a support modelling across disciplines. Decomposition is an effective approach to reduce complexity in multi-technology product development and has made its way into current design methodologies and practice in each studied discipline (even in building design, using blocks representing displaceable spatial areas [14, 48]), but especially into mechatronic process models.

Nevertheless, decomposition rarely produces truly discipline-specific modules in mechatronic product development. Modelling of a mechatronic system needs to address both physical and virtual elements from each of the disciplines involved (material, energy, signals), thus suggesting the need for an integrative way of modelling. Especially, as a solution for a module developed by one discipline can often be exchanged by a solution from another discipline: exchanging a mechanical element by an electro-mechanical one.

6 CONCLUSIONS

The communication between designers from different disciplines, using design models, is not sufficiently supported in design research and practice for the development of mechatronic products (or multi-technology products in general). The presented framework is based on an extensive literature study on design models proposed in design methodologies from various international authors and enables a detailed comparison of the individual discipline-specific modelling approaches across disciplines. The literature review led to the identification of various *discipline-specific* as well as *generic* design states, which could be sub-divided further into problem and product states. Within a generic design state, corresponding design models address similar information across disciplines. Based on the modelling morphology proposed by Buur and Andreasen [9], the specific way (the used code, symbols, etc.) this similar information is modelled – irrespective of disciplines – can be analysed systematically and the specific similarities and differences be elaborated.

How the gained insights into the individual modelling approaches can be used to integrate the different discipline-specific perspectives of modelling in mechatronic products efficiently and correctly will be elaborated in future research, through analysing the practical application of the studied design models.

REFERENCES

[1] Shi, B., Design for Multi-Technology Systems, 2003, (Dissertation, University of Erlangen-Nürnberg)

[2] Redenius, A., Verfahren zur Planung von Entwicklungsprozessen für fortgeschrittene mechatronische Systeme, 2006, (Dissertation, Heinz Nixdorf Institut, University of Paderborn)

[3] Valkenburg, R. C., The reflective practice in product design teams, 2000, (Dissertation, Industrial Design Engineering, Delft University of Technology)

[4] van Beek, T., Tomiyama, T., Connecting Views in Mechatronic Systems Design, a Functions Modelling Approach, in *Proceedings of ASME International Conference on mechatronic and Embedded Systems and Application*, 2008

[5] Goel, V., Sketches of Thought, 1995, (MIT Press)

[6] Stacy, M., Eckert, C., Against Ambiguity, in *Computer Supported Cooperative Work*, pp. 153-184, 2003, (Kluwer Academic Publishers)

[7] Andreasen, M. M., Modelling – the Language of the Designer, in *Journal of Engineering Design,* Vol.5, pp. 103-115, 1994

[8] Royce, W. W., Managing the Development of Large Software Systems, in *Proceedings of IEEE, Wescon*, 1970

[9] Buur, J., Andreasen, M. M., Design Models in Mechatronic Product Development, in *Design Studies Vol. 10 No. 3,* 1989, (Butterworth & Co, Belfast)

[10] Zimmermann, T., Implementing PLM Across Organisations – For Multi-disciplinary and Cross-function Product Development, 2008, (Dissertation, Department of Product and Production Development, Chalmers University of Technology)

[11] Gries, B., Design Flaws and Quality-Related Feedback in Product Development, 2007, (Engineering Design and Methodology, Technische Universität Berlin)

[12] VDI 2206, Design Methodology for Mechatronic Systems, 2004, (VDI Verlag GmbH)

[13] VDI 2221, Systematic Approach to the Design of Technical Systems and Products, 1993, (VDI Verlag GmbH)

[14] Engel, H., Methodik der Architektur – Planung, 2002, (Bauwerk, Berlin)

[15] Stachowiak, H., Allgemeine Modelltheorie, 1973, (Springer, Wien)

[16] Hubka, V., Terminology of Science of Design Engineering in 6 Languages, 1980, (Schriftenreihe WDK 3, Heurista, Zürich)

[17] Brandt, E., How do Tangible Mock-ups Support Design Collaboration?, in *Proceedings of Nordic Design Research Conference, Copenhagen*, 2005

[18] Dym, C. L., Engineering Design – A Synthesis of Views, 1994, (Cambridge University Press)

[19] Roth, K., Modellbildung für das methodische Konstruieren ohne und mit Rechnerunterstützung, in *VDI Z Vol 128 No. 1*, 1986, (Springer, Düsseldorf)

[20] Blessing, L., Comparison of Design Models Proposed in Prescriptive Literature, in *Social Sciences Series Vol. 5,*1996, (COST A3 / COST A4 International research workshop, Perrin, J. and Vinck, D., Lyon, European Committee)

[21] Pahl, G.,Beitz, W., Feldhusen, J., Grote, K. H., Engineering Design – A Systematic Approach, 2007, (Springer, London)

[22] Blessing, L. T. M., A Process-Based Approach to Computer-Supported Engineering Design, 1994, (Dissertation, University of Twente)

[23] Maher, M. L., Poon, J., Boulanger, S., Formalising Design Exploration as Co-Evolution, in *Advances in Formal Design Methods for CAD,* Editor: Gero, J., June 1995, (Chapman and Hall, London)

[24] Wynn, D., Clarkson, J., Models of designing, in *Design Process Improvement – a review of current practice*, Editors: Clarkson, J., Eckert, C., 2005, (Springer, London)

[25] Walker, A. R., Thomas, D. E., A Model of Design Representation and Synthesis, *Proceedings of 22nd Conference on Design Automation*, 1985, pp. 453-459

[26] Dalziel, B., Architect's Job Book, 1995, (RIBA Publications, London)

[27] Zeiler, W., Savanovic, P., General Systems Theory Based Integral Design Method, *Proceedings of International Conference on Engineering Design*, Stanford, 2009, pp. 217-228

[28] Cross, N., Engineering Design Methods – Strategies for Product Design, 2008, (Wiley, Chichester)

[29] Ulrich, K. T., Eppinger, S. D., Product Design and Development, 2008, (McGraw-Hill, New York)

[30] Ehrlenspiel, K., Integrierte Produktentwicklung – Denkabläufe, Methodeneinsatz, Zusammenarbeit, 2007, (Carl Hanser, Vienna, Munich)

[31] Hubka, V., Principles of Engineering Design, 1980, (Butterworth & Co, Belfast)

[32] French, M., Conceptual Design for Engineers, 1999, (Springer, London)

[33] Archer, B., Systematic Method for Designers, in *Developments in Design Methodology*, Editor: Cross, N., 1984, (Wiley, Chichester)
[34] Pugh, S., Total Design – Integrated Methods for Successful Product Engineering, 1991, (Pearson Education, Edinburgh Gate, Harlow)
[35] Roozenburg, N. F. M., Eekels, J., Product Design: Fundamentals and Methods, 1995, (Wiley, Chichester)
[36] Tjalve, E., Systematische Formgebung für Indutrieprodukte, 1978, (VDI-Verlag)
[37] Dewey, A., Digital and Analog Electronic Design Automation, in *The Electrical Engineering Handbook*, 2000, (CRC Press LLC)
[38] Scheffer, L., Lavagno, L., Martin, G., EDA for IC System Design Verification and Testing, 2006, (Taylor & Francis) n Inc., Boston)
[39] Bleck, A., Goedecke, M., Huss, A., Waldschmidt, K., Praktikum des Modernen VLSI-Entwurfs, 1996, (Teubner, Stuttgart)
[40] Boehm, B., A Spiral Model of Software Development and Enhancement, in *IEEE Software Design Vol.21 No.5*, 1988, pp. 61-72
[41] IGAB, V-Modell XT, https://www.v-model.iabg.de
[42] Kroll, P., Kruchten, P., The Rational Unified Process Made Easy – A Practitioner's Guide to the RUP, 2003, (Pearson Educatio
[43] Schwaber, K., Agile Project Management with Scrum, 2007, (Microsoft Press, Germany)
[44] Dubberly, H., Alan Cooper and the Goal Directed Design Process, *AIGA Journal of the Network Economy*, Vol.1 No.2, 2001
[45] Darke, J., The Primary Generator and the Design Process, in *Developments in Design Methodology*, Editor: Cross, N., 1984, (Wiley, Chichester)
[46] Rittel, H., Planen, Entwerfen, Design – Ausgewählte Schriften zu Theorie und Methodik, 1992
[47] Alexander, C., Notes on the Synthesis of Form, 1964, (Harvard University Press)
[48] Jones, J. C., Design Methods – Seeds of Human Future, 1980, (Wiley, Chichester)
[49] Lawson, B., How Designers Think – The Design Process Demystified, 2000, (Architectural Press)
[50] Salminen, V., Verho, A. J., 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

Contact: Boris Eisenbart
Université du Luxembourg
Research Group in Design Methodology
6, rue Richard Coudenhove Klergie
Luxembourg, L-1359
Grand Duchy of Luxembourg
Tel: +352-4666445422
Email: Boris.Eisenbart@uni.lu

ACKNOWLEDGEMENTS

The authors would like to thank the *Fonds Nationale de la Recherche* Luxembourg which has been funding this research (project: AFR PhD-09-186).