Functional Decomposition: On Rationality and Incommensurability in Engineering

Functional Decomposition
On Rationality and Incommensurability in Engineering

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op donderdag 21 april 2011 om 12.30 uur

door Dingmar van Eck
doctorandus in de theoretische psychologie

geboren te Gouda

Dit proefschrift is goedgekeurd door de promotor:
Prof. dr. ir. P.A. Kroes

Copromotor:
Dr. P.E. Vermaas

Samenstelling promotiecommissie
Rector Magnificus, voorzitter
Prof. dr. ir. P.A. Kroes, Technische Universiteit Delft, promotor
Dr. P.E. Vermaas, Technische Universiteit Delft, copromotor
Prof. dr. P. Hoyningen-Huene, University of Hannover
Prof. dr. S. Hartmann, University of Tilburg
Prof. dr. T. Tomiyama, Technische Universiteit Delft
Dr. M.P.M. Franssen, Technische Universiteit Delft
Dr. C.M. Eckert, Open University UK
Prof. dr. M.J. van den Hoven, Technische Universiteit Delft, reservelid

Dr. P.E. Vermaas heeft als begeleider in belangrijke mate aan de totstandkoming van het proefschrift bijgedragen.

Research for this thesis was made possible by the Netherlands Organization for Scientific Research (NWO).

ISBN: 978-90-386-2463-1
ISSN: 1574-941X

Contents

Acknowledgements ix

1 Introduction 1

1.1. On the anomalous character of engineering functional decomposition 1
1.2. Engineering functional decomposition and methodological incommensurability 6
1.3. Discussion 27

2 On Relating Functional Modeling Approaches: Abstracting Functional Models from Behavioral Models 35

2.1. Introduction 36
2.2. Functional modeling: a survey 38
2.3. Establishing function-behavior correspondences 45
2.4. Abstracting functional models from behavioral models 49
2.5. Conclusions 54

3 Functional Decomposition in Engineering: a Survey 61

3.1. Introduction 62
3.2. The Functional Basis approach 63
3.3. The Function-Behavior-State approach 68
3.4. The Chakrabarti-Bligh approach 73
3.5. Comparison 76
3.6. Conclusions and outlook 82

4 Explaining and Relating Different Engineering Models of Functional Decomposition 87

4.1. Introduction 88
4.2. Simplifying full descriptions of technical devices: relating goal to behavior and/or structure in different ways 90
4.3. Engineering models of functional decomposition 92
4.4. Choosing functional decomposition models: design knowledge employment and design objectives 100
4.5. Concluding remarks 103

5 Incommensurability and Rationality in Engineering Design: The Case of Functional Decomposition 107

5.1. Introduction 107
5.2. Engineering notions of function and functional decomposition 110
5.3. Explaining co-existence by variation of objectives or design knowledge employment 113
5.4. Methodological incommensurability in engineering 118
5.5. Rationality in engineering 128
5.6. Conclusion 132

6 Supporting Design Knowledge Exchange by Converting Models of Functional Decomposition 137

6.1. Introduction 137
6.2. Survey: technical function and functional decomposition 140
6.3. Approaches toward functional decomposition 143
6.4. The conversion methodology 148
6.5. Converting behavior functional decompositions into behavior models 154
6.6. Discussion 164
6.7. Conclusions 166

7 On the Conversion of Functional Models: Bridging Differences in the Modeling of User Actions 171

7.1. Introduction 172
7.2. Functional modeling approaches 174
7.3. Functional model conversions 178
7.4. Problems concering user functions 183
7.5. A strategy for translating user functions 187
7.6. Discussion: generalizing the derivation strategy 191
7.7. Conclusions 193

8 Mechanisms, Functional Hierarchies, and Levels of Explanation 199

8.1. Introduction 199
8.2. Craver's concept of role function 201
8.3. Evaluating Craver's concept of role function 208
8.4. Functional hierarchies of role functions 218
8.5. Capturing functional hierarchies of role functions 221
8.6. Concluding remarks 224

Summary 229

Samenvatting 235

About the Author 243

Acknowledgements

Writing this dissertation has been a challenge and adventure. I vividly remember the words of my supervisor Pieter Vermaas when I started this project that doing philosophical research requires that you need to like problems. Problems are nothing but challenges. These words turned out to be very true. Analytic philosophy of engineering is a young field with little or no default positions and with no established research agenda. No small amount of research challenges to tackle in such a relatively unexplored field. I have been very fortunate that during my research I could rely on the expert knowledge and research skills of several people to address them and make this research a success.

First of all I wish to thank my day-to-day supervisor Pieter Vermaas. His constructive and incisive feedback on draft papers and chapters proved invaluable in writing this dissertation. In addition to his expert knowledge on philosophy of engineering, his approach to doing philosophy provided an exemplar. I also wish to thank my promoter and official supervisor Peter Kroes for his constructive and incisive feedback and ideas. In particular our discussions on incommensurability and mechanistic explanations proved highly rewarding. Furthermore I wish to thank Maarten Franssen who generously read and commented on several papers and chapters. His sharp feedback and knowledge helped to greatly improve their quality.

I also thank my colleagues who made my workplace in Delft a pleasant one. Especially chats and discussions with my fellow PhD students and teaching philosophy of science courses with Jelle and Sjoerd were pleasant distractions from daily research.

During my research I have had the pleasure to participate in several conferences abroad. Especially the ASME and ICED conferences (USA), DRS conference (Canada), and MACITS conference (UK) were memorable events. I would like to thank the Netherlands Organization of Scientific Research (NWO) who financially supported this project and made these research visits possible.

On a personal note I would like to thank Christa and my boys Evan and Viggo. The joy you give is the best support I could wish for.

1 Introduction

1.1. On the anomalous character of engineering functional decomposition

The concept of technical function is a key concept to engineering.[1] Yet it has different meanings for different engineers.[2] Some characterize functions as *conversions of materials, energies, and signals* (see Pahl and Beitz: 1988; Stone and Wood: 2000). The function "loosen/tighten screws" of an electric screwdriver is then represented as a conversion of "screws" and "electricity" to "screws", "torque", "heat", and "noise" (see Stone and Wood: 2000, 364). Others, rather, take such conversions of materials, energies, and signals to refer to physical behaviors and use the concept of function to refer to the *desired effects of the physical behaviors* (see Deng: 2002; Chandrasekaran: 2005). The function of an electric screwdriver is then solely represented as 'loosen/tighten screws'. Or, to give another example, the function of a stapler is described as "combine sheets" (see Ookubo et al.: 2007, 154.9). In addition, the concept of function is used to refer to the *purposes for which technical artifacts are designed* (Chakrabarti: 1998: Deng: 2002). The function of an electric screwdriver is then described as 'to connect materials'. Or, to give another example, the function of a washing machine is described as "to wash laundry" (Deng: 2002, 349). How is this conceptual divergence possible? Why do engineers use the key concept of function in such different ways? Is this use irrational when measured against basic scientific standards of conceptual clarity? The situation is even more intricate in engineering because functions are often broken down or split up into a number of other (sub) functions. The relationships between functions and the sets of sub functions into which they are decomposed are then often graphically represented in *functional decomposition models*.[3] Now, like the concept of function, such models also come in a variety of flavors. They can, among others,

[1] E.g., see Umeda et al.: 1996; Chandrasekaran and Josephson: 2000; Stone and Wood: 2000; Chakrabarti and Bligh: 2001.

[2] E.g., see Chittaro and Kumar: 1998; Chakrabarti: 1998; Chandrasekaran and Josephson: 2000; Deng: 2002; Far and Elamy: 2005; Erden et al.: 2008; Van Eck: 2009; Vermaas: 2009.

[3] Functional decomposition models are the result of *functional decomposition strategies*, i.e., the breaking down of a function into a number of other functions.

represent sets of sub functions that refer to physical behaviors, sets of sub functions that refer to the desired effects of behavior, and sets of sub functions that refer to purposes (see chapters 4-6). Engineers put such models to a variety of uses. They use them, among others, in the conceptual phase of engineering designing to specify the desired functions of some artifact-to-be (see Stone and Wood: 2000; Chakrabarti and Bligh: 2001); in the reverse engineering of existing artifacts to identify their functions (see Otto and Wood: 2001); and engineers use functional decomposition models to identify malfunctions of artifacts (see Bell et al.: 2007).

What leads different engineers to employ different notions of function and different models of functional decomposition? This lack of consensus is surprising, given the importance of the concept of function to engineering and the by-engineers-recognized problems that this conceptual diversity leads to. Engineers readily acknowledge that the usage of different meanings of the concept of function hampers adequate information exchange between (collaborating) engineers.[4] For instance, it creates problems for the translation of functional decomposition models across functional modeling frameworks. And this conceptual diversity, similarly, is acknowledged to hamper the exchange of information between computer systems with different underlying functional conceptualizations (see Kitamura et al.: 2007). Given these communication problems one would, both from an analytic philosophical position and an instrumental point of view, expect a focused debate to arrive at commonly shared functional conceptualization(s).[5] Yet, engineering is defying expectations: there is no such commonly agreed upon functional conceptualization. And signs that engineering will eventually establish one are hard to detect. Whereas in science debates on the adequacy of key conceptualizations seem commonplace, such debate is by and large not waged in engineering. And whereas in science such debates have led to commonly agreed-upon key conceptualizations,[6] such

4 E.g., see Rosenman and Gero: 1999; Szykman et al.: 2001; Deng: 2002; Kitamura et al.: 2007.

5 With an analytic philosophical position I hear mean the philosophy of scientific method or meta-methodology (see Kuhn: 1977; Laudan: 1987; Worrall: 1988; Sankey: 2002). This field debates on the pros and cons of particular scientific methods as well as on meta-scientific prescriptions for doing scientific research. Discussions on methodological incommensurability and theory choice are part of these debates.

6 Examples are the efforts spent in psychiatry and clinical psychology to arrive at unambiguous and shared classification criteria for psychiatric disorders as laid down in the Diagnostic and Statistical Manual of Mental Disorders (DSM-IV) (see American Psychiatric Association:

conceptual convergence seems not in view in engineering. A few authors do express wishes to develop a common framework for functional modeling by means of specific functional conceptualizations (see Erden et al.: 2008; see also Chandrasekaran: 2005). Yet, the majority of authors seem to resist such steps toward conceptual uniformity. Most engineers simply stick to advancing their favored functional conceptualizations without superiority claims over others, merely enriching the available spectrum of functional modeling frameworks. Recent attempts to formalize specific notions of function and/or functional decomposition model reinforce this status quo. These formalizations, laid down in what are called *function ontologies*, are aimed to support the consistent storage, retrieval, and cross-communication of functional descriptions of artifacts (see e.g., Szykman et al.: 2001; Kitamura and Mizoguchi: 2004). Yet, different engineers formalize different functional conceptualizations rather than that they come up with a commonly shared formalization. Furthermore, some engineers express an ambivalent midway position by on the one hand favoring particular functional conceptualizations and on the other suggesting that the current multitude of functional conceptualizations is valuable to engineering as well (e.g., see Umeda and Tomiyama: 1997; Deng: 2002). This ambivalence is illustrated by work on the translation of functional descriptions across functional modeling frameworks (see Kitamura et al.: 2007). This research aims to facilitate communication across frameworks, thus assuming the value of co-existing functional conceptualizations (for otherwise, why engage in translation work?). Yet these translations are developed in terms of one specific (overarching) conceptualization of function, thus also suggesting that there is a single best notion of function after all.

Common to the above perspectives is that one does not find a focused debate to fix a best notion of function and/or model of functional decomposition. Given that developments towards conceptual uniformity are hard to find in engineering, the usage side-by-side of different notions of function and models of functional decomposition model may thus be a phenomenon that will persist. If so, this (again) seems different from science. Co-existence of different conceptualizations of scientific key concepts is often a temporary phenomenon and often

2000), or the burgeoning work and debate in neuroscience to establish shared conceptualizations of key concepts such as neuron, action potential, and neurotransmitter (see Kandel et al.: 1991).

followed by displacement or, alternatively, (partial) integration (e.g., see Kuhn: 1970). Such dynamics are in general not detectable in the engineering function/functional decomposition case. Why is that so? What are the factors underlying the current (and persisting) status quo that different engineers use different notions of function and/or models of functional decomposition side-by-side? Co-existence of different conceptualizations of scientific key concepts spawned extensive debate among philosophers of science on whether scientists' choices of competing key conceptualizations can be considered rational (e.g., see Kuhn: 1977). What about rationality in the engineering case? Can engineers' choices for different functional conceptualizations be considered rational from an instrumental point of view? I will address these questions in the engineering context of functional decomposition. The (possible) response that different engineers use different models of functional decomposition as a result of variation in engineering objectives does not provide an answer: I will show that different engineers choose different functional decomposition models to attain the same objective (see chapter 5).[7] But why, then, are different models used side-by-side in engineering? My principal goal in this thesis is to explain this phenomenon. That is:

(I) To explain why different models of functional decomposition are used side-by-side in engineering.

The phenomenon that different scientific theories, models, and/or explanations are advanced for the same objective has in the philosophy of science been extensively analyzed under the (related) headings of *incommensurability* and *rationality of theory choice*.[8] In this thesis I explain what we may call the *co-existence* of engineering models of functional decomposition in terms of and by expanding on certain insights from this body of work. I focus this research on the following questions:

---

7 Also when objectives and models would relate one-to-one, this would, given the key role of functional decomposition models, still be anomalous: in science key concepts do not seem to co-vary with objectives. For instance, although different contemporary neuroscientific theories address different objectives the concept of neuron is the same across these theories.

8 E.g., see Kuhn: 1970, 1977, 1983, 1991; Hempel: 1983; Laudan: 1987; Worrall: 1988; Hoyningen-Huene, 1992; Sankey: 1995, 2002; Teller: 2008.

(1) What sort of functional decomposition models are used side-by-side?
(2) What are the differences and commonalities between them; what are the possibilities and impossibilities of translating/cross-communicating functional decomposition models?
(3) What leads engineers to choose specific functional decomposition models?
(4) Have each of these functional decomposition models specific advantages or, instead, is there one model that fares significantly better than others?
(5) Is it to be expected that co-existence of functional decomposition models will persist?

The answers given to these questions impact both the relevance and feasibility of engineering attempts to cross-communicate functional descriptions. If one model would on all relevant aspects be better than the rest, the usage of different models seems from a practical/instrumental point of view non-rational. And attempts to translate models then seem misguided. Also, if their content would be radically different, translation attempts seem doomed to fail. With respect to these issues, I will advance the position that there is significant overlap in the content between models, and that it is instrumentally rational to use and persist in using different models side-by-side. Given this overlap in content I also pursue a second goal in this thesis:

(II) To improve the communication of functional decomposition models across functional modeling frameworks.

I address these issues in the context of functional modeling in the electro-mechanical domain, focusing my analyses on those concepts that are central in engineering to describe technical artifacts: purpose or goal, action, function, behavior, and structure (see Vermaas: 2009; see also Brown and Blessing: 2005). This focus is both empirical and a-historical: I analyze and compare empirical models of functional decomposition as advanced in contemporary engineering.

In this thesis I explicitly seek engagement with two different audiences: engineers and philosophers. I take the question why models of functional decomposition co-exist to be of interest to both engineers and philosophers. The answer developed to this question provides a careful reflection on the practical benefits of current engineering practice to use different models side-by-side (see chapter 4). Issues relating to co-existing and competing conceptualizations receive (and have received) substantial attention in philosophy of science, and have also gotten attention in philosophy of engineering (see chapter 5). The communication problems that co-existence of models engenders seems a topic primarily of concern to engineers: it is a challenge that engineers acknowledge and attempt to deal with (chapters 6-7). My two research objectives – explaining co-existence and improving cross-communication – reflect my aim to engage both audiences.

1.2. Incommensurability and rationality in engineering

In this thesis I apply the analysis of scientific theory choice in terms of incommensurability to engineering, rather than to science. I show that one can then explain the co-existence of engineering models of functional decomposition in terms of the Kuhnian thesis of *methodological incommensurability*. Broadly speaking, two different notions of incommensurability can be distinguished in Kuhn's work. *Semantic* incommensurability holds between competing theories when kind terms of such theories cannot be translated into one another. Such translation failure occurs when the classification schemes underlying these respective theories classify the same objects into different kinds, which are (taken to be) subject to incompatible laws.[9] The thesis of *methodological* incommensurability asserts that there is no commonly shared algorithm available on the basis of which scientists can unambiguously choose between competing scientific theories. Rather, epistemic standards that scientists' use to evaluate and choose theories vary between rival theoretical frameworks.[10] Kuhn (1977) pressed the point that such standards do not function as algorithmic rules by which one is able to determine theory choice but, rather, as *values* guiding such choices. Epistemic values refer to characteristics or properties of scientific theories that are considered desirable by scientists relative to their objectives.[11] For instance, a scientist that considers it important that a theory is able to explain a broad range of phenomena (scope) will choose the available theory that satisfies that value best. Yet another scientist may find it important that all the

9 See Kuhn: 1991; cf. Hoyningen-Huene, 1992; Sankey, 1999.

10 See Kuhn: 1970, 1977; Sankey: 1999; Carrier: 2008; Soler: 2008; Oberheim and Hoyningen-Huene: 2009.

11 See Kuhn: 1977; see also McMullin: 1982; Laudan: 1987; Sankey: 2002.

processes postulated by a theory are empirically verifiable and, hence, will choose the available theory that satisfies that value best. Now, based on such divergent assessments of the merits of scientific theories different scientists may choose different theories: scope may favor one theory, yet empirical verifiability another.[12] Based on this construal of theory choice in terms of values and the observation that scientists (can) differ in the values they employ, Kuhn (1977) concluded that scientists may rationally disagree in theory choice.[13]

1.2.1. Engineering functional decomposition and methodological incommensurability

Of these two notions of incommensurability, I use the thesis of methodological incommensurability to explain the side-by-side usage or co-existence of models of functional decomposition. The answers I develop to the first question of what sort of models are used side-by-side and the second question of whether there are possibilities to translate them, indicate that semantic incommensurability does not apply in the functional decomposition case. I will show that there is overlap in the content or conceptualizations behind the functional descriptions used in these models. Let me briefly elaborate. In addition to the three mentioned notions of *behavior function*, *effect function*, and *purpose function*, I define a fourth one: *action function*. Action functions are used to characterize intentional behaviors that agents carry out when using artifacts. For instance, the manual insertion of a screw into the screw bit of a screwdriver. Based on these four notions of function I regiment models as advanced in engineering into four different functional decomposition models (see chapters 6-7):

- Functional decomposition as a model of an organized set of behavior functions

---

12 I adopt this particular example from Sankey (1995).

13 To be sure, Kuhn (1977) pressed the point that there is no *commonly* shared algorithm capable of dictating theory choice. This leaves open the possibility that it is possible to formulate such an algorithm within a scientific framework. Yet, such an algorithm will be constructed in terms of the values that scientists working within a particular framework have. Since values can vary across frameworks, such algorithms will likewise vary. Hence, Kuhn's claim that there is no common algorithm shared by all scientists involved in the debate (see Sankey: 1995).

- Functional decomposition as a model of an organized set of effect functions
- Functional decomposition as a model of an organized set of purpose functions
- Functional decomposition as a model of an organized set of behavior functions and action functions

Now, I argue that there is common ground across functional modeling frameworks. This common ground is to be found in concepts like behavior, effect, and action that underlie notions of behavior function, effect function, and action function. These conceptualizations behind notions of functions are shared by different modeling frameworks, albeit in differ ways. For instance, I argue that the notion of behavior function as advanced in some models corresponds with the concept of behavior as used in other frameworks. Similarly, an action function corresponds with the concept of user action in other frameworks. I relate behavior functions and effect functions as follows: effect function descriptions refer to particular features of behavior function descriptions, namely their effects. Behavior function descriptions are more elaborate and refer to both effects and to behavioral features by which these effects are brought about. For instance, a behavior function of a lamp may be described as 'converting electricity into light and heat' whereas an effect function may be described as 'producing light'. The former description highlights that the effect of having light results from a conversion of electricity. In the latter description this information is excluded, merely representing the effect of having light. Finally, purpose function descriptions refer to the final result or outcome of behaviors. Say, 'having illumination in a room'.[14] These correspondences and relationships

[14] The distinction between effect and purpose function is not completely clear-cut: both relate to features of behavior. Yet, purpose function descriptions, such as 'having illumination in a room', are, typically, phrased in terms of a result of behavior in the environment of a technical artifact. Effect function descriptions, such as 'producing light', are phrased in terms of behavioral features of a technical artifact (this distinction originates from Chandrasekaran and Josephson (2000) who distinguish between device-centric and environment-centric descriptions of functions). Behavior functions can be distinguished clearly from effect and purpose functions: in behavior function descriptions physical conservation laws are taken into account, whereas this is not the case in effect and purpose function descriptions (e.g., see chapters 2, 4, 6). For instance, in the description 'producing light', the conservation of energy is not taken

indicate that it is, in principle, possible to translate and/or relate functional descriptions across frameworks. Hence, semantic incommensurability does not apply to engineering models of functional decomposition.

In addition, the (third) question that concerns me is why different engineers choose different models of functional decomposition. Semantic incommensurability in itself is silent on choice considerations. The thesis of methodological incommensurability does provide tools to analyze choice considerations, both in science and, as I will show, in engineering. In particular, I employ and expand on Kuhn's notion that one can explain divergence of theory choice in terms of variation in values. Applying this notion to engineering, I argue that engineers' choices for particular models of functional decomposition are influenced by the values that they employ in model choice. Let us call these values *engineering values*. I define an engineering value as a characteristic or property of a functional decomposition model or a functional decomposition strategy that is considered desirable by an engineer relative to an objective.[15] [16] I argue that depending on the engineering value(s) that an engineer considers important, he/she chooses a model that satisfies that value best. And, as I will argue, there is not one model that satisfies all these engineering values best. Rather, different models satisfy different engineering values best. Hence, since engineers differ in the engineering values that they employ in model choice, different engineers choose different models. This analysis answers our question of what leads engineers to choose specific models of functional decomposition. I demonstrate

into account. In 'converting electricity into light and heat' the input energy of electricity is supposed to equal the output energy of light, thus taking physical law into account.

15 Applying the notion of variation in values to engineering rather than science is unproblematic. Values are not specific to science (e.g., see McMullin: 1983). In different contexts, in casu science and engineering, values convey the (same) idea that a characteristic or property of an item or entity is considered desirable. The more discriminative notions of epistemic value and engineering value, of course, are specific to these contexts and relate to different items: epistemic values relate to scientific theories, and engineering values relate to functional decomposition models or strategies.

16 Whereas Kuhn's (1977) notion of a value refers to a characteristic of a scientific theory that is considered desirable by a scientist, I broaden this notion in the engineering case. Engineering values may refer to both characteristics of models and strategies. Sankey (2002) advances, in the science case, a similar maneuver by distinguishing 'standards' from 'rules': standards refer to desirable features of scientific theories and rules refer to prescriptions for doing scientific research. For instance, the rule to avoid ad hoc hypotheses (see Sankey: 2002). In Sankey's view both standards and rules impact the choice for scientific theories.

this analysis in terms of two case studies. In the first case study the models are not in competition. In the second case study different models are advanced for a common objective.

In the first case study, I address the question why different engineers use different functional decomposition models (rather than one model) in terms of engineers' usage of (archived) design knowledge in the construction of models of functional decomposition (see chapter 4). Whether or not knowledge of known connections between functions and structures is employed in functional decomposition strategies is one of the most explicit features on which functional decomposition strategies diverge (see chapter 3).

I argue that engineers' choices for particular functional decomposition models are influenced by the specifics of the design knowledge that they employ in the construction of these models. Depending on these specifics, particular models are chosen by engineers to achieve their main *objectives*. I define a main objective as the main goal or state of affairs that an engineer aims to achieve with a functional decomposition model.[17] I explain, for instance, that when engineers pursue the objective of innovative design and adopt the engineering value that known function-structure connections are not employed in model construction, models of behavior functions are chosen (see Pahl and Beitz: 1988; Stone and Wood: 2000).[18] [19] Such behavior function descriptions contain the requisite information to select (potentially novel) structures after the model is built, whereas effect and purpose functions descriptions are too course-grained to do so in a precise manner. In another example, I explain that when engineers pursue the objective of design analysis and embrace the engineering value to

17 Engineers typically pursue multiple objectives with a particular sort of model (e.g., see Stone and Wood: 2000). A main objective is thus relative to a specific context in which an engineer advances his/her model.

18 In this analysis I use the characterization by Pahl and Beitz (1988) and Stone and wood (2000) of this objective: the designing of new artifacts that have potentially novel (combinations of) function-structure connections.

19 Throughout this thesis I use the terms structure or design solution interchangeably. To be precise, by these terms I mean a specific conceptualization, like behavior or function conceptualizations, of technical artifacts or artifacts-to-be-designed, and not artifacts themselves. Design knowledge, such as function-structure and behavior-structure connections, thus refers to relationships between concepts. In Takeda et al. (1990) the notion of design knowledge as relationships between concepts is analyzed in more detail. I thank Tetsuo Tomiyama for insisting upon being clear about the distinction between a structure and a conceptualization of a structure.

employ known behavior-structure connections in model construction, models of effect functions are chosen.[20] Given that behavior and structure are known, effect function descriptions contain the requisite information to verify whether behaviors achieve functions in the intended fashion. For instance, the purpose function 'illumination in a room' may be sufficient for determining whether an artifact's behavior achieves this function. Yet it does not provide information to assess whether the function is achieved by its underlying behavior as intended. Say, the switch connected to a lamp might be 'off' whilst the light is still on. An effect function description such as 'switch on-light on' does contain the information to assess these behavioral features. Based on such examples I advance the point that knowledge usage specifics in functional decomposition strategies, which correspond to different engineering values, influence the choice for particular models. This explains why different engineers use different models of functional decomposition rather than one model.

Yet, in this case the models are not in competition: different engineering values relating to knowledge usage co-vary with objectives. Hence, this case does not establish that different engineers with different engineering values will choose different models to attain the same objective.[21]

In the second case study the models are in competition. In this case different engineers advance different models of functional decomposition to achieve a *common* objective of routine designing. The proponents of these models characterize this objective as the designing of new artifacts by using knowledge of function-structure connections of existing types of the to-be-designed artifact (see chapter 5). Also in this case engineering values vary, leading to different model choices. I will discuss two examples here. For instance, some engineers

20 In this analysis I use the characterization of design analysis that can be found in Bell et al. (2007): verifying whether the functions of an artifact are achieved by the behaviors of an artifact in the intended manner.

21 One can argue in hypothetical fashion for this case though. For instance, when engineers pursue the objective of innovative design and would adopt the different engineering value that known-function structure connections are employed in model construction, they will likely choose other models than models of behavior functions. When functions in a model are already connected to structures, engineers can opt for more course-grained models of effect or purpose functions. In such cases more elaborate models of behavior functions are unnecessarily complex to select (potentially novel) structures for an artifact-to-be. I submit this example as a plausible one. In the second case study that I discuss there is actual (rather than merely hypothetical) competition between models (see text).

deem it important that the realization of a function by a structure does not depend on the (prior) realization of another function by another structure (see Deng et al.: 2000; Deng: 2002).[22] Given this engineering value, I explain why these engineers choose models of purpose functions. Such models allow one to conceive most clearly of the realization of functions as being independent from the realization of other functions. For instance, realization of the behavior function 'transmitting torque' of an electric screwdriver requires, say, prior realization of the behavior function 'converting electricity into torque' (both behavior functions are sub functions of an electric screwdriver's function of 'loosen/tighten screws'). Similarly, realization of the electric screwdriver's effect (sub) function 'produce torque' requires, say, prior realization of the effect sub function 'generate electricity'. In contrast, realization of the purpose function of, say, 'having a rotational force', which is a sub function of the screwdriver's function 'to connect materials', is more easily conceived as independent from the realization of other sub functions. Hence, models of purpose functions satisfy this engineering value best.[23]

Other engineers consider it important that a functional decomposition defines a configuration of design solutions or structures in which all structures are compatible with one another, so that all the functions in the model are realized (see Chakrabarti and Bligh: 2001). Given this engineering value, I explain why models of behavior functions are chosen. Such models contain the requisite information to assess whether the output characteristics of one structure's sub function are compatible with the input characteristics of another structure's sub function. Say, the heat generated when energy is converted into torque of an electric screwdriver's motor may negatively interact with the electric wiring connected to the motor. This may cause the 'transmitting electricity' sub function of the electric wiring to fail (and hence, both the motor's sub function

---

22 This is also known as modularization. The idea behind this engineering value is that a change to a structure – for instance, a structure that gets broken or is replaced by a different type of structure – affects just one function rather than several ones.

23 The realization of particular behaviors of technical artifacts requires the prior realization of certain other behaviors (see Umeda et al.: 1996; Deng; 2002). The realization of particular effects of behaviors thus also seems to depend on the prior realization of certain other behaviors and their effects. Since purpose functions are typically phrased in terms of a result of behavior in the environment of a technical artifact rather than in terms of behavioral features of a technical artifact (see note 13), such descriptions are best suited to conceive of the realization of functions as being independent from the realization of other functions.

and the function of the screwdriver to 'loosen/tighten screws' as well). Models of effect and purpose functions seem too course-grained to satisfy this engineering value. For instance, the effect (sub) function 'produce torque' of a screwdriver's motor does not contain the information required to assess its compatibility with the electrical wiring.[24]

Based on such examples I argue that engineering values influence the choice for particular models. Given that engineering values vary, so do engineers' choices for functional decomposition models.[25] And since the models are in competition, I submit that this case exemplifies an instance of methodological incommensurability in the engineering domain. As we will see in the next section, such cases engender questions concerning the practical rationality of engineers' usage of different models side-by-side.

1.2.2. Engineering functional decomposition and practical rationality

In a similar vein as Kuhn (1970, 1977, 1983) explained scientists' choices for different theories in terms of differences in epistemic values, I thus offer an explanation why different models of functional decomposition are used side-by-side in engineering in terms of variation in engineering values. Kuhn's analysis of values, in addition, led him to conclude that scientists' choice of (competing) theories can be considered rational. This conclusion has spawned extensive debate in philosophy of science.[26] Initially, a key issue was whether in the absence of a commonly shared algorithm scientists' choice of theories can in fact be considered rational (see Kuhn: 1977). More recently, this debate has shifted in orientation: both advocates of a single method for theory choice and authors that accept variation in values are pressed to show that their preferred single method

---

24 To be sure, the constraint that all structures are compatible with one another (and, hence, all functions realized) is of course a crucial constraint that is valued in all functional modeling frameworks. Modeling frameworks differ, however, in which design phase this value is to be satisfied. In some accounts functional decomposition models should satisfy this value (see Chakrabarti and Bligh; 2001), whereas in others it should be satisfied in later design phases (see Deng et al.: 2000; Deng: 2002). Thus, only in some accounts is it a value that applies to functional decomposition models.

25 I do not claim that the engineering values that I consider exhaust the list of values that engineers might have. The ones that I consider suffice to meet my goal to explain co-existence of models.

26 E.g., see Kuhn: 1977; McMullin: 1983; Laudan: 1987; Worrall; 1988: Sankey; 1995, 2002.

or spectrum of values ensure the rationality of scientists' choice of theories (see Worrall: 1988; Sankey: 1995, 2002). The challenge is to show that the method or values one considers are appropriate ones for the evaluation and choice of scientific theories. In the case of values, a value is considered appropriate for theory choice if a theory that satisfies a particular value contributes to the attainment of a scientific objective (that one aims to achieve with the theory) precisely because the theory satisfies that value (see Hempel: 1983; McMullin: 1983; Sankey: 2002). Stated differently, that the (desired) characteristics or properties of the chosen theory indeed are the features by means of which the theory contributes to attainment of an objective that one aims to achieve with the theory.[27]

I will address this issue in the engineering functional decomposition case. In order to answer the fourth question whether specific models have specific advantages or, rather, that one specific model is significantly better than the competition I need to demonstrate that the engineering values that I consider are appropriate ones for the evaluation and choice of models. I thus need to demonstrate that a model that satisfies a particular engineering value contributes to the attainment of the objective for which the model is used precisely because it satisfies that engineering value, i.e., that the (desired) characteristics or properties of the chosen model indeed are (among) the features by means of which the model contributes to attainment of the objective for which it is used. Otherwise, if the engineering values I considered are not suited for the evaluation of models, I cannot assess whether or not specific models have specific advantages. By implication, I then cannot indicate whether the usage of different models by different engineers is rational from a practical point of view. If, say, a model should satisfy the engineering value that 'the minimum number of functions in the model is thirty', the chosen model may contribute to an objec-

27 Both semantic and methodological interpretations of incommensurability have spawned extensive debate on both the rationality of scientists' choice of theories and on the very possibility to rationally compare competing theories (e.g., see Sankey: 1995, 2002). Some commentators took semantic incommensurability to imply that there is no rational comparison possible between the content of alleged (semantically) incommensurable theories (see Oberheim and Hoyningen-Huene: 2009). Others, instead, argue that there can be a great deal of overlap in content between theories and/or models of rival paradigms, making content comparisons across theories and/or models possible (see Teller: 2008). I argue, as indicated earlier, that in the engineering functional decomposition case there is overlap in the content across modeling frameworks, thus supporting content comparisons.

tive of, say, design analysis, but the model will not contribute to this objective because the model satisfies this value. A characteristic such as 'a minimum number of thirty functions in a model' is an inappropriate instrument to evaluate the benefits of models and assess whether their use is rational from a practical point of view.

Several authors aim to distinguish appropriate from inappropriate epistemic values in terms of the (aforementioned) idea that appropriate ones contribute to the attainment of scientific objectives. When theories contribute to the attainment of scientific objectives because they satisfy certain values, these values are considered appropriate.[28] For instance, a theory that satisfies the value that it does not contain ad hoc hypotheses conduces to the scientific objective of maximizing the falsifiability of theories. Similarly, a theory that satisfies the value that it is consistent with findings from related disciplines conduces to the scientific objective of increased plausibility. And insofar as values are appropriate, maintaining variation of these values in theory choice is considered rational. Advocates of value variation consider such means-end interpretations of values an asset (see Laudan: 1987; Teller: 2008). It allows for the possibility to rationally compare the merits of competing theories (or scientific models): this theory/model is better with respect to this value, that theory/model is better with respect to that value.

Several interpretations of such means-end relationships between epistemic values and scientific objectives are given in philosophy of science. Some assert that appropriate values contribute to the attainment of a main or ultimate objective of science, such as empirical adequacy or truth (see McMullin: 1983). Others do not invoke the notion of an ultimate objective and argue that specific values contribute to more specific objectives (see Laudan: 1987; Teller: 2008). For instance, the relationship between ad hoc hypotheses and falsifiability as mentioned above. Sankey (2002) gives a third interpretation by combining the two interpretations above. Sankey views these more specific objectives as subordinate to a main or ultimate objective of science, which in his book is advancement on the truth. He takes the achievement of subordinate objectives as sub serving this main or ultimate objective of science. According to Sankey one can only understand why scientists employ different values when the

[28] See Kuhn: 1983; McMullin: 1983; Laudan; 1987; Hoyningen-Huene: 1992; Sankey; 2002; Teller; 2008.

objectives served by these values contribute, when achieved, to a common ultimate objective for which theories are advanced. In Sankey's scheme one thus finds epistemic values, their related subordinate objectives, and a common ultimate objective.[29]

I use Sankey's (2002) interpretation of means-end relationships between epistemic values and scientific objectives to clarify that the engineering values that I consider are appropriate ones for the evaluation of functional decomposition models (see chapter 5). Specifically, I use his distinction between subordinate objectives and a main objective to clarify that the different engineering values I consider are appropriate ones for the evaluation of models. In the engineering case, I speak of sub objectives rather than subordinate ones. By introducing the distinction between sub objectives and a main objective I can make clear how different models that satisfy different (and conflicting) values can all contribute to a common main objective.

Sankey (2002) asserts that one can evaluate whether an epistemic value contributes to a scientific objective by invoking empirical evidence from the history of science. If there is statistical covariance between the past use of a value and the achievement of an objective, the value is appropriate. I do not have such empirical evidence available in the engineering functional decomposition case. I will hence follow a different tack. I attempt to make it plausible that models contribute to the objectives for which they are used because they satisfy particular engineering values (and thus make it plausible that such engineering values are appropriate ones for the evaluation of models). To this end, I use the following (admittedly weaker) strategy: examples and argument.[30] I argue by way of examples that models satisfying the engineering values that I consider are

---

[29] This, of course, invites the question how competing theories can both advance on the truth. It seems that Sankey (2002) wants to stave of methodological incommensurability: he asserts that one can choose that theory which satisfies all these values best. Teller (2008) rejects the idea that science is in the truth-business and seems to suggest that there is no ultimate or common objective. All (competing) scientific models are idealizations that are advanced depending on specific values. If there is no common objective, however, it seems not appropriate to speak about incommensurability as Teller does. I do not flesh this issue out.

[30] The engineering values and objectives that I consider are derived from the engineering literature. Although I strongly believe, given the expert knowledge that engineers possess, that these values are appropriate ones for the evaluation of models I cannot use this belief as a premise in an argument that purports to show that they are appropriate. I would then presuppose what I am trying to demonstrate.

suitable means to achieve the objectives for which they are used. As said, to demonstrate this suitability I distinguish between main and sub objectives of engineers. In the engineering literature one can find explicit statements of main objectives, such as innovative design or routine design, and, if looking more closely, also of sub objectives. These sub objectives relate more directly (as desired ends) to engineering values than main objectives do. By explicating sub objectives the link between engineering values and main objectives becomes clearer as well. I define sub objectives as by-engineers-desired states of affairs that they aim to achieve by models that satisfy particular engineering values. A model satisfying a specific engineering value thus is supposed to contribute to a particular sub objective. I argue that sub objectives in turn, when achieved, contribute to the attainment of main objectives.[31] So, models satisfying particular engineering values directly contribute to sub objectives and indirectly, via the realization of sub objectives, to main objectives.

For instance, consider the earlier-mentioned engineering value that the realization of a function by a structure does not depend on the (prior) realization of another function by another structure. I argue that if this engineering value is satisfied by a functional decomposition model (of purpose functions) the model is conducive to the sub objective of having 'broad range of function-structure mapping' (see Deng: 2002). If function-structure connections can be considered independent from other function-structure connections, one can search the available spectrum of design solutions to a given function. If the realization of a function by a structure would depend on the (prior) realization of another function by another structure, the range of structure-function connections would decrease. A selection of a particular design solution to a function would then constrain the possible design solutions one can choose for functions that must be realized prior to this function. For instance, if one selects a specific type of water valve to realize the sub function of, say, 'regulate influx of water' (which is a sub function of the function 'to wash laundry' of a washing machine) this may constrain the types of water supply hoses one can select to realize the sub function 'transport water' (which is another sub function of the 'washing laundry' function). Both structures are to be compatible for realization of their functions. By considering function-structure connections as independent, this

[31] This analysis in terms of sub objectives thus makes it insightful why there occurs variation in engineering values in cases where different models are in competition.

constraint does not apply. Hence, a broad range of function-structure connections can be considered. Models satisfying the above engineering value thus contribute to having broad range of function-structure connections.[32] In turn, having a broad range of function-structure connections is a sub objective that, when achieved, contributes to a main objective of, in this example, routine design. By keeping the range of function-structure connections broad, chances increase that one can select the most adequate candidate structure for a function (e.g., one that is cheap or easily manufactured). And if a chosen structure turns out inadequate one can replace it with a single other one (rather than that such a replacement implicates the effort of changing other structures as well as is the case when function-structure connections are dependent). Having a broad range of function-structure connections thus supports routine design.

On the other hand, I argue that a model (of behavior functions) that satisfies the (earlier mentioned) engineering value that the design solutions or structures defined by the model are compatible with one another contributes to the sub objective of 'having all functions in the model realized' (see Chakrabarti and Bligh: 2001). Attainment of this sub objective, in turn, of course supports the main objective of routine design.

Based on such examples I argue that different models, precisely because they satisfy particular engineering values, directly contribute to the attainment of particular sub objectives and indirectly, via the achievement of sub objectives, to main objectives. Therefore, I submit that the engineering values that I consider are appropriate ones for the evaluation of models. By means of engineering values we can thus address the fourth question whether specific models have specific advantages. Yes, they have: examples such as given above show that depending on the values that engineers have, specific models are better than others. And, in addition, models can be rationally compared with respect to a particular value. For instance, if one values compatibility of structures, then one better opts for a model of behavior functions; if one values independence of function-structure connections, one better picks a model of purpose functions. There is not one model that satisfies all engineering values best. Hence, I submit that the usage of different functional decomposition models by different engineers is rational from a practical point of view.

---

32 As mentioned in note 24, compatibility of structures is of course vitally important. However, Deng et al. (2000) take care of this constraint in later design phases, after models are built.

Returning to my principal goal (I) of explaining co-existence, the main *pointe* I aim to advance in this thesis is that one can understand the co-existence of engineering models of functional decomposition in terms of the idea that different models satisfy different engineering values, all of which are relevant to engineering. And, hence, that the usage side-by-side of different models of functional decomposition is rational from an instrumental point of view. Given this result, I conjecture with respect to the fifth question that it is also likely that co-existence will persist.

My second case study of methodological incommensurability in the case of routine designing both expands on earlier analyses of co-existing conceptualizations as given in the philosophy of engineering, as well as on my first analysis of variation in design knowledge employment (see chapter 5). I interpret the analyses of Bucciarelli (1994, 2003) and Vermaas (2009) as relating specific conceptualizations (as suitable means) to specific objectives, thus explaining (and validating) co-existing conceptualizations in engineering in terms of a variety of engineering objectives. The incommensurability analysis shows that different functional decomposition models are also advanced in engineering as means for the same objective. Hence, such cases cannot be explicated with the explanatory construct of variety of engineering objectives as (I interpret them to be) advanced in the analyses of Bucciarelli (1994, 2003) and Vermaas (2009).[33] My explanation in terms of engineering values does cover these cases. In addition, my incommensurability analysis covers cases that my other analysis in terms of variation in design knowledge usage does not accommodate. In the case of routine designing, all models are built using knowledge of existing function-structure connections. And this knowledge already contains or refers to specific notions of function and/or specific functional decomposition models. Now, given that this knowledge already implies a notion of function and/or specific functional decomposition model, one can easily indicate which models are constructed and chosen. However, such an analysis is both non-informative and

[33] It seems that the objectives Bucciarelli (1994, 2003) and Vermaas (2009) have in mind are what I coined sub objectives. For instance, Vermaas (2009) argues that specific meanings of the concept of function are used in engineering to advance specific descriptions of technical artifacts. He asserts that all these descriptions are useful to engineering. One could argue that having a specific description of a technical artifact is a sub objective that is useful or contributes to the achievement of a certain main objective.

circular. In the incommensurability analysis I consider other engineering values than ones related to knowledge usage. This analysis does clarify engineers' choices for particular models in those cases in which the models are built in terms of known function-structure connections.

1.2.3. Cross-communicating functional decomposition models

In addition to these philosophical analyses I aim, as said, to engage engineers. Given the results that both comparisons of the content of models are feasible and that the use of different models side-by-side has practical engineering value, I pursue a second goal in this thesis: to improve the communication of functional decomposition models across frameworks. Broadly speaking, one can identify two positions in engineering on dealing with the communication problems that the co-existence of models engenders. One, solving communication problems by adopting a commonly shared functional conceptualization. Two, addressing communication problems by developing translations of functional descriptions across frameworks.

The first position attempts to build up a commonly shared functional modeling framework that "can serve as a general and common communication framework" (see Erden et al.: 2008, 147).[34] Yet, at the same time these authors acknowledge that there are incompatibilities between functional modeling accounts. However, in order to build up a general and common framework, this suggests that a number of functional modeling accounts need to be replaced. For otherwise, when such incompatibilities will remain in place, the prospects for establishing a general and common framework seem slim. Based on my analysis that different engineers have different engineering values, I submit that the chances that engineering in general will adopt such a common framework are small. In addition, I argue that replacing a number of modeling accounts has disadvantages. I develop this argument in terms of my first case study on engineering values related to the use of (archived) design knowledge (see chapter 4). By opting for a single framework, in which only specific engineering values are adopted, one only has a specific model available to achieve (main) objectives. Yet, more importantly, one cuts off the possibility to achieve a number of sub

[34] Which they aim to do, mainly, in terms of the functional conceptualizations of Umeda et al. (1996) and Chandrasekaran and Josephson (2000).

objectives when opting for a single framework. For instance, returning to the case discussed on pages 9 and 10 (and footnote 21), consider two ways in which different engineers may use functional decomposition models to support a common (main) objective of innovative designing, i.e., the designing of new artifacts that have potentially novel (combinations of) function-structure connections. When an engineer considers it important that known connections between functions and structures are not employed (which is an engineering value) in the construction of models, this engineer chooses, as argued, a model of behavior functions. And a model (of behavior functions) satisfying this engineering value supports the sub objective of 'avoiding bias toward known function-structure connections' (see Stone and Wood: 2000). This sub objective, when achieved, supports innovative design: without such bias potentially novel function-structure connections come in view. Now, when another engineer uses a model of, say, effect functions that is constructed based on known connections between effect functions and structures (which is an engineering value), this model (when satisfying this engineering value) supports the sub objective of 'revealing when known knowledge of function-structure connections is insufficient to take care of all required functionalities' (see Chakrabarti and Bligh: 2001). This sub objective, when achieved, also supports innovative design: it indicates that new knowledge on function-structure connections is required (see Chakrabarti and Bligh: 2001). Use of these different models thus contributes to the achievement of different sub objectives, which cannot be achieved with a single model. Avoiding bias cannot be achieved with a model that is based on known function-structure connections. Revealing that known functions-structure connections are insufficient cannot be achieved with a model that is not build in terms of known function-structure connections. Hence, by opting for a single framework, rather than co-existence of models, a spectrum of sub objectives cannot be achieved. Therefore, I submit that one better not opt for a single framework: co-existence of models has instrumental value.

By taking this position I do take a research challenge aboard: handling cross-communication problems across functional modeling frameworks (which would be solved when fixing a single commonly shared framework).[35] My second goal in this thesis, therefore, is to improve the communication of functional descrip-

[35] As mentioned in section 1.1, the co-existence of different functional conceptualizations is hampering information exchange in engineering, both between engineers and engineering computer systems.

tions across frameworks. The feasibility of such a project is grounded in my analysis that there is overlap in content between the functional terms used in different functional decomposition models (see chapters 6-7). Hence, the translation failure obstacle of semantic incommensurability can, in principle, be overcome. To my knowledge, one methodology has been advanced in engineering that is specifically focused on translating models of functional decomposition across functional modeling frameworks (see Kitamura et al.: 2007; Ookubo et al.: 2007). In this thesis I review this methodology and develop conceptual improvements to the translation steps proposed in it (see chapters 6-7). In this methodology models are translated in terms of one specific (overarching) conceptualization of function and functional decomposition. This overarching conceptualization, laid down in what they call a "reference ontology", is used to identify notions of function and functional decomposition model as advanced in engineering (see Kitamura et al.: 2007, 2). Based on these identifications, translations of models across frameworks are developed. However, I argue that this overarching conceptualization collapses distinctions between notions of function and functional decomposition models. Resultantly, in the translation process, functional information is both changed and lost. For instance, action functions in particular models are re-interpreted as effect functions, leading to information change. And functional decomposition models of behavior functions are translated under the assumption that they are models of effect functions. This leads to information change and also information loss. Those features of to-be-translated models that are incompatible with the (effect) functional conceptualization of the reference ontology are removed in the translation process (see Ookubo et al.: 2007).[36] I argue that my regimenting of models into four different functional decomposition models allows me to highlight and bypass these problems. As mentioned, I argue that the notion of behavior function as advanced in some models corresponds with the concept of behavior as used in other frameworks. Similarly, an action function corresponds with the concept of user action in other frameworks. And behavior functions and effect functions can be related in the sense that effect functions descriptions

[36] By re-interpreting functions, certain aspects of the meaning of these functions are lost as well of course. I use the distinction between information loss and meaning change in the sense that in the former case information is removed/no longer represented in translated models; in the latter case information is still represented in translated models but misinterpreted/misclassified.

refer to particular features of behavior function descriptions, namely their effects. Behavior function descriptions are more elaborate and refer to both effects and to behavioral features by which these effects are brought about. Finally, purpose function descriptions refer to the final result or outcome of behaviors (in the environment of a technical artifact). Using these correspondences and relationships I translate and/or relate functional descriptions across frameworks, without the above mentioned information loss.

The basic idea of my alternative conversion strategy is to first translate functional decomposition models of behavior functions into models of (organized sets of) physical behaviors. This step preserves behavior function information. In a second step the relevant effects of the behaviors in translated models are abstracted. These effect descriptions are then used to construct functional decomposition models of effect functions and/or models of purpose functions. This second step relates different functional decomposition models. These results achieve my second goal to improve the communication of functional decomposition models across frameworks.

What more do analyses of engineering functional decomposition have on offer? Both De Ridder (2006) and Vermaas and Garbacz (2009) identify a connection between engineering functional decompositions and the philosophical literature on mechanistic explanations. Definitions of 'mechanism' vary across accounts of mechanistic explanation (see chapter 8), but the basic idea, colloquially put, is that a mechanism is an enduring system that regularly exhibits a behavior. And this overall behavior of the mechanism results from the coordinated behaviors of its component parts. The manner in which the component parts are organized spatially is crucial for the performance of the behaviors by these component parts. And the performance of these behaviors is likewise constrained by the manner in which they are organized temporally. Hence, spatial-temporal organization is crucial for the mechanism to perform its overall behavior. Mechanistic explanations purport to specify and make insightful how behaviors and component parts are organized temporally and spatially such that together they instantiate the overall behavior of the mechanism.[37] [38]

---

37 Mechanistic explanations are mainly discussed in terms of examples drawn from (neuro)biology.

38 I phrase the general idea of mechanism(s) and mechanistic explanation here in terms of 'behavior' and 'component part'. Terms (and differences in meaning) vary across accounts:

De Ridder (2006) takes the notion of mechanistic explanation as starting point for advancing an account of what he coins “mechanistic artifact explanation” (81). Mechanistic artifact explanations purport to show how the overall behavior of a technical artifact results from the behaviors of the component parts of the artifact.[39] Whereas De Ridder took a direction from mechanistic explanations to technical artifacts, on the way expanding on certain mechanistic notions to meet his goals, I submit that proceeding in the opposite direction leads to results as well. In this thesis I present such a (first) step in the opposite direction, going from engineering functional decomposition to mechanistic explanation. In the context of mechanistic explanations, the concept of function is pivotal to make insightful how a mechanism performs its overall behavior (see Craver: 2001). And the connection between function and mechanistic organization is crucial for the ascription of functions to the behaviors of component parts (see Craver: 2001). Yet, it is also acknowledged that the notion of mechanistic organization needs a more sustained analysis (see Wright and Bechtel: 2006). I demonstrate this insight by showing that Craver’s treatment of function, i.e., how he relates this concept to mechanistic organization, leads to (several) problems, amongst which that contributions between different functions of an item cannot be explicated in detail (see chapter 8). I invoke a specific engineering model of functional decomposition to remedy this problem. Since many different engineering models of functional decomposition are advanced, with specific ways in which the functions in such models are organized, one can envision that invoking these models will further contribute to an analysis and extension of the notion of mechanistic organization. My analysis grounds the prospects of such a strategy.

This thesis is organized as follows. Each chapter consists of an article that is either published or currently submitted for publication.

---

some speak of behaviors, others of operations, interactions, functions, or activities; some speak of component parts, others of working parts, entities, or structures. Since I merely intend to provide the general flavor here, I ignore these differences.

39 In mechanistic artifact explanation reference is also made to a context of human action (see De Ridder: 2006). This sets mechanistic artifact explanations apart from mechanistic explanations in (neuro)biology.

Chapter 2 provides a systematic survey of engineering accounts toward functional decomposition that compares different functional decomposition models as advanced in the surveyed accounts.[40]

Chapter 3 consists of a survey that compares different functional decomposition strategies as advanced in the surveyed accounts, focusing on the use of design knowledge in building models of functional decomposition.[41]

Chapter 4 presents the analysis in which the choice for (and suitability of) particular functional decomposition models (for their objectives) is shown to be influenced by whether or not particular design knowledge is employed in their construction.[42]

In chapter 5 the thesis of methodological incommensurability is extended to the engineering domain and the means-end analysis of engineering values presented.[43]

In chapter 6 translations are presented of four different models of functional decomposition.[44]

In chapter 7 one such translation of models referring to organized sets of behavior functions and user functions is spelled out in more detail.[45]

---

[40] Van Eck, D. (2009) 'On Relating Functional Modeling Approaches: Abstracting Functional Models from Behavioural Models', in: *Proceedings of the International Conference on Engineering Design (ICED 09), 24-27 August 2009, Stanford, CA, USA*: 2.89-2.100.

[41] Van Eck, D., McAdams, D.A., and Vermaas, P.E. (2007) 'Functional Decomposition in Engineering: A Survey', in: *Proceedings of the ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE), September 4-7, 2007, Las Vegas, Nevada, USA*. DETC2007-34232: 1-10.

[42] Van Eck, D. (2010) 'Explaining and Relating Different Engineering Models of Functional Decomposition', in: *Proceedings Design Research Society (DRS) International Conference, 07-09 July 2010, Montréal, Canada*. 122.1-122.11.

[43] Van Eck, D. (Forthcoming 2011) 'Incommensurability and Rationality in Engineering Design: The Case of Functional Decomposition'. *Techné: Research in Philosophy and Technology*.

[44] Van Eck, D. (2010) 'Supporting Design Knowledge Exchange by Converting Models of Functional Decomposition'. Submitted Manuscript.

In chapter 8 the concept of function in Craver's account of mechanistic explanations is analyzed, and the notion of mechanistic organization extended by invoking a specific engineering model of functional decomposition.[46]

A few caveats are in order. First of all, given the paper-based character of the thesis a certain degree of repetition occurs between chapters. Secondly, there are some terminological differences between the papers. In earlier written papers (chapters 2, 3, and 7) I speak of "functional model" whereas in more recent papers (chapters 4, 5, and 6) I use the more precise term of "functional decomposition model". Both terms refer to the same thing: a graphical representation of an organized set of functions.
Furthermore, in chapter 2 I distinguish two notions of function: "behavior" and "intended behavior". In more recent work (chapters 4, 5, and 6) I distinguish these two notions by the more precise terms of "behavior function" and "effect function" (and also introduce two other notions of function: "purpose function" and "action/user function"). Both the terms "behavior" and "behavior function" refer to desired behavior of a technical artifact. And both the terms "intended behavior" and "effect function" refer to desired effects of behavior of a technical artifact. Thirdly, some papers contain analyses that are rudimentary or contain superseded elements when measured against other (more recent) papers. In chapter 2 I suggest that a specific representational scheme for function indicates a specific meaning or notion of function. To wit: that operation-on-flow-representations of functions refer to behaviors. This leads me to (erroneously) conclude that in the functional modeling account of Lind (1994) functions refer to behaviors. This one-to-one correspondence between representation and notion or meaning of function is incorrect however. In later work (chapters 4, 5, and 6) this error is corrected. There I argue that the notion of function advanced by Lind (1994) is an effect function. In addition, I advise the reader to skip the rough characterization of a translation strategy that I present at the end of

---

45 Van Eck, D. (2010) 'On the Conversion of Functional Models: Bridging Differences Between Functional Taxonomies in the Modeling of User Actions', in: *Research in Engineering Design* 21 (2): 99-111. DOI 10.1007/s00163-009-0080-7.

46 Van Eck, D. (2010) 'Mechanisms, Functional Hierarchies and Levels of Explanation'. Submitted Manuscript.

chapter 2 and instead focus on the translation strategy that I spell out in chapters 6 and 7.

Please also heed the following more important caveat: in chapter 4 I argue that the choice for particular functional decomposition models is influenced by whether or not particular design knowledge is employed in their construction. In addition, I also argue that the choice to employ specific design knowledge is affected by specific design objectives, arriving at the conclusion that the choice for and suitability of models of functional decomposition depends on the design objectives for which they are employed. However, further research shows that this conclusion is too strong (see the discussion on pages 10-12, and 19). Therefore, this chapter should be read as advancing the claim that the choice for particular functional decomposition models is influenced by whether or not particular design knowledge is employed in their construction. In other words, knowledge usage in the construction of models is the crucial parameter in this analysis to explicate the choice for models of functional decomposition. I incorporate this more recent insight in chapter 5. Also, I do not phrase choices for knowledge usage in chapter 4 in terms of engineering values. I do so in chapter 5.

1.3. Discussion

Will the current co-existence of engineering models of functional decomposition persist in the future? From my analyses it does not automatically follow that functional modeling research will not eventually converge toward a single and commonly shared notion of functional decomposition model. What the analyses do show is that there are valid reasons available not to do so, and my bet is that it also will not happen: if steps toward convergence would be taken up in general, functional modeling researchers will in all likelihood become more explicit on their choices for particular models and thus more explicitly consider the merits of specific models of functional decomposition. I conjecture that such considerations will lead to a spectrum of what I coined engineering values and sub objectives. And that the majority view will be one of persisting in using different models side-by-side in order to be able to satisfy all these engineering values and sub objectives. Another related issue is whether the modeling field will eventually settle on a best notion of each of the four functional decomposition models that I considered. Given the current plethora of functional modeling accounts, it may turn out at some point in the future that the current situation is then interpreted as, say, ‘pre-paradigmatic’, and consensus is reached on the most adequate way to represent each of the four models of functional decomposition. I bet that this scenario is unlikely as well: I expect that such considerations also lead to the establishment of a spectrum of engineering values and sub objectives, which turn out to be satisfied best when particular representational variants of the considered models of functional decomposition are all kept aboard. For instance, in a design analysis context it might be relevant to represent functions in a model in terms of triggering conditions and behavioral effects, say ‘switch on-light on’, rather than in verb-noun fashion, say, ‘produce light’. If the ‘lighting-function’ of a room light fails, the former but not the latter description contains information of a possible source of this failure, say a broken switch. In other contexts, verb-noun descriptions may do a better job. Say, in the initial design phase of a room light when one wants to keep choices on how to operate the artifact still open, the description ‘produce light’ is more apt.

Regarding translations of functional decomposition models, two steps can be envisioned that (naturally) succeed my analyses. Firstly, I considered four functional decomposition models. It may turn out that yet other models emerge from other analyses, which then would require additional research into the translation of such models. Secondly, a next step is to investigate how to automate translations of functional decomposition models in computer tools. One may anticipate that usage of translation schemes such as mine will drive a new set of research questions, such as how to implement translation instructions in unambiguous fashion in these computer tools. Taking this automation step requires specific computer science skills and knowledge, in addition to conceptual schemes. Such work is beyond the scope of this thesis and would have to be carried out by experts possessing the relevant knowledge and skills.

Acknowledgements

This research is part of the VIDI research program *Functional Decomposition in Philosophy and Engineering*, supported by the Netherlands Organization of Scientific Research (NWO).

References

Bell, J., Snooke, N., and Price C. (2007) 'A Language for Functional Interpretation of Model Based Simulation', in: *Advanced Engineering Informatics* 21: 398-409.

Brown, D.C. and Blessing, L. (2005) 'The Relationship between Function and Affordance', in: *Proceedings of the ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE '05, September 24-28, Long Beach, California, USA*. DETC2005-85017:1-6.

Bucciarelli, L.L. (1994) *Designing Engineers*, MIT Press, Cambridge MA.

Bucciarelli, L.L. (2003) *Engineering Philosophy*, Delft University Press, Delft.

Carrier, M. (2008) 'The Aim and Structure of Methodological Theory', in: Soler, L., Sankey, H. and Hoyningen-Huene, P. (eds.), *Rethinking Scientific Change and Theory Comparison: Stabilities, Ruptures, Incommensurabilities?*, Springer, Dordrecht: 273-290.

Chakrabarti, A. (1998) 'Supporting Two Views of Function in Mechanical Designs', in: *Proceedings 15th National Conference on Artificial Intelligence, AAAI'98, July 26-30, 1998, Madison, Wisconsin, USA*.

Chakrabarti, A. and Bligh, T. P. (2001) 'A Scheme for Functional Reasoning in Conceptual Design', in: *Design Studies* 22: 493-517.

Chittaro, L. and Kumar, A. N. (1998) 'Reasoning about Function and its Applications to Engineering', in: *Artificial Intelligence in Engineering* 12: 331-336.

Chandrasekaran, B. (2005) 'Representing Function: Relating Functional Representation and Functional Modeling Research Streams', in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 19: 65-74.

Chandrasekaran, B. and Josephson, J.R. (2000) 'Function in Device Representation', in: *Engineering with Computers* 16: 162-177.

Craver, C.F. (2001) 'Role Functions, Mechanisms, and Hierarchy', in: *Philosophy of Science* 68: 53-74.

Deng, Y.M. (2002) 'Function and Behavior Representation in Conceptual Mechanical Design', in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 16: 343-362.

Deng, Y.M., Tor, S.B., and Britton, G.A. (2000) 'A Dual-Stage Functional Modelling Framework with Multi-Level Design Knowledge for Conceptual Mechanical Design', in: *Journal of Engineering Design* 11 (4): 347-375.

De Ridder, J (2006) 'Mechanistic Artefact Explanation', in: *Studies in History and Philosophy of Science* 37: 81-96.

Erden, M.S., Komoto, H., Van Beek, T.J., D'Amelio, V., Echavarria, E, and Tomiyama, T. (2008) 'A Review of Function Modelling: Approaches and Applications', in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 22: 147-169.

Far, B.H. and Elamy, A.H. (2005) 'Functional Reasoning Theories: Problems and Perspectives', in: *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing* 19: 75-88.

Hempel, C.G. (1983) 'Valuation and Objectivity in Science', in: Cohen, R.S., and Laudan, L. (eds.), *Physics, Philosophy and Psychoanalysis: Essays in honor of Adolf Grunbaum*, Reidel, Boston: 73-100.

Hoyningen-Huene, P. (1992) 'The Interrelations between the Philosophy, History and Sociology of Science in Thomas Kuhn's Theory of Scientific Development', in: *British Journal for the Philosophy of Science* 43: 487-501.

Kandel, E., Schwartz, J., and Jessell, T. (1991) *Principles of Neural Science*, Elsevier, New York.

Kitamura, Y. and Mizoguchi, R. (2004) 'Ontology-Based Systematization of Functional Knowledge', in: *Journal of Engineering Design* 15 (4): 327-351.

Kitamura, Y., Takafuji, S., and Mizoguchi, R. (2007) 'Towards a Reference Ontology for Functional Knowledge Interoperability', in: *Proceedings of the ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE '07, September 4-7, 2007, Las Vegas, Nevada, USA*. DETC2007-35373: 1-10.

Kuhn, T. S. (1970) *The Structure of Scientific Revolutions*, University of Chicago Press, Chicago.

Kuhn, T.S. (1977) 'Objectivity, Value Judgment and Theory Choice', in: Kuhn, T.S., *The Essential Tension*, University of Chicago Press ,Chicago: 320-339.

Kuhn, T.S. (1983) 'Rationality and Theory Choice', in: *Journal of Philosophy* 80: 563-570.

Kuhn, T. S. (1991) 'The Road since Structure', in: *Proceedings of the 1990 Biennial Meeting of the Philosophy of Science Association* 2: 3-13.

Laudan, L. (1987) 'Progress or Rationality: The Prospects for Normative Naturalism', in: *American Philosophical Quarterly* 24: 19-31.

Lind, M. (1994) 'Modeling Goals and Functions of Complex Industrial Plants', in: *Applied Artificial Intelligence* 8: 259-283.

McMullin, E. (1983) 'Values in Science', in: *Proceedings of the 1992 Biennial Meeting of the Philosophy of Science Association* 2: 3-28.

Oberheim, E. and Hoyningen-Huene, P. (2009) 'The Incommensurability of Scientific Theories', in: *Stanford Encyclopedia of Philosophy*: 1-27. Retrieved at 15-05-2009.

Ookubo, M., Koji, Y., Sasajima, M., Kitamura, Y., and Mizoguchi, R. (2007) 'Towards Interoperability Between Functional Taxonomies Using an Ontology-Based Mapping', in: *Proceedings of the International Conference on Engineering Design, ICED '07, August 28-31, 2007, Paris, France*: 154.1-154.12.

Otto, K.N. and Wood, K.L. (2001) *Product Design: Techniques in Reverse Engineering and New Product Development*, Prentice Hall, Upper Saddle River NJ.

Pahl, G. and Beitz, W. (1988) *Engineering Design: A Systematic Approach*, Springer, Berlin.

Rosenman, M.A. and Gero, J.S. (1999) 'Purpose and Function in a Collaborative CAD Environment', in: *Reliability Engineering and System Safety* 64: 167-179.

Sankey, H. (1995) 'The Problem of Rational Theory-choice', in: *Epistemologia* 18 (2): 299-312.

Sankey, H. (1999) 'Incommensurability: An Overview', in: *Divinatio: Studia Culturologica Series* 10: 1-13.

Sankey, H. (2002) 'Methodological Pluralism, Normative Naturalism and the Realist Aim of Science', in: Nola, R., and Sankey, H. (eds.), *After Popper, Kuhn and Feyerabend: Recent Issues in Scientific Method*, Kluwer, Dordrecht: 211-229.

Soler, L. (2008) 'Introduction', in: Soler, L., Sankey, H. and Hoyningen-Huene, P. (eds.), *Rethinking Scientific Change and Theory Comparison: Stabilities, Ruptures, Incommensurabilities?*, Springer, Dordrecht:1-17.

Stone, R. B. and Wood K.L. (2000) 'Development of a Functional Basis for Design', in: *Journal of Mechanical Design* 122: 359-370.

Szykman, S., Fenves, S.J., Keirouz, W., and Shooter, S.B. (2001) 'A Foundation for Interoperability in Next-Generation Product Development Systems', in: *Computer-Aided Design* 33: 545-559.

Takeda, H., Hamada, S., Tomiyama, T., and Yoshikawa, H. (1990). 'A Cognitive Approach to the Analysis of Design Processes', in: Rinderle, J. (ed.), *DE-Vol. 27, Design Theory and Methodology DTM ' 90*, ASME: 153-160.

Teller, P (2008) 'Of Course Idealizations are Incommensurable!', in: Soler, L., Sankey, H. and Hoyningen-Huene, P. (eds.), *Rethinking Scientific Change and Theory Comparison: Stabilities, Ruptures, Incommensurabilities?*, Springer, Dordrecht: 247-264.

Umeda, Y. and Tomiyama, T. (1997) 'Functional Reasoning in Design', in: *IEEE Expert: Intelligent Systems and Their Applications* 12: 42-48.

Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y., and Tomiyama, T. (1996) 'Supporting Conceptual Design based on the Function-Behavior-State-Modeler', in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 10: 275-288.

Van Eck, D. (2009) 'On Relating Functional Modeling Approaches: Abstracting Functional Models from Behavioral Models', in: *Proceedings of The 17th International Conference on Engineering Design (ICED 09), August 24-27, 2009, Stanford, California, USA*: 2.89-2.100.

Van Eck, D. (2010a) 'Explaining and Relating Different Engineering Models of Functional Decomposition', in: *Proceedings Design Research Society (DRS) International Conference, 07-09 July 2010, Montréal, Canada*: 122.1-122.11.

Van Eck, D. (2010b) 'On the Conversion of Functional Models: Bridging Differences Between Functional Taxonomies in the Modeling of User Actions', in: Research in Engineering Design 21 (2): 99-111. DOI 10.1007/s00163-009-0080-7.

Vermaas, P.E. (2009), 'The Flexible Meaning of Function in Engineering', in: *Proceedings of the 17th International Conference on Engineering Design (ICED 09), August 24-27, 2009, Stanford, California, USA*: 2.113-2.124.

Vermaas, P.E. and Garbacz, P. (2009) 'Functional Decomposition and Mereology in Engineering; in: Meijers, A (ed.), *Handbook Philosophy of Technology and Engineering Sciences*, Elsevier Science, Amsterdam: 235-271.

Worrall, J. (1988) 'The Value of a Fixed Methodology', in: *British Journal for the Philosophy of Science* 39: 263-275.

Wright, C. D. and Bechtel, W. (2006) 'Mechanisms and Psychological Explanation', in: Gabbay, D.M., Thagard, P., and Woods, J (eds.), *Handbook Philosophy of Psychology and Cognitive Science*, Elsevier Science, New York: 31-79.

2 On Relating Functional Modeling Approaches: Abstracting Functional Models from Behavioral Models

This chapter appeared as an article in the proceedings of the International Conference on Engineering Design (ICED):

Van Eck, D. (2009) 'On Relating Functional Modeling Approaches: Abstracting Functional Models from Behavioral Models', in: *Proceedings of the International Conference on Engineering Design (ICED 09), 24-27 August 2009, Stanford, CA, USA*: 2.89-2.100.

Abstract

This paper presents a survey of functional modeling approaches and describes a strategy to establish functional knowledge exchange between them. This survey is focused on a comparison of function meanings and representations. It is argued that functions represented as input-output flow transformations correspond to behaviors in the approaches that characterize functions as intended behaviors. Based on this result a strategy is presented to relate the different meanings of function between the approaches, establishing functional knowledge exchange between them. It is shown that this strategy is able to preserve more functional information than the functional knowledge exchange methodology of Kitamura, Mizoguchi, and co-workers. The strategy proposed here consists of two steps. In step one, operation-on-flow functions are translated into behaviors. In step two, intended behavior functions are derived from behaviors. The two-step strategy and its benefits are demonstrated by relating functional models of a power screwdriver between methodologies.

Keywords: Behavior, function, functional modeling, knowledge exchange

2.1. Introduction

As can be seen in a current review by Erden et al. (2008) engineering design research has produced a wealth of functional modeling approaches. In these approaches a variety of definitions of functions, representations for functions and strategies for decomposing functions into sub functions are proposed. Such different conceptualizations can however lead to cross-communication problems between engineers working with different frameworks (see Bucciarelli: 1994; Rosenman and Gero: 1999; Deng: 2002). The emerging field of engineering ontology aims to handle such communication problems by developing ontologies in which concepts relevant to the engineering sciences are formalized (see Ahmed et al.: 2007; Kim et al.: 2008). A part of this engineering ontology research consists of developing function ontologies, in which specific concepts of technical function are formalized (see Szykman et al.: 2001; Zhang et al.: 2005a; Kitamura et al.: 2005/6; Borgo et al.: 2009). These function ontologies prove useful in the storage, retrieval, and communication of functional information between engineers and engineering teams using the *same* ontology (see Kitamura et al.: 2005/6). It is however commonplace that different meanings are attached to the concept of technical function in the engineering domain (see Chittaro and Kumar: 1998; Chandrasekaran and Josephson: 2000; Far and Elamy: 2005; Van Eck et al.: 2007; Erden et al.: 2008). A methodology, developed by Kitamura et al. (2007; 2008) and Ookubo et al. (2007) is specifically aimed at bridging such different conceptions of technical functions *between* different functional taxonomies. It does so by converting functional models between functional taxonomies.

It is argued in this paper that this conversion methodology, valuable though it is, may lead to information loss, undermining its purpose of establishing functional knowledge exchange between taxonomies. In this paper an alternative strategy is formulated by which this functional information can be preserved. This alternative strategy is based on an analysis and comparison of function meanings and representations between functional modeling approaches. The approaches included in this analysis are: the Multi Level Flow modeling methodology of Lind (1994), the Reverse Engineering and Redesign methodology of Otto and Wood (1996, 1998, 2001), the Functional Basis methodology of Stone and Wood (2000), the Functional Reasoning methodology of Chakrabarti and Bligh (2001), the Dual Stage methodology of Deng, Tor, and Britton (2000a, 2000b, 2002), the Functional Concept Ontology methodology of Kitamura, Mizoguchi, and co-workers (2003, 2005/6), and the Functional Interpretation

Language methodology of Price, Bell, and Snooke (2007). In the Multi Level Flow modeling methodology, the Reverse Engineering and Redesign methodology, the Functional Basis methodology, and the Functional Reasoning methodology functions are modeled in terms of material, energy, and signal flows. In the Dual Stage methodology, the Functional Concept Ontology methodology, and the Functional Interpretation Language methodology functions characterize intended roles or abstractions of behaviors. It is argued that this distinction in functional representation formats strongly suggests a difference in function meaning. More specifically, that functions in the Multi Level Flow modeling methodology, the Reverse Engineering and Redesign methodology, the Functional Basis methodology, and the Functional Reasoning methodology correspond to behaviors or features of behaviors in the Dual Stage methodology, the Functional Concept Ontology methodology, and the Functional Interpretation Language methodology.

Taking these differences in function meaning as starting point, a two-step strategy is formulated to establish functional knowledge exchange between these approaches without information loss. In the first step, operation-on-flow functions are translated into behaviors. In the second step, intended behavior functions are derived from behavior characterizations. This strategy is demonstrated by relating functional models of a power screwdriver represented in terms of the Functional Basis, Functional Concept Ontology, and Functional Interpretation Language frameworks.

The method adopted in this paper is analytic and example-based. Concepts and assumptions that underlie the functional modeling approaches are analyzed, and the proposed strategy to relate them is illustrated by way of examples. This method has advantages and disadvantages. It is suited for elucidating concepts, but less so for empirical testing. This limitation is acknowledged in this paper and empirical validation is left with the relevant experts.

The paper has the following organization. It starts by discussing the approaches in the second section. The analysis of function meanings is presented in section three. The strategy to support functional knowledge exchange is given in section four, and illustrated with examples of different functional models of a power screwdriver. The paper ends with conclusions in section five.

2.2. Functional modeling: A survey

In this section a brief overview of the functional modeling approaches is presented. It is focused on the engineering applications and domains of the methodologies, the definitions and representations of functions that are used, and the methods for functional decomposition that are proposed. The interested reader is referred to the original papers for more conceptual and empirical details.

2.2.1. Multi Level Flow modeling methodology

The Multi Level Flow modeling methodology formulated by Lind (1994) is a functional modeling methodology that is used for modeling the goals and functions of industrial plants. The methodology is aimed at supporting diagnosis and planning tasks for plant operators and the design of plant control systems. The methodology is employed in academic research projects in several universities. In 2002, Larsson (2002) stated the expectation that applications based on Multi Level Flow modeling will be brought into industrial practice within the next ten years.

In this methodology, functions describe behaviors of components that are useful for achieving goals (see Lind: 1994; Larsson: 1996). Overall functions are represented by natural language terms. Sub functions are represented as operations on material, energy, or information flows. These operations are selected from a predefined set of operations, coined functional concepts, for these flows. Operations on material and energy flows represent the mass and energy processes occurring in plants. Operations-on- information flows represent operations of control systems or activities of plant operators that are aimed at making or counteracting changes in plant states.

In Multi Level Flow models the goals, functions, and physical components of plants are represented. The decomposition of a goal into sub goals is the starting point for a functional decomposition. Based on this goal decomposition, sub functions that achieve the sub goals are ascribed to a system and specified in a functional decomposition. These sub functions are represented as operations-on-flows and linked to physical components that implement them. Sub functions in a functional decomposition are grouped together into mass, energy, or information flow structures. Flow structures consist of functions connected by flows. Goals, functions, and physical components are connected in terms of three types of relations (see Lind: 1994; Larsson: 1996). An "achieve relation" connects a set of functions to a goal, indicating that the goal is achieved by the set of functions. A "condition" relation connects a goal to a function, indicating that the goal must be achieved first in order for the function to be achieved. A "realization" relation connects physical components to functions, indicating the components that realize the functions (see Lind: 1994, 267).

2.2.2. Reverse Engineering and Redesign methodology

The Reverse Engineering and Redesign methodology formulated by Otto and Wood (1996, 1998, 2001) is a methodology that is aimed at facilitating the redesign of existing products. In this methodology product redesign consists of three phases: a first reverse engineering phase, a second modeling and analysis phase, and a third redesign phase. Functional modeling is used in the reverse engineering phase. The methodology is focused on the electromechanical and mechanical domains. In an academic setting, the methodology is taught at two U.S. universities (see Otto and Wood: 1998).

In this methodology, an overall product function is defined as a reproducible relationship between available input and desired output (see Otto and Wood: 2001). This overall function is described in verb-noun format and represented by a black-boxed operation on flows of materials, energies, and signals. Sub-functions are also described in verb-noun format and represented by operations on material, energy, or signal flows. Sub functions can correspond to either "device functions" or "user functions" (see Otto and Wood: 1998, 229). Device functions are defined as operations carried out by products, and user functions are defined as customer activities during product usage. A common set of operations and a common set of flows, developed by Little et al. (1997), are used to represent sub functions.

The reverse engineering phase of the methodology starts with describing the overall hypothesized function of a product. This overall function is represented as a black-boxed operation on flows of materials, energies, and signals. In a second step customer needs are gathered and inventoried for the product. In a third step, a process description or activity diagram is developed. An activity diagram specifies customer activities during usage of the product (see Otto and Wood: 2001). Based on this activity diagram, characteristics of the product's functional model are chosen. These characteristics include the system boundary, parallel and sequential chains of sub functions and interactions between device functions and user functions. In a fourth step, using the activity diagram and gathered customer needs, a functional model for the product is hypothesized and developed. The development of a functional model starts with identifying major flows associated with the customer needs. A sequence of sub functions, a function chain, is then described for each of these flows that consists of device functions and sometimes also user functions. Aggregating these function chains then completes the functional model. In a later step in the reverse engineering phase, the actual product is disassembled into its components and a functional model is developed in which the actual sub functions of the product's components are represented. This actual model is then compared with the hypothesized functional model. The aim of this comparison is to help design teams understand different physical principles by which a product can operate.

2.2.3. Functional Basis methodology

The Functional Basis methodology formulated by Stone and Wood (2000) is an approach to functional modeling that is aimed at creating a common and consistent functional design language, dubbed a functional basis. This language allows designers to model overall product functions as sets of interconnected sub functions. The Functional Basis approach is focused on, especially, the electromechanical and mechanical domains. The approach is presented as supporting the archiving, comparison, and communication of functional descriptions of existing products, and the engineering designing of new products. Since the approach was proposed it has been developed further. It is, for instance, used to develop a method to identify modules from functional models (see Stone et al.: 2000). It is also used to build a web-based repository in which functional decompositions of existing products are archived as well as the design solutions for the sub functions that are part of these decompositions (see http://function.basiceng.umr.edu/delabsite/repository.html)

In this approach, an overall product function refers to a general input/output relationship defined by the overall task of the product. This overall product function is described in a verb-object form and represented by a black-boxed operation on flows of materials, energies, and signals. A sub function, describing a part of the product's overall task, is also described in a verb-object form but represented by a well-defined basic operation on a well-defined basic flow of materials, energies, or signals. The black-boxed operations on general flows representing product functions are derived from customer needs, and the basic operations and basic flows representing sub functions are laid down in common and limited libraries that span the functional design space. These libraries are called a *functional basis*.

Stone and Wood (2000) present a three-step methodology to develop functional models or functional decompositions of products. The methodology starts with describing a product function in a verb-object form, represented by a black-boxed operation on flows of materials, energies, and signals. A chain of operations-on-flows is then specified, called a function chain, for each black box input flow, which transform that flow step-by-step into an output flow. These operations-on-flows are selected from the functional basis libraries. Finally, these temporally ordered function chains are aggregated into a single functional model of a product.

2.2.4. Functional Reasoning methodology

The Functional Reasoning methodology developed by Chakrabarti and Bligh (2001) is a methodology that is aimed at supporting engineering design of new products. They present what they call a "functional reasoning model" (see Chakrabarti and Bligh: 2001, 506) to support the transformation of functional design requirements into schematic descriptions of design solution concepts. This reasoning scheme is aimed at assisting computational design tasks by providing a formal model of the conceptual design process and a common language in which functions and design solution concepts can be described. This reasoning scheme uses knowledge of functions and solution concepts of existing designs. The approach is focused on the mechanical domain.

In this approach, a function is defined as an effect that is required by a design problem or that is provided by a solution (see Chakrabarti and Bligh: 2001). Effects are defined as intended aspects of causal behavior (see Chakrabarti et al.: 2005). Both functions and sub functions are represented as input-output transformations of flow variables. Input and output flow variables are characterized by their kind (material, energy, or signal), orientation, direction, position, and magnitude (see Chakrabarti and Bligh: 1994).

A functional decomposition starts in a first step by defining a design problem as an overall desired function or set of functions, represented as an input-output transformation of flow variables (see Chakrabarti and Bligh: 2001). A sub function is then selected for function-structure mapping. It is required that sets of known technical solutions that can solve the sub functions of the overall function are available. In the second step different technical solutions for this sub function are selected, and the first found technical solution is chosen. Technical solutions are chosen when their input characteristics match the input characteristics of the overall function. After choosing a technical solution for a sub function, the output characteristics of the chosen solution become the input characteristics of the remaining design problem. This leads to a revision of the overall function: the revised overall function is represented as an input-output transformation in which the input corresponds to the output of the chosen solution, and the output corresponds to the output of the original overall function. In the third step it is evaluated which functional requirements of the revised overall function still need to be solved. In the fourth step another sub function of the revised overall function is selected, and alternative technical solutions are selected that can solve the sub function. The first found technical solution is chosen, which again leads to a revision of the overall function and an evaluation of the remaining unsolved functional requirements. This decomposition process continues until technical solutions for all sub functions are found, resulting in a configuration of technical solutions that can solve the overall desired function as defined in the first step. In the fifth step, this process goes back one step and another solution for the last mapped sub function is selected. This leads to an alternative revised function. This process is reiterated until all possible configurations of technical solutions that can solve the overall function have been found.

2.2.5. Dual Stage methodology

The Dual Stage methodology developed by Deng, Tor, and Britton (2000a, 2000b, 2002), is an approach to functional modeling that is aimed at supporting the conceptual phase of product design in the mechanical domain. It is presented as supporting functional descriptions of designs and the identification of design solution concepts. The approach has also been used to build functional knowledge bases for automated design support systems (see Zhang et al.: 2001; Zhang et al.: 2002, 2005b) and to build function ontologies for support of knowledge exchange in collaborative design environments (see Zhang et al.: 2005a).

In the Dual Stage approach, following a distinction made by Chakrabarti (1998), two types of functions are defined: purpose functions and action functions (see Deng: 2002). A purpose function is defined as a description of the designer's intention or the purpose of a design. An action function is defined as an abstraction of intended and useful behavior of an artifact. Behaviors refer to the physical interactions between the components of a design, or to the interactions between the design and its environment. Purpose functions are represented in natural language terms. Action functions are either represented in natural language terms or as input-output transformations, in which the input and output represent a physical interaction. Overall functions correspond to purpose functions, and sub functions correspond to purpose functions or to action functions.

A functional decomposition starts in a first stage by decomposing a purpose function into sub-functions, which are usually also purpose functions (see Deng: 2002). These sub functions are then mapped onto technical solutions. The sub functions corresponding to action functions that cannot be mapped onto technical solutions are further developed in a second stage. The sub functions corresponding to purpose functions that cannot be mapped onto technical solutions are, in this first stage, mapped onto action functions. This mapping is done either by using stored knowledge on specific mappings, or by using libraries that store "physical phenomena" (see Deng et al.: 2000a; Deng, 2002, 350). Physical phenomena refer to behavioral processes, the physical structures realizing these behavioral processes and the effect(s) of these behavioral processes. The effect of a behavioral process corresponds to and is retrieved as an action function. Action functions that can achieve the purpose sub functions are then selected. The second stage starts by mapping the action functions onto technical solutions. This is done by finding the causal behavioral processes that can instantiate the action functions. Physical phenomenon libraries are again employed to find and select these behavioral processes and the technical solutions that instantiate them. After these two stages, the identified technical solutions are assembled and then it is verified whether they realize all functional design requirements.

2.2.6. Functional Concept Ontology methodology

The Functional Concept Ontology methodology developed by Kitamura and Mizoguchi (2003) and Kitamura et al. (2005/6) is an approach to functional modeling that is aimed at facilitating the sharing of engineering functional knowledge. In this approach, a set of modeling guidelines and a functional modeling language has been developed to assist the systematic and reusable description of functional models of devices. The approach supports various tasks. It is for instance employed in building an ontology for functions and in developing an automated design support system (see Kitamura and Mizoguchi: 2003).

In this approach, behavioral models and functional models of devices are developed concurrently. Behaviors of devices and their components are defined as input-output relations between operand states. Operands refer to energy, fluid, material, motion, force, or information. Behaviors are represented as input-output state changes of properties of operands. Both overall functions and sub functions of devices are defined as roles played by behaviors intended by designers or by users. Functions and sub functions are represented in terms of verb-operand pairs. The functional modeling language used in this approach consists of a generic set of verbs, called functional concepts (see Kitamura and Mizoguchi: 2003; Kitamura et al.: 2005/6).

In a functional decomposition a set of sub functions is specified that realize the overall function. Sub functions and overall functions are represented in terms of functional concepts. In a functional decomposition it is furthermore specified by means of which technical principles the sub functions achieve the overall function. These specifications are referred to as "way of achievement" (see Kitamura and Mizoguchi: 2003, 157).

2.2.7. Functional Interpretation Language methodology

The Functional Interpretation Language methodology developed by Price, Bell, and Snooke (2007) is an approach to functional modeling that is aimed at supporting design analysis tasks. The methodology is based upon the functional modeling approach for design analysis developed by Price (1998). The functional interpretation language approach is presented as supporting analysis tasks such as failure mode and effect analysis, sneak circuit analysis and design verification. The approach has been used in industry for interpreting electro-mechanical, hydraulic, and fluid-transfer systems (see Bell et al.: 2007). In this approach functions for devices are defined as follows:

> an object O has a function F if it achieves an intended goal by virtue of some external trigger T resulting in the achievement of an external effect E" (see Bell et al.: 2007, 400)

A function is represented in terms of three elements: the purpose of the function, the trigger associated with the function and the effect associated with the function. States of a function are represented by assigning truth-values to the triggers t and the effects e of the function. This allows four possible states of a function to be described:

(1) inoperative, expressed as: In $(f) \leftrightarrow \neg\, t \,\&\, \neg\, e$
(2) failed, expressed as: Fa $(f) \leftrightarrow t \,\&\, \neg\, e$
(3) unexpected, expressed as: Un $(f) \leftrightarrow \neg\, t \,\&\, e$
(4) achieved, expressed as Ac: $(f) \leftrightarrow t \,\&\, e$

Overall functions are represented in terms of their trigger, effect and purpose. Sub functions are either represented in terms of triggers, effects and purposes, or in terms of combinations of two of these elements (see Bell et al.: 2007). Three types of functions that combine two elements are described. One, a "purposive incomplete function" (PIF) consists of an effect and a purpose, and shares a trigger with another PIF. Two, a "triggered incomplete function" (TIF) consists of a trigger and purpose, and shares an effect with another TIF. Three, an "operational incomplete function" (OIF) consists of a trigger and effect, and does not have a purpose of its own. OIF's contribute to the overall function and its associated purpose (see Bell et al.: 2007, 404-405)

In a functional decomposition, an overall function is decomposed into either complete or incomplete sub functions, depending on the type of system analysed (see Bell et al.: 2007). An overall function is decomposed when its achievement depends on more than one trigger and effect. The triggers and effects of the sub functions then replace the triggers and effects associated with the overall function. The possible states of the overall function are expressed in terms of the possible states of the sub functions. With these function types they describe four types of functional decompositions:

(1) functional decomposition into complete sub functions
(2) functional decomposition into two OIF's
(3) functional decomposition into two PIF's
(4) functional decomposition in two TIF's

2.3. Establishing function-behavior correspondences

In this section the position is developed that functions in the Multi Level Flow modeling methodology, the Reverse Engineering and Redesign methodology, the Functional Basis methodology, and the Functional Reasoning methodology may plausibly be taken to correspond to behaviors or features of behaviors in the

Dual Stage methodology, the Functional Concept Ontology methodology, and the Functional Interpretation Language methodology. This is done by analyzing assumptions on the meaning of function that are part of the latter three methodologies in sub-section 3.1, and by analyzing differences in criteria for modeling functions between the former four and latter three methodologies in sub-section 3.2. It is then shown in section 4 that this position facilitates the exchange of functional knowledge between these approaches, and is able to preserve functional information that is lost in the functional knowledge exchange methodology of Kitamura, Mizoguchi, and co-workers (see Kitamura et al.: 2007; Ookubo et al.: 2007: Kitamura et al.: 2008).

2.3.1. Behaviors and design intent

In the Dual Stage methodology input-output flow transformations are taken to correspond to behaviors. Material, energy and signal flows are regarded as attributes of behaviors (see Deng: 2000) and input-output flow transformations are interpreted as behavior representations (see Deng et al.: 2000a). Functional representations in the Dual Stage methodology in terms of actions and purposes are based on the notion that these concepts represent design intent, whereas input-output flow transformations do not. Deng et al. (2000a) state that since input-output flow transformations do not represent design intent, they do not represent artifact functionality.

In the Functional Concept Ontology methodology, behavior is distinguished from a function based on design intent (see Kitamura et al.: 2005/6). Behavior is defined as a black box input-output relationship and called *objective* in the sense that the interpretation of its input-output relation is not based upon design intent (see Ookubo et al.: 2007). Design intent is captured in terms of the role concept to specify behavioral roles. Input-output flow transformations correspond to black box input-output relationships. Since these are not described in term of the roles played by them, it is not apparent how input-output flow transformations relate to design intent, considered from the Functional Concept Ontology perspective. It can be defended that from the Functional Concept Ontology perspective they correspond to behaviors as objective black box input-output relationships. This explains the statement of Ookubo et al. (2007) that design intent is implicit in the Functional Basis approach.

The Functional Interpretation Language approach may also be interpreted to hold the position that input-output flow transformations correspond to physical behaviors. Functional descriptions in the Functional Interpretation Language approach are aimed at capturing purpose at the system level. Functional representations only represent those behavior states that are relevant for the achievement of systemic purposes. Bell, Snooke, and Price (2007) remark that what they call low-level functions, referring to an application of the Functional Basis (see Van Wie et al.: 2005), do not assist in the explanation of purpose at the system level. From this systemic viewpoint on function and purpose, low-level functions, i.e., input-output flow transformations, may be interpreted as behaviors.

These viewpoints, some more explicit than others, seem similar to the position of Chandrasekaran (2005) who argues that the modeling of functions as operations-on-flows is more aptly labeled behavioral modeling, because these functional primitives do not represent design intent.

2.3.2. Conservation laws and input-output connections between functions

Two differences in modeling criteria between the methodologies validate the analysis presented above. One, whether or not functions are modeled in accordance with physical conservation laws. Two, whether or not input-output connections between functions are modeled. In the Reverse Engineering and Redesign methodology it is required that functional models are physically valid and comply with conservation laws for material and energy flows (see Otto and Wood: 2001). Operation-on-flow representations thus accord with conservation laws. This requirement makes perfect sense when operation-on-flow representations correspond to physical behaviors. Although this requirement is not explicitly mentioned in the Functional Basis and Functional Reasoning methodologies it is plausible to assume that it also holds in these methodologies. Functional models presented in these approaches that violate conservation laws are hard to find. The Multi Level Flow modeling methodology is an exception to this requirement. Functions that represent the creation or destruction of mass and energy are described in Multi Level Flow models (see Vermaas: 2008).

In the Dual Stage, Functional Concept Ontology and Functional Interpretation Language methodologies it is not required that functional descriptions obey conservation laws. Deng (2002), for instance, describes the function of a flywheel as providing mechanical energy and the function of a battery as providing electricity. Kitamura et al. (2002) describe functions of a power plant as generating heat and generating electricity. In the Functional Interpretation Language approach, a functional description of a torch is given in terms of switch positions as triggers that achieve the effect of the light being on (see Bell et al.: 2007). In these functional descriptions energy is created, violating conservation laws (see Vermaas: 2008). The physical behavior of technical artifacts in these methodologies, of course, complies with conservation laws. This requirement is however taken care of by the concept of behavior that is introduced alongside the concept of function in these methodologies. Functional descriptions in these approaches, instead, may be taken to represent only those elements of physical behaviors that are intended or are relevant for the achievement of system purposes. Such descriptions then do not have to comply with conservation laws. This distinction, with the exception of the Multi Level Flow modeling methodology, validates the claim that input-output flow transformations may be interpreted as and corresponding to behaviors in the Dual Stage, Functional Concept Ontology and Functional Interpretation Language methodologies.

A second modeling distinction grounds this interpretation. In the methodologies that characterize functions as input-output flow transformations connections between functions are modeled in terms of flows of material, energy and signal. Output flows of preceding functions are the input flows of succeeding functions. In the Dual Stage, Functional Concept Ontology and Functional Interpretation Language methodologies, in contrast, functions are not connected by input and output flows. In these methodologies, behaviors are the units that are connected by input and output. In the Functional Concept Ontology approach, behavioral models represent connections between behaviors of components in terms of material, energy and signal operands (see Ookubo et al.: 2007). In the Dual Stage approach, behaviors are connected in sequences in which the output of preceding behaviors provides the input to succeeding behaviors (see Deng: 2002). In the Functional Interpretation Language approach, behaviors are represented as sequentially ordered state transitions (Bell and Snooke: 2004). The modeling of input-output connections between functions thus marks a distinction between the approaches in the modeling of functions. Yet it marks a commonality between functional models in which functions are represented as input-output flow transformations and behavioral models in the above three approaches. This commonality supports the view that functions qua input-output flow transformations correspond to behaviors in the Dual Stage, Functional Concept Ontology and Functional Interpretation Language methodologies.

Adopting this position has practical utility. The distinction between behaviors and intended behaviors provides a strategy to relate the different notions of function and functional model between the approaches, establishing functional knowledge exchange between them. Additionally, it is able to preserve functional information that is lost in the conversion methodology. This strategy is the topic of the next section.

2.4. Abstracting functional models from behavioral models

2.4.1. Background: model conversions and the problem of information loss

The strategy presented in this section is based upon the conversion methodology of Kitamura et al. (2007; 2008), and Ookubo et al. (2007). With this methodology knowledge exchange between functional taxonomies is aimed to be supported by converting functional models between functional taxonomies. These conversions consist of two steps. In step one, function terms are translated between taxonomies. After this first step translation step, conceptual differences between functional models of different taxonomies are explicated in step two. Modifications are then developed to reduce these conceptual differences. These modifications are aimed at reducing information loss and enhancing functional knowledge exchange. After these translation and analysis/modification steps a functional model is converted between taxonomies. This methodology has been applied by its developers to a conversion of functional models between the Functional Basis taxonomy and the Functional Concept Ontology taxonomy.

In these model conversions, functional information is however lost (see Van Eck: 2009). Most Functional Basis-functions are translated into Functional Concept Ontology-functions under the assumption that they match in meaning (see Kitamura et al.: 2007; Ookubo et al.: 2007: Kitamura et al.: 2008). Yet, by these translations, functional information attached to the concept of function in the Functional Concept Ontology approach is lost. Ways of achievement, for instance, are not represented in converted models (see Ookubo et al.: 2007). The relationship between a Functional Concept Ontology-function and its underlying behavior is lost as well. On the other hand, features that in the Functional Concept Ontology approach are characteristic of behavioral models are now part of converted functional models. The connections between functions by input-output of material, energy, and signal, for instance, are described in converted models. However, input-output connections between functions are not part of functional models in the Functional Concept Ontology approach. Behavioral models in this approach, instead, describe connections between behaviors by input-output operands (see Ookubo et al.: 2007). It is acknowledged in the conversion methodology that the above differences are sources of information loss. And that they need to be handled to avoid information loss and enhance knowledge exchange (see Ookubo et al.: 2007).

To address information loss and increase functional knowledge exchange, it is proposed in this paper to switch the order of the translation step and the analysis step, and start with the analysis step. The analysis presented in section 3 namely solves the above research challenges. Based on the distinction between physical behaviors and intended behaviors the above-mentioned conceptual differences emerge as differences between functional models and behavioral models, instead of a difference between functional models. Modifications do not need to be developed to handle these conceptual differences. As argued, physical behavior functions are conceptually distinct from intended behavior functions. This is the reason why in the Functional Concept Ontology approach both behavioral models and functional models are developed concurrently. The challenge, instead, now becomes how to relate functional models qua behavioral models with functional models qua intended behavior models without information loss. A strategy to do so is formulated in the next section. Its utility is demonstrated in section 4.3 by relating different functional models of a screwdriver.

2.4.2. An alternative method

The strategy developed here incorporates a proposal by Garbacz (2006). Garbacz (2006) has developed a logical formalization of functional decomposition in which he defines behaviors as changes of flows and functions as abstracted behaviors. He states that these definitions allow for a reconciling of functional modeling approaches that define functions as abstractions or interpretations of behaviors with functional modeling approaches that define functions in terms of input-output flow relationships. By combining his reconciliatory step of abstracting functions from behaviors with my analysis of the distinction between physical behavior functions and intended behavior functions one can imagine the following solution. The physical behavior function vs. intended behavior distinction can be handled in two steps. In step one, input-output flow transformations are translated into behaviors. In step two, the relevant parts of these physical behavior representations are abstracted and incorporated into intended behavior function descriptions.

2.4.3. Applying the method: relating behavioral and functional models

Based on my analysis one can interpret the functional models described in the Multi Level Flow modeling methodology, the Reverse Engineering and Redesign methodology, the Functional Basis methodology, and the Functional Reasoning methodology as representing behavior models. The functional models in the Dual Stage methodology, the Functional Concept Ontology methodology, and the Functional Interpretation Language methodology then describe abstractions or interpretations of behaviors, i.e., the intended, abstracted parts of behaviors. This strategy is illustrated by relating functional models of a power screwdriver represented in terms of the Functional Basis methodology, the Functional Concept Ontology methodology and the Functional Interpretation Language methodology. The Functional Basis model in Figure 1 can be taken to represent a behavior model; the Functional Concept Ontology-inspired model in Figure 2 to represent a functional model, derived from the behavioral model (I have omitted the step of translating the Functional Basis model into a Functional Concept Ontology-behavioral model); and the Functional Interpretation Language-inspired model in Figure 3 to represent a functional model of the screwdriver at the overall system level.

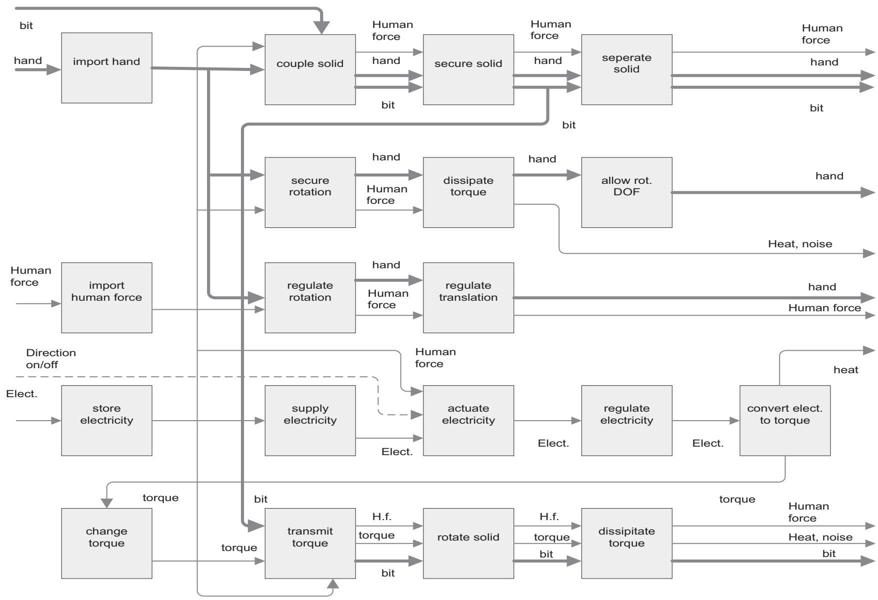

Figure 1: Functional Basis model of a power screwdriver, adopted from Stone et al. (2000).

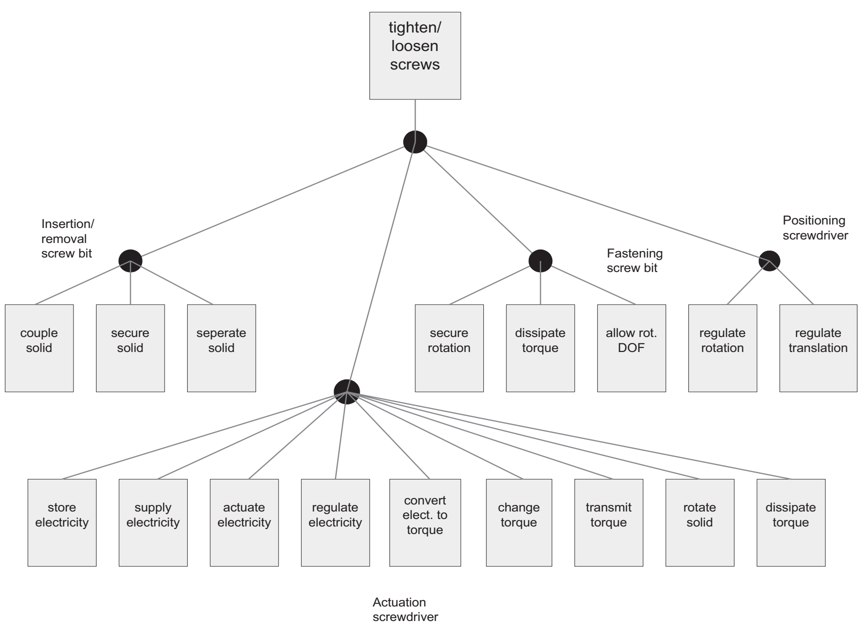

Figure 2: Functional Concept Ontology-inspired model of a power screwdriver. The overall function corresponds to the overall function of the Functional Basis-screwdriver (see Stone et al.: 2000).

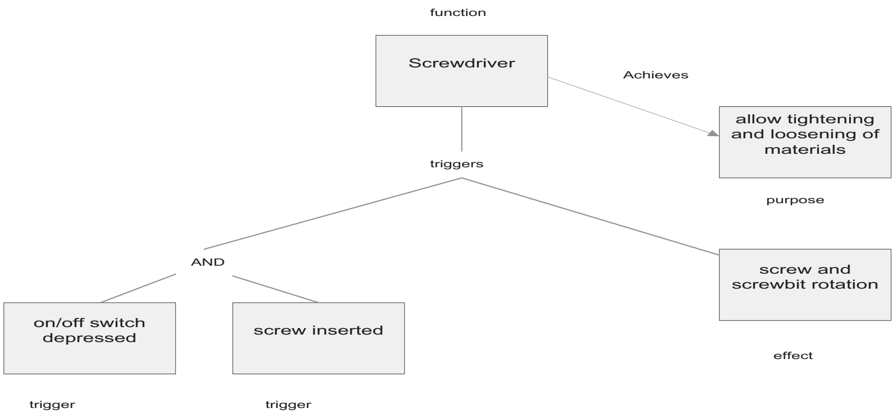

Figure 3: Functional Interpretation Language-inspired model of a power screwdriver, based upon the Functional Basis-overall function of the screwdriver. The sub functions share both an effect and a purpose, but have separate triggers.

These three models together provide a layered perspective on the behaviors, intended roles of these behaviors, and overall intended systemic behavior states of a power screwdriver. The sub functions in the functional concept ontology-inspired model are represented according to their grouping in function chains in the functional basis model. The functionality of the function chains is also described (see Stone et al.: 2000). Since Functional Basis "import" and "export" operations have no counterparts in the Functional Concept Ontology taxonomy (see Ookubo et al.: 2007), the Functional Basis functions "import hand" and "import human force" are not described in this model.

It can be seen that functional information is preserved with this strategy. The relation between Functional Concept Ontology-functions and behaviors is restored. If needed, ways of achievement can be added to the Functional Concept-Ontology model without introducing a conceptual difference between Functional Basis-functional models and Functional Concept Ontology-functional models. In addition, conceptual differences in connections between functions are addressed. Connections between functions are now features of the Functional Basis model understood as a behavioral model. By deriving the Functional Concept Ontology model from the Functional Basis model, functional knowledge exchange is thus established without information loss. This derivation step can be repeated. A Functional Interpretation Language model is derived from the overall function of the Functional Concept Ontology model, and represented by triggers and effects.

In sum, by reversing the translation and analysis steps of the conversion methodology and adding the abstraction step of Garbacz (2006), these functional modeling frameworks can be related and functional knowledge exchange established between them, without information loss.

2.5. Conclusions

In this paper a survey is presented of functional modeling approaches and a strategy is formulated to establish functional knowledge exchange between them. The position is developed that functions represented as input-output flow transformations can be taken to correspond to behaviors in the approaches that characterize functions as intended roles of behaviors or abstractions of intended behaviors. Based on this position a strategy is then presented to relate the different meanings of function and establish functional knowledge exchange between the approaches. It is shown that with this strategy functional information can be preserved that is lost in the functional knowledge exchange methodology of Kitamura, Mizoguchi, and co-workers. The strategy proposed here consists of two steps. In step one, operation-on-flow functions are translated into behaviors. In step two, intended behavior functions are derived from behaviors. The two-step strategy and its benefits are demonstrated by relating functional models of a power screwdriver between methodologies.

Acknowledgements

I would like to thank Pieter Vermaas and two anonymous reviewers for their helpful comments and suggestions. This research is supported by the Netherlands Organization of Scientific Research (NWO).

References

Ahmed, S., Kim, S., and Wallace, K.M. (2007) 'A Methodology for Creating Ontologies for Engineering Design', in: *Journal of Computing Information Science and Engineering* 7: 132-140.

Bell, J. and Snooke, N. (2004) 'Describing System Functions that Depend on Intermittent and Sequential Behaviour', in: *Proceedings of 28th International Workshop on Qualitative Reasoning, 2004, Wales, UK*: 51-57.

Bell, J., Snooke, N., and Price C. (2007) 'A Language for Functional Interpretation of Model Based Simulation', in: *Advanced Engineering Informatics* 21: 398-409.

Borgo, S., Carrara, M., Garbacz, P., and Vermaas, P.E. (2009) 'A Formal Ontological Perspective on the Behaviors and Functions of Technical Artifacts', in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 23: 3-21.

Bucciarelli, L.L. (1994) *Designing Engineers*, MIT Press, Cambridge MA.

Chakrabarti, A. (1998) 'Supporting Two Views of Function in Mechanical Designs', in: *Proceedings 15th National Conference on Artificial Intelligence, AAAI'98, July 26-30, 1998, Madison, Wisconsin, USA*.

Chakrabarti, A. and Bligh, T.P. (1994) 'An Approach to Functional Synthesis of Solutions in Mechanical Conceptual Design. Part1: Introduction and Knowledge Representation', in: *Research in Engineering Design* 6: 127-141.

Chakrabarti, A. and Bligh, T.P. (2001) 'A Scheme for Functional Reasoning in Conceptual Design', in: *Design Studies* 22: 493-517.

Chakrabarti, A., Sarkar, P., Leelavathamma, B., and Nataraju, B.S. (2005) 'A Functional Representation for Aiding Biomimetic and Artificial Inspiration of New Ideas', in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 19: 113-132.

Chandrasekaran, B. (2005) 'Representing Function: Relating Functional Representation and Functional Modeling Research Streams', in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 19: 65-74.

Chandrasekaran, B. and Josephson, J.R. (2000) 'Function in Device Representation', in: *Engineering with Computers* 16: 162-177.

Chittaro, L. and Kumar, A. N. (1998) 'Reasoning about Function and its Applications to Engineering', in: *Artificial Intelligence in Engineering* 12: 331-336.

Deng, Y.M. (2002) 'Function and Behavior Representation in Conceptual Mechanical Design', in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 16: 343-362.

Deng, Y.M., Tor, S.B., and Britton, G.A. (2000a) 'A Dual-Stage Functional Modelling Framework with Multi-Level Design Knowledge for Conceptual Mechanical Design', in: *Journal of Engineering Design* 11 (4): 347-375.

Deng, Y.M., Tor, S.B., and Britton G.A. (2000b) 'Abstracting and Exploring Functional Design Information for Conceptual Mechanical Product Design', in: *Engineering with Computers* 16: 36-52.

Erden, M.S., Komoto, H., Van Beek, T.J., D'Amelio, V., Echavarria, E, and Tomiyama, T. (2008), 'A Review of Function Modelling: Approaches and Applications', in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 22: 147-169.

Far, B.H. and Elamy, A.H. (2005) 'Functional Reasoning Theories: Problems and Perspectives', in: *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing* 19: 75-88.

Garbacz, P. (2006) 'A Formal Model of Functional Decomposition', in: *Proceedings of the 2006 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, September 10-13, 2006, Philadelphia, Pennsylvania, USA*. DETC2006-99097:115-124.

http://function.basiceng.umr.edu/delabsite/repository.html

Kim, S., Bracewell, R., and Wallace, K.(2008) 'Some Reflections on Ontologies in the Engineering Domain', in: *Proceedings of TMCE 2008 Symposium, April 21-25, 2008, Izmir, Turkey*: 1271-1284.

Kitamura, Y. and Mizoguchi, R. (2003) 'Ontology-Based Description of Functional Design Knowledge and its Use in a Functional Way Server', in: *Expert Systems with Applications* 24: 153-166.

Kitamura, Y., Koji, Y., and Mizoguchi, R. (2005/2006) 'An Ontological Model of Device Function: Industrial Deployment and Lessons Learned', in: *Applied Ontology* 1: 237-262.

Kitamura, Y., Takafuji, S., and Mizoguchi, R. (2007) 'Towards a Reference Ontology for Functional Knowledge Interoperability', in: *Proceedings of the ASME 2007 International Design Engineering Technical Conferences &*

Computers and Information in Engineering Conference, IDETC/CIE '07, September 4-7, 2007, Las Vegas, Nevada, USA. DETC2007-35373: 1-10.

Kitamura, Y., Sano, T., Namba, K., and Mizoguchi, R. (2002) 'A Functional Concept Ontology and its Application to Automatic Identification of Functional Structures', in: *Advanced Engineering Informatics* 16 (2): 145-163.

Kitamura, Y., Segawa, S., Sasajima, M., Tarumi, S., and Mizoguchi, R. (2008) 'Deep Semantic Mapping Between Functional Taxonomies for Interoperable Semantic Search', in: Dominique, J. and Anutariya, C (eds.), *The Semantic Web*, Springer, Berlin:137-151.

Larsson, J.E. (1996) 'Diagnosis Based on Explicit Means-End Models', in: *Artificial Intelligence* 80: 29-93.

Larsson, J.E. (2002) 'Diagnostic Reasoning Based on Means-End Models: Experiences and Future Prospects', in: *Knowledge-Based Systems* 15: 103-110.

Lind, M. (1994) 'Modeling Goals and Functions of Complex Industrial Plants', in: *Applied Artificial Intelligence* 8: 259-283.

Little, A., Wood, K., and McAdams, D. (1997) 'Functional Analysis: A Fundamental Empirical Study for Reverse Engineering, Benchmarking, and Redesign', in: *proceedings of the 1997 ASME Design Theory and Methodology Conference, 1997, New York, USA*. DETC97/DTM-3879.

Ookubo, M., Koji, Y., Sasajima, M., Kitamura, Y., and Mizoguchi, R. (2007) 'Towards Interoperability Between Functional Taxonomies Using an Ontology-Based Mapping', in: *Proceedings of the International Conference on Engineering Design, ICED '07, August 28-31, 2007, Paris, France*. 154.1-154.12.

Otto, K. N. and Wood, K.L. (1996) 'A Reverse Engineering and Redesign Methodology for Product Evolution', in: *The ASME 1996 International Design Engineering Technical Conferences & Computers and Methodology Conference, August 18-22, 1996, Irvine, California, USA*. 96-DETC/DTM-1523: 1-15.

Otto, K.N. and Wood, K L. (1998) 'Product Evolution: A Reverse Engineering and Redesign Methodology', in: *Research in Engineering Design* 10 (4): 226-243.

Otto, K.N. and Wood, K.L. (2001) *Product Design: Techniques in Reverse Engineering and New Product Development*, Prentice Hall, Upper Saddle River NJ.

Price, C.J. (1998) 'Function-Directed Electrical Design Analysis', in: *Artificial Intelligence in Engineering*, 12 (4): 445-456.

Rosenman, M.A. and Gero, J.S. (1999) 'Purpose and Function in a Collaborative CAD Environment', in: *Reliability Engineering and System Safety* 64: 167-179.

Stone, R. B. and Wood K.L. (2000) 'Development of a Functional Basis for Design', in: *Journal of Mechanical Design* 122: 359-370.

Stone, R.B., Wood ,K.L., and Crawford, R.H. (2000) 'Using Quantitative Functional Models to Develop Product Architectures', in: *Design Studies* 21: 239-260.

Szykman, S., Fenves, S.J., Keirouz, W., and Shooter, S.B. (2001) 'A Foundation for Interoperability in Next-Generation Product Development Systems', in: *Computer-Aided Design* 33: 545-559.

Van Eck, D. (2009) 'On the Conversion of Functional Models: Bridging Differences Between Functional Taxonomies in the Modeling of User Actions' *Delft University of Technology Manuscript*.

Van Eck, D., McAdams, D.A., and Vermaas P.E. (2007) 'Functional Decomposition in Engineering: A Survey', in: *Proceedings of the International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE '07, September 4-7, 2007, Las Vegas, Nevada, USA*. DETC2007-34232: 1-10.

Van Wie, M., Bryant, C.R., Bohm, M.R., McAdams, D.A., and Stone, R.B. (2005) 'A Model of Function-Based Representations', in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 19: 89-111.

Vermaas, P.E. (2008) 'On Engineering Meanings and Representations of Technical Functions', in: *Proceedings of the ASME 2008 Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE '08, August 3-6, 2008, Brooklyn, New York, USA*. DETC2008-49342: 255-263.

Zhang, W.Y., Tor, S.B., and Britton, G.A. (2002) 'Automated Functional Design of Engineering Systems', in: *Journal of Intelligent Manufacturing* 13: 119-133.

Zhang, W.Y., Tor, S.B., and Britton, G.A. (2005b) 'A Graph and Matrix Representation Scheme for Functional Design of Mechanical Products', in: *International Journal for Advanced Manufacturing Technology* 25: 221-232.

Zhang, W.Y., Tor, S.B., Britton, G.A., and Deng, Y.M. (2001) 'EFDEX: A Knowledge-Based Expert System for Functional Design of Engineering Systems', in: *Engineering with Computers*17: 339-353.

Zhang, W.Y., Lin, L.F., Tong, R.F., Li X., and Dong, J.X. (2005a) 'XB-FES/o: An Ontology-Based Scheme for Functional Modelling in Design for Semantic Web Applications', in: *The 9th International Conference on Computer Supported Cooperative Work in Design, Vol. 2*: 1146-1151.

3 Functional Decomposition in Engineering: a Survey

This chapter appeared as an article, jointly written with Daniel McAdams and Pieter Vermaas, in the proceedings of the ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference:

Van Eck, D., McAdams, D.A., and Vermaas, P.E. (2007) 'Functional Decomposition in Engineering: A Survey', in: *Proceedings of the ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE), September 4-7, 2007, Las Vegas, Nevada, USA*. DETC2007-34232: 1-10.

Abstract

Functional reasoning is regarded as an important asset to the engineering designers' conceptual toolkit. Yet despite the value of functional reasoning for engineering design, a consensus view is lacking and several distinct proposals have been formulated. In this paper some of the main models for functional reasoning that are currently in use or discussed in engineering are surveyed and some of their differences clarified. The models included are the Functional Basis approach by Stone and Wood (2000), the Function Behavior State approach by Umeda et al. (1996, 1997, 2005), and the Functional Reasoning approach of Chakrabarti and Bligh (1994, 2001). This paper explicates differences between these approaches relating to: (1) representations of function and how they are influenced by design aims and form solutions, and (2) functional decomposition strategies, taken as the reasoning from overall artifact functions to sub-functions, and how these decomposition strategies are influenced by the use of existing engineering design knowledge bases.

3.1. Introduction

Functional reasoning is increasingly regarded as an important technique in engineering (see Pahl and Wallace: 2002; Hirtz et al.: 2002). For instance, in engineering designing of innovative products functional reasoning is required in the initial conceptual phase of the design process to fix the functional structure of a product relatively independently of specific design solutions, and in engineering descriptions and redesigning of existing products, functional reasoning complements physical descriptions of the products indicating the roles and uses of products and their components. Moreover, functional reasoning is increasingly regarded as a common technique, which is shared in engineering communication about functional structures of products that are (collaboratively) designed or redesigned, and in the archiving and use of functional structures of existing products in knowledge bases.

This importance of functional reasoning in engineering warrants an ongoing effort at formalization to arrive at uniform and commonly shared methods for the modeling of functions, to fix the relations that hold between functions, and to define algorithms that support reasoning about functions. Such formalization would not only facilitate the above-mentioned uses of functional reasoning, but also allow for the next step of automating functional reasoning in computer tools, ranging from CAD-CAM systems to engineering knowledge bases.

Yet, despite the broadly accepted value of a common and uniform language for functional reasoning, the current situation in the engineering sciences is rather one of plurality. Just as that there is no consensus in engineering about the meaning of the term function itself (see Chittaro and Kumar: 1998; Deng: 2002; Far and Elamy: 2005), there is currently a series of functional reasoning models being proposed and developed that are distinct rather than convergent.
This plurality has – apparently – not hampered engineering and designing so far. Yet, with the increasing use of computer tools such as CAD-CAM systems and engineering knowledge bases, and with the increasing effort to couple these systems by communication or integration, it may be envisaged that this lack of consensus may play up at some point in the near future. If, for instance, the current models for functional reasoning are developed successfully in competition with one another, and then incorporated in separate engineering computer systems, one is bound to end up with design tools and knowledge bases that are difficult to integrate. The source of this problem will then be identified as consisting of the different models of functional reasoning underlying these computer systems, which will then become topic of analysis. Anticipating these

problems, and – possibly somewhat ideally – trying to prevent them, we here set out in this paper to give this analysis immediately by surveying a number of the main models of functional reasoning, as currently discussed in the engineering literature.

We focus in this survey on conceptual differences between three models of functional reasoning, specifically on how the term function itself is understood in these models, on how functional decomposition taken as the reasoning from overall functions to sub-functions is captured, and on how this decomposition of functions is related to engineering knowledge about design solutions to functions. These models are the *Functional Basis* approach by Robert Stone and Kristin Wood (2000), the *Function Behavior State* approach by Yasushi Umeda et al. (1996, 1997, 2005), and the *Functional Reasoning* approach by Amaresh Chakrabarti and Thomas Bligh (1994, 2001). These three models were chosen because they describe functional reasoning explicitly but reveal already on first inspection relevant differences. First, in the Function Behavior State approach reasoning from functions to physical structure takes place via the intermediate notion of behavior, whereas in the other two approaches this reasoning is done without an intermediate concept. Second, the Functional Basis approach advances functional reasoning that is independent of existing design solutions to functions, whereas in the other two approaches functional reasoning explicitly depends on such solutions.

The sections 2, 3, and 4 are used for this presentation. Then, in section 5, we briefly compare the key elements of these models and in section 6 we end with conclusions.

3.2. The Functional Basis approach

The Functional Basis approach by Robert Stone and Kristin Wood is an approach to functional modeling that aims at creating a common and consistent functional design language, dubbed a functional basis, which allows designers to describe overall product functions in terms of interconnected sub-functions (see Stone and Wood: 2000).

The Functional Basis approach is focused on especially the electromechanical and mechanical domain and is presented as supporting the archiving, comparison, and communication of functional descriptions of existing products, as well as the engineering designing of new products. Archiving, comparison, and communication are assisted since the sub-functions into which overall product functions are decomposed, are described in a common universal language. Designing of new products is supported since functional modeling allows designers to make critical design decisions about the product's architecture in the early conceptual stage of designing at which only functional descriptions are considered.

Functions and sub-functions in the Functional Basis approach are captured in terms of operations on flows of materials, energies, and signals. This operation on flows description of functions is reminiscent of the work on functions by Pahl and Beitz (1996).

3.2.1. Functions as operations on flows

In the Functional Basis approach an overall product function refers to a general input/output relationship of a products' overall task, which is described in a verb-object form and represented by a black-boxed operation on flows of materials, energies, and signals. A sub-function is also described in a verb-object form but represented by a well-defined basic operation on well-defined basic flows of materials, energies, and signals. The black-boxed operations on general flows representing product functions are derived from customer needs, and the basic operations and basic flows representing sub-functions are laid down in common and limited libraries that span the functional design space. These libraries are called a *functional basis* (the current libraries have been fixed in (Hirtz et al.: 2002) by integrating the libraries proposed in (Stone and Wood: 2000) with similar libraries developed at the US National Institute of Standards and Technology (see Szykman et al.: 1999).

To give an idea on how overall product functions and connected sub-functions are represented in the Functional Basis approach, consider the example of a power screwdriver (adapted from Stone and Wood: 2000, 363); the overall product function of the screwdriver is represented (in verb-object form) as the operation "loosen/tighten screws", and has input and output flows of energies, materials, and signals. This overall product function representation thus black boxes the internal flows of the power screwdriver that transform the input flows into output flows. Focusing on the energy flow, the input side consists of flows of electricity, human force, relative rotation, and weight, and the output flows consist of torque, heat, noise, human force, and weight. Extracting the electricity part of the (internal) energy flow, the representation of the detailed (temporally ordered) sub-functions operating on the incoming electricity flow are represented as “store electricity”, “supply electricity”, “transmit electricity”, “actuate electricity”, and “regulate electricity”, respectively (see Stone and Wood: 2000, 364). These sub-functions are thus represented as basic operations on basic flows and transform an input flow step-by-step into an output flow.

Primary basic operations	Branch, Channel, Connect, Control Magnitude, Convert, Provision, Signal, Support
Primary basic flows	Material, Signal, Energy

Table 1: Primary basic operations and flows in the Functional Basis (see Hirtz et al.: 2002)

3.2.2. Developing sub-function chains

The Functional Basis approach of Stone and Wood allows, as said, to model overall product functions as sets of connected sub-functions. Their functional modeling framework provides the means to decompose overall product functions into what they call ‘small, easily solvable sub-functions’, i.e., sub-functions for which solutions exist such that ‘the [structural] form of the device [product to be designed] follows from the assembly of all sub-function solutions’ (see Stone and Wood: 2000, 360).

These sub-function assemblies, i.e., function structures, result from the following decomposition steps or tasks:

The first task is to arrive at an overall product function of a product to be designed, described in a verb-object form and represented by a black-boxed operation on flows of materials, energies, and signals. This black-boxed operation originates from customer needs and may initially be quite general, and is later on in the design process refined. The verb-object description of the product function and this (black-boxed) operation-on-flows representation are related in the sense that the verb corresponds with the operation and the object corresponds with (parts of) the flows.

The second task is to define for each input flow a chain of sub-functions that transform that flow step-by-step into an output flow. These sub-functions are also described in verb-object forms and represented by operations on flows. But now the verbs are to be chosen from the functional basis. The sub-functions part of the different chains must be ordered in time with respect to one another.

The third task is then to integrate these temporally ordered chains of sub-functions by connecting them, thereby arriving at the functional model. In this three-step process the designer thus has to analyze the input flows of the overall function in terms of basic flows from the library of basic flows, and come up with a series of operations from the library of basic operations that sequentially and/or in parallel transform the input flows step-by-step into the output flows. This transition from input to output is expressed by Stone and Wood thus: "think of each operation on the flow from entrance until exit of the product (or transformation to another flow) and express it as a sub-function in verb-object form" (see Stone and Wood: 2000, 364).

The Stone-Wood approach allows functional models to be developed with a high level of detail. It is argued that the functional basis defines a comprehensive set of basic functions, able to fully span the mechanical (and electromechanical) design space, and that these basic functions have a level of detail, which allows them to be "easily solvable" (see Stone and Wood: 2000, 360).

3.2.3. Design Repositories and the Concept Generator

Having discussed the function definitions and functional decomposition strategy of the Functional Basis approach, we discuss in this section a design tool called the Concept Generator that can be taken as a natural extension of the Functional Basis approach (see Bryant et al.: 2006). The Concept Generator is an automated, mathematically based design tool that aids in the development of functional models by retrieving design solutions to sub-functions from a repository archiving functional models of products. It thus is a tool that aids in and implements a main objective of functional decomposition, i.e., the generation of functional design solutions. The other functional decomposition approaches surveyed in this paper also employ such archived design knowledge bases in generating functional models, and we therefore include the Concept Generator in our survey in order to be able to discuss in section 5 the different ways in which knowledge bases assist in developing functional models.

Function	General input/output relationship of a product having the purpose of performing an overall task, represented by a black box operation on flows of material, energy, and signal, determined by costumer needs
Sub-function	Part of a product's overall task, represented by a basic operation on basic flows of material, energy, or signal, as defined by the functional basis libraries of basic operations and basic flows
Decomposition strategy	Analysis of the black-box operation on flows corresponding to the overall product function in terms of connected basic operations on basic flows corresponding to the sub-functions
Design knowledge base	Archived functional models of products and archived components that counts as design solutions to sub-functions

Table 2: Function definitions, decomposition strategy, and design knowledge in the Functional Basis approach

The Concept Generator can be found at a site created at the Design Engineering Lab of the University of Missouri-Rolla, which hosts also a web-based repository (see http://function.basiceng.umr.edu/delabsite/repository.html) that contains functional models of up to 102 products and their components, and that stores the design solutions – the physical structures that can perform these sub-functions – for the sub-functions part of these models. The Concept Generator is now aimed at creating new functional models for any overall product function that is fed into it, and at generating design solutions for these overall product functions on the basis of the design solutions of their sub-functions that are already stored in the repository.

In the first step of the algorithm, the Concept Generator translates the functional model of the overall product function (only models consisting of single chains of sub-functions are considered in (Bryant et al.: 2006) into information about the sub-functions and their adjacency. Secondly, the Concept Generator collects for each individual sub-function in the chain design solutions consisting of components that are stored in the repository as having that specific sub-function. In the third step, all design solutions for the product as a whole are generated by describing, on the basis of the information gathered in the first two steps, all theoretically possible component chains that solve the overall product function. Fourthly, additional information is collected from the repository on which sets of components have been actually combined in existing products, and in that sense can be taken as sets of compatible components. Finally, in a fifth step, this information about the compatibility of components is used to prune the set of theoretically possible component chains to a set of feasible component chains that solve the overall product function. In this final step a ranking is also added to these feasible chains, to "bubble the most promising solutions to the top" (see Bryant et al.: 2006, 4). The algorithm puts constraints on the functional modeling of overall product functions: the algorithm selects only those functional models that consists of sub-functions for which there are components available in the design repository that have these sub-functions, and the algorithm selects only those models that combine sub-functions that are actually combined in functional models of products in the repository.

To be sure, there is more to be said on the algorithm of the Concept Generator and on the way it is implemented in computer systems. For the purposes of this paper, i.e., comparing the conceptual differences between models of functional reasoning, the above description suffices (but see Bryant et al.: 2006; http://function.basiceng.umr.edu/delabsite/repository.html) for more details).

3.3. The Function-behavior-State approach

Like the Functional Basis approach, the Function Behavior State (FBS) approach developed by Yasushi Umeda et al. (1996, 1997, 2005) aims to support and facilitate functional design. In particular, the FBS approach is reported to support the synthetic phase of functional design, described by its developers as the process by which functional descriptions representing user intentions are transformed into structural descriptions of products-to-be (see Umeda et al.: 1996). To support such synthetic design tasks, the FBS approach employs a computer-based design tool, called the FBS modeler, which generates functional models that represent mappings amongst functions, behaviors, and product structures. Structures are referred to as states in the FBS approach and the state of a design object is represented by entities, their attributes, and relations between entities (see Umeda et al.: 1996). The FBS modeler thus is a design tool supporting "the embodiment process of required functions" (see Umeda et al.: 2005, 169), in which the concept of behavior is used as an intermediate step in transforming functions into states or structures.

3.3.1. Functions as intended behaviors

In the FBS approach functions are equated with intended behaviors and described thus: "a description of behavior abstracted by human through recognition of the behavior in order to utilize it" (see Umeda et al.: 1990, 183) and elsewhere as "a function is an association between the designer's intention and a behavior that realizes the function" (see Umeda and Tomiyama: 1997, 45). A function is represented by an association of two concepts; a verb-object pair that represents designer intentions, expressed in the form of "to do something" (see Umeda et al.: 1996, 276; Umeda et al.: 2005), and behavior(s) that can instantiate the function. Behaviors in the FBS approach are defined as "sequential state transitions along time" (see Umeda et al.: 1996, 276).

Functions are archived in so-called function prototypes. Function prototypes are designer knowledge bases that represent and store typical patterns of functions that often appear in a certain design domain (see Umeda et al.: 2005). Instead of defining functional primitives, in the FBS approach, functions are defined by developing and archiving various function prototypes from existing product designs, basing representations of functions on them. Function prototypes represent, amongst others, functions and networks of sub-functions into which the function can be decomposed, which the FBS modeler employs in developing functional models of design objects (see section 3.3).

An example (adapted from (Umeda et al.: 1996; Umeda et al.: 2005)) of function, behavior, and state representations in the FBS modeler is, for instance, the function "to charge drum" in the design of a photocopier. In the FBS modeler, this function is decomposed into the sub-functions "to rotate drum" and "to discharge voltage to drum", of which the latter sub-function is achieved by the behaviors "electrical charging" and "electrical discharging", occurring on the entities (states) "drum" and "discharger", respectively (see Umeda et al.: 1996, 280)

3.3.2. Developing Function-Behavior-State mappings

Functional design in the FBS approach, employing the FBS modeler, consists of six steps that involve two distinct types and phases of functional decomposition in which two distinct knowledge bases are used (see Umeda et al.: 1996; Umeda and Tomiyama, 1997): First, the designer specifies a required function by selecting an appropriate function prototype (see Umeda et al.: 1996). This specification and selection is somewhat unclear in the FBS approach, but the authors seem to suggest that choosing a function prototype fixes what the required function will be, instead of an appropriate function prototype being selected based on the (prior) formulation of a required function (see section 3.1). Second, the required function is then, by applying decomposition knowledge stored in the selected function prototype, decomposed into sub-functions, a procedure coined "task decomposition" (see Umeda et al.: 1996, 277). Third, these sub-functions are then mapped onto design solutions by instantiating appropriate *physical features* selected by the designer. Physical features, also archived in the selected function prototypes, refer to sets of components, their relations, and physical phenomena that together instantiate specific behavior(s) and thus particular function(s) (see Umeda et al.: 1996).

Fourth, after instantiation of physical features it may be the case that some of them cannot occur because not all the requisite physical conditions to instantiate the physical phenomena are met. For instance, for a particular physical phenomenon to occur, additional components might be needed, and connected to the ones already selected via physical feature instantiation. Resultantly, not all task-decomposed functions can be mapped onto design solutions. If this occurs, a subsystem of the FBS modeler, coined Qualitative Process Abduction System (QPAS), reasons out additional physical features to realize these conditions, thereby realizing the embodiment of the sub-functions that remained un-instantiated after the task decomposition. This procedure is dubbed "causal decomposition" (Umeda et al.: 1996, 277). QPAS is a knowledge-based reasoning system that generates design solution candidates by using physical phenomenon and physical feature knowledge bases (see Yoshioka et al.: 2004). It takes as input a particular (series of) state transition(s), including relevant entities and physical conditions, and then derives candidate physical phenomena that can achieve the state transition(s). Subsequently, QPAS derives physical features that include the physical phenomenon, after which the designer can select an appropriate one, thus instantiating the physical phenomenon (see Yoshioka et al.: 2004; for additional details on QPAS see also Ishii and Tomiyama: 1996).

Then, after physical feature selection and instantiation, the FBS modeler again employs the function prototypes knowledge base and reasons out and selects a function prototype that has the added physical feature in its function-behavior relation and thus explains the function(s) of the added physical feature. The function(s) is (are) then incorporated in the FBS model, and connected as a cause to the initially un-instantiated sub-function(s) of the task decomposition.

In the next fifth step, the designer constructs a behavioral network by connecting the instantiated physical features, completing the functional hierarchy. Then, in the sixth step, a behavior simulation is run that evaluates the FBS model. If the simulation is unsatisfactory the designer can make adjustments in either the functional hierarchy or in the behavioral network.

This six-step functional decomposition strategy thus involves two distinct decomposition phases and types; a task decomposition in the first phase (step 2) based on knowledge stored in function prototypes, and a causal decomposition in the second phase (step 4) based on knowledge derived from QPAS (see Umeda et al.: 1996). The difference between the decomposition types seems to be one of detail; in task decomposition, the (sub) functions reflect designer intentions ("to do something") and their instantiating behaviors are not causally dependent on one another and cannot be derived from physical knowledge (see Umeda et al.: 1996). By contrast, (sub) functions in causal decompositions are physically derivable and their instantiating behaviors are causally related (see Umeda et al.: 1996). Umeda et al. (1996) explain this difference thus:

> the task-decomposed functional hierarchy can be considered as a description of the designer's intention, the extracted functions by the causal decomposition can be considered as additional functions for explaining the mechanism to realize the designer's intention (Umeda et al.: 1996 , 286)

Their reference to mechanisms realizing a functional hierarchy suggests that functions in casual decomposition are more specific than functions in task decomposition.

3.3.3. Function prototypes

The abovementioned function prototypes refer to designer knowledge bases that archive and represent typical patterns of functions for a certain design domain (see Umeda et al.: 2005). The FBS strategy for defining functions, instead of an a priori method, consists in collecting various function prototypes from existing design results, and then to base representations of function on these prototypes (see Umeda et al.: 1996). The schemes within function prototypes represent task functions that include a symbol for the designer's intention, a network of subfunctions into which the task function can be decomposed, as well as function-behavior relations that are represented in the form of physical features, thus specifying candidate embodiments of the (sub) functions. Function prototypes thus contain archived function-behavior-state relations or mappings, derived from previous design results. Physical features are viewed as "building blocks" of functions (see Umeda et al.: 1996, 279), and consist of entities, relations between entities, and physical phenomena occurring on them. In the FBS modeler these elements of physical features are dubbed behavioral nodes, which together comprise a behavioral network (see Umeda et al.: 1996). As discussed in the last section, the FBS approach relies on these function prototypes in specifying required functions and in the first phase (steps 1-3) of its functional decomposition scheme where function-behavior-state-mappings are developed. Next to these function prototypes, in the second phase of the FBS decomposition strategy (steps 4-6), the FBS modeler employs QPAS that uses physical phenomenon and physical feature knowledge bases to realize un-instantiated functions from the first phase by finding out "appropriate mechanisms for realizing task-decomposed functions with physical knowledge" (see Umeda et al.: 1996, 278).

Function	Intended behavior, expressed as a combination of designer intentions and behavior(s) that can exhibit the function
Sub-function	No separate definition given: sub-functions are part of an overall function, derived from either causal decomposition or task decomposition
Decomposition strategy	Function-state mapping via behaviors and physical features. Two distinct decomposition phases and types: first, task decomposition and resulting function-state mapping; second, causal decomposition and resulting function-state mapping of task-decomposed functions that could not be mapped onto states in the first phase
Design knowledge base	Archived function-behavior-state mappings

Table 3: Function definitions, decomposition strategy, and design knowledge in the FBS approach

3.4. The Chakrabarti-Bligh approach

The Functional Reasoning approach developed by Amaresh Chakrabarti and Thomas Bligh (see Chakrabarti and Bligh: 1994, Chakrabarti and Bligh: 2001) is yet another framework for functional reasoning in the conceptual mechanical design space, in particular the domain of mechanical transmission design (Chakrabarti and Bligh: 1994). It aims to facilitate computational approaches to design and takes a prescriptive stance towards designing. Chakrabarti and Bligh (2001) identify three requirements of an ideal functional reasoning approach: it should support design of any nature, from routine to innovative, it should support the synthesis of design solutions at a given level of design, and it should support the elaboration of solution concepts through levels of detail (see Chakrabarti and Bligh: 2001). They argue (2001) that these aims can only be met when function based reasoning approaches employ a known set of design solutions (structures) in its reasoning schemes. These solutions or structures are to be represented in terms of the functions they provide or require (to operate properly). In the Chakrabarti-Bligh approach, the aim of functional reasoning is taken to provide "physical descriptions of designs, sufficient for their implementation, which would provide the intended functions of the problem" (see Chakrabarti and Bligh: 2001, 493). This approach consists of a piecemeal approach in which function-component mappings are integrated from the outset (and step by step) in a functional reasoning process until the solution space (of known structures) has been searched completely. It thus tackles the design problem, i.e., the realization of the overall product function, in parts.

3.4.1. Function as action or effect

In the Chakrabarti-Bligh approach, function is understood as a description of the action or effect required by a design problem or supplied by a solution (see Chakrabarti and Bligh: 2001). In earlier work of Chakrabarti and Bligh, function is exclusively tied to an artifact's structure or solution and expressed as a "description of the action or effect (intended to be) produced by an object, i.e., what it (is intended to do or) does" (see Chakrabarti and Bligh: 1994, 128). So, functional representations in this approach describe what a structure does, or is supposed to do (see Chakrabarti and Bligh: 2001), i.e., objects are described in terms of their known functions (see Chakrabarti and Bligh: 1994).

Functions are, moreover, defined in terms of mathematical expressions (see Chakrabarti and Bligh: 1994, Chakrabarti and Bligh: 2001) that express transformations between a set of input characteristics and a set of output characteristics. For instance, they present the example of a bicycle drive where a particular structure of this device has the function "to transform input torque into output torque" and a connected structure has the function "to transform input torque into output force" (see Chakrabarti and Bligh: 1994, 134). These functions of physical entities are, moreover, classified into three distinct types: structures that have as their primary role the transmitting or transforming of force and energy (like in the example above); structures that have couple or connect functions; and structures that take unwanted forces away (see Chakrabarti and Bligh: 1994). These function types are defined for the domain of mechanical transmission design for which they take input-output motion transformations as functional primitives. Such a classification of functions into distinct types has also been proposed by other design methodologies such as Pahl and Beitz's (1996) and Hubka and Eder's (2001). Pahl and Beitz, for instance, distinguish main functions from auxiliary functions.

3.4.2. Recursive problem redefinition

Chakrabarti and Bligh's model for functional reasoning consists of several steps and presupposes the existence of a given set of design solutions, and a design problem defined in terms of a function or a set of functions. Moreover it is required that these design solutions can solve the design problem. The steps included in their Functional Reasoning approach to arrive at a design solution for an overall design problem are (see Chakrabarti and Bligh: 2001): First, the overall problem is described in terms of the intended function(s) of the physical design to be developed. A part of this problem, the sub-function, is then chosen for synthesis, i.e., for function-structure mapping. Then, second, from a known set of structures, alternative solutions are chosen that can instantiate the sub-function. Third, the first of these alternative solutions is chosen and its (sub) function is evaluated with respect to the overall function or design problem. By incorporating the chosen part-solution, the overall function is revised and narrowed down; the number of sub-functions for which solutions have to be found is decreased. Fourthly, another sub-function of the now revised overall function is selected for synthesis, i.e., structure-function mapping. This step-by-step recursive procedure continues until all the sub-functions of the overall function are solved, i.e., all realizing structures are found. These steps thus solve a problem part (sub-function) of the overall design problem (overall function) at a time, thereby reducing the overall problem space, until the overall function is completely solved and implementing structures are identified for each sub-function. This four-step procedure leads to the generation of a single solution structure, being an aggregate of the partial solutions chosen. (The Chakrabarti-Bligh approach is prescriptive in nature; the steps above are abstractly defined and the authors do not present rules of thumb on how to select partial solutions for an overall design problem).

These steps lead to the development of a single solution structure. Then, to find alternative solution structures, the functional reasoning strategy continues by going back one step and choosing another partial solution resulting in another revision of the overall function. Then one repeats this step by going back yet another step and choosing again another partial solution which solves a part of the revised overall function, thereby revising it again. These steps continue until the complete solution space has been searched, i.e., all possible partial solutions for sub-functions have been identified and thus all possible configurations of solution aggregates (component configurations) have been identified that can solve the overall design problem or function (see Chakrabarti and Bligh: 2001).

To summarize, the functional reasoning scheme of Chakrabarti and Bligh amounts to a piecemeal, sub-function to component mapping approach incorporating these function-component mappings at each successive stage of the reasoning process (and then back again) until the solution space (of known structures) has been searched completely. It thus thereby tackles the problem, i.e., the realization of the overall device function, in parts.

3.4.3. Known physical solutions

The inclusion of known solutions in the Functional Reasoning approach of Chakrabarti and Bligh is derived from the idea that function structures cannot be developed usefully in a solution-neutral way (see Chakrabarti and Bligh: 2001); without known solutions for function realization there is no guarantee that one moves from problem state towards solution space. The lynchpin of this argument is what they refer to as the “problem of partitioning” (see Chakrabarti and Bligh: 2001, 501): the same component may support multiple sub-functions and thus there is not always a one-to-one mapping between components and sub-functions. A set of sub-functions *together* can have one solution and this many-one mapping would remain undetectable in a solution-neutral reasoning scheme, the argument goes. Moreover, a given function vocabulary is likely to be incomplete and therefore, when it would be developed in a solution-neutral way, the possibility exists that it cannot express the functions of all structures. The other way around, by ignoring solution concepts from the outset, the possibility exists that function structures are developed which cannot be solved in terms of known solutions or structures. To circumvent these possible problems, Chakrabarti and Bligh (2001) incorporate function-structure relations, based on known mappings, from the outset; structure-function couplings are incorporated from start to finish in their recursive problem reformulation approach.

The approach of Chakrabarti-Bligh is prescriptive in nature and suggests how functional reasoning schemes ought to be developed and used. Their approach, at least as laid down in (Chakrabarti and Bligh: 1994) and (Chakrabarti and Bligh: 2001), does not employ existing design knowledge bases, but instead prescribes that such knowledge bases are to be used in functional reasoning schemes.

Function	The intended action or effect of a physical object, expressed in terms of mathematical input-output transformations
Sub-function	Function of a component of a design object, expressed in terms of mathematical input-output transformations
Decomposition strategy	Recursive function-structure mapping, incorporating partial function-structure solutions in each successive stage of the reasoning process
Design knowledge base	Known function-structure mappings

Table 4: Function definitions, decomposition strategy, and design knowledge in the Chakrabarti-Bligh approach

3.5. Comparison

Having discussed all three approaches, in this section they will be compared. We will focus in particular on how the term function itself is understood in these approaches, how functional decomposition taken as the reasoning from overall functions to sub-functions is captured, and on how this decomposition of functions is related to engineering knowledge about design solutions to functions.

3.5.1. Diverging function definitions: form-independency

Whereas the three approaches take a product function to be intended or required entities that are derived from user or customer demands (although the latter is not made explicit in the Chakrabarti-Bligh approach (see Chakrabarti and Bligh: 1994, Chakrabarti and Bligh: 2001), we assume that their scheme underwrites this), they nevertheless put forward somewhat different definitions of functions.

Functional Basis	General input/output relationship of a product having the purpose of performing an overall task, represented by a black box operation on flows of material, energy, and signal determined by customer needs
FBS	Intended behavior, expressed as a combination of designer intentions and behavior(s) that can exhibit the function
Chakrabarti-Bligh	The intended action or effect of a physical object, expressed in terms of mathematical input-output transformations

Table 5: Function in the Functional Basis, FBS, and Chakrabarti-Bligh approaches

Functional Basis	Part of a product's overall task, represented by a basic operation on basic flows of material, energy, or signal, as defined by the functional basis libraries of basic operations and basic flows
FBS	No separate definition given: sub-functions are part of an overall function, derived from either causal decomposition or task decomposition
Chakrabarti-Bligh	Function of a component of a design object, expressed in terms of mathematical input-output transformations

Table 6: Sub-function in the Functional Basis, FBS, and Chakrabarti-Bligh approaches

A main difference between these approaches concerns the manner in which they implement forms or objects in their definitions of function, as will become clear below. This difference is likely related to the different aims that the respective approaches associate with function-based designing. Whereas a functional model in the Functional Basis approach is regarded as a "form-independent blueprint" (see Stone and Wood: 2000, 359) of an artifact, the main task of functional design is in the FBS approach regarded as "the embodiment process of required functions" (see Umeda et al.: 2005, 169). In line with this, Chakrabarti and Bligh (2001) argue that known structure-function solutions are to be incorporated from the outset in functional reasoning schemes, if the solution space is to be searched exhaustively. More schematically, the three approaches take functions, respectively, as *form-independent operations on flows*, as *intended behaviors of forms or objects*, and as *intended transformations of forms or objects*.

The Functional Basis approach defines a function as "a description of an operation to be performed by a device or artifact" (see Stone and Wood: 2000, 360) and, in accordance with the aim to deliver form independent functional models, its descriptions of basic operations on basic flows do not include reference to form solutions. Functions in this approach thus refer to form-independent operations. For instance, in the power screwdriver example given in section 2.1 the sub-functions operating on the electricity flow are all defined independently from any component solutions that can implement them.

By contrast, in the FBS approach form assumptions are incorporated in its representation of function. Consider, for instance, the example provided in section 3.1 where the overall function "to charge drum" is decomposed, in the FBS modeler, into the sub-functions "to rotate drum" and "to discharge voltage to drum". This example shows that, in the FBS approach, functional descriptions of overall functions and sub-functions already include form solutions. This is in line with the FBS design aim to arrive at the embodiment level of functions.

Moreover, function representations in the FBS approach do not have the level of detail or granularity that the basic functions of the functional basis have, and are not expressed in an input-output operation-on-flow type style, but instead represented in terms of the above generic functional descriptions.

In the Functional Reasoning approach of Chakrabarti and Bligh, functional representations describe what structures do or are supposed to do, and thus refer to the functions of objects. These are furthermore expressed as transformations between a set of input characteristics and a set of output characteristics. Recall the bicycle drive example of section 4.1, in which structures of the bicycle drive are ascribed a certain function, such as "to transform input torque into output force". In one way, this resembles the Functional Basis notion of functions in the sense that function representations in the Chakrabarti-Bligh approach (see Chakrabarti and Bligh: 1994, Chakrabarti and Bligh: 2001) are, like functions in the Functional Basis approach, defined in terms of transformations/operations. On the other hand, there is a difference; whereas the Functional Basis approach defines functions in a form-independent manner, Chakrabarti and Bligh speak explicitly of the functions of physical parts or objects. This definition of functions (likely) relates to the prescriptive aims that are associated with functional designing, i.e., the development of meaningful function structures is contingent on using known structure-function couplings, i.e., object functions (see section 4.3).

The object function definition of the Chakrabarti-Bligh approach is similar to the FBS approach in the sense that in the latter function definitions also include (assumptions about) forms. There are however also some differences: as said, the FBS framework does not express functions in terms of transformations and, moreover, its functional descriptions are more broadly defined than object functions in the Chakrabarti-Bligh approach. For instance, the object function "a shaft transmits torque" (see Chakrabarti and Bligh: 1994, 128) is more specific than the functional description "to move the table with a motor fast" (Umeda et al.: 1996, 278). Whereas the former defines the function of an object (shaft), the latter description explicates that an object (motor) will be used to move another object (table), but does not stipulate a precise object function.

So, summing up, these three approaches differ somewhat in how they represent functions, to wit: in the Functional Basis approach functions are form-independently defined, in the FBS approach representations of function include object or form solutions, and in the approach of Chakrabarti and Bligh functions are represented as the functions of physical objects. In the next section we suggest that these different definitions have an effect on the proposed strategies regarding the development of function structures.

3.5.2. Decomposition strategies decomposed: using design knowledge bases in different ways

The inclusion or exclusion of form assumptions in functional definitions seems to lead to different steps taken in the reasoning from overall product functions to sub-function assemblies.

Functional Basis	Analysis of the black-box operation on flows corresponding to the overall product function in terms of connected basic operations on basic flows corresponding to the sub-functions
FBS	Function-state mapping via behaviors and physical features. Two distinct decomposition phases and types: first, task decomposition and resulting function-state mapping; second, causal decomposition and resulting function-state mapping of task-decomposed functions that could not be mapped onto states in the first phase
Chakrabarti-Bligh	Recursive function-structure mapping, incorporating partial function-structure solutions in each successive stage of the reasoning process

Table 7: Decomposition strategy in the Functional Basis, FBS, and Chakrabarti-Bligh approaches

Functional Basis	Archived functional models of products and archived components that counts as design solutions to sub-functions
FBS	Archived function-behavior-state mappings
Chakrabarti-Bligh	Known function-structure mappings. The approach is prescriptive and does not employ existing knowledge bases

Table 8: Functional knowledge bases in the Functional Basis, FBS, and Chakrabarti-Bligh approaches

The Functional Basis approach of functional modeling translates overall product functions in a three-step sequence into functional models consisting of highly specific and form-independent sub-functions. These functional models consist of temporally organized and interconnected basic operations on basic flows, excluding references to implementation details. After these functional models have been developed, the Concept Generator produces design solutions for the individual sub-functions on the basis of existing design solutions of sub-functions that are stored in a repository. Use of this repository in the Functional Basis approach does not commence in the early three-step phase of developing functional models, at least not in innovative designing (and not in (Stone and Wood: 2000)). The repository is essentially a knowledge base that stores known functional models and sub-function to component mappings from actual products, but is not a necessary add-on in the development of functional models, i.e., function structures of products-to-be. These functional models are developed, using the functional basis libraries and without employing design repositories.

In contrast, in the FBS approach the use of function prototypes, from which function definitions are derived, is an integral and necessary part of functional designing. From the start, and in the different decomposition phases, the FBS approach employs archived function relations and function-structure couplings in its functional decomposition scheme. Consider, for instance, the following example of functional decomposition knowledge extracted from a function prototype (Umeda et al.: 1996, 278): the function "to move a table fast and precisely" is decomposed into the sub-functions "to move the table with a motor fast" and "to stop the table precisely". This example shows that, in the FBS approach, functional descriptions of overall functions and sub-functions already include form solutions and that the possible functional decompositions in the FBS approach are constrained by archived solutions regarding the embodiment of overall functions and sub-functions, i.e., in the above example that a motor should be used to move the table fast. This thus shows that, in the FBS approach, function definitions and the ways in which an overall function is decomposed are constrained by the use of design knowledge bases.

The Chakrabarti-Bligh approach of functional reasoning is similar to the FBS approach (and different from the Functional Basis approach) in the sense that it also, from the outset, employs design knowledge bases, storing structure-function relations, in its functional reasoning scheme. A difference with the FBS approach is that in the Chakrabarti and Bligh approach, alternative function-structure solutions must be incorporated in the functional reasoning steps in order to search the complete design solution space. It must be said that this suggestion is prescriptive in nature and not descriptive; in contrast with the Functional Basis approach and the FBS approach, the approach of Chakrabarti and Bligh suggests how function based reasoning ought to proceed and does not employ existing design knowledge databases (Chakrabarti and Bligh: 2001).

To sum up, whether functions are form-independently defined or not seems, in the surveyed approaches at least, to be related to the manner in which design knowledge bases are employed in developing functional decomposition schemes. Whereas in the Functional Basis approach the use of knowledge bases is not primary in developing functional models, these knowledge bases are an integral part in the FBS approach and the Chakrabarti-Bligh approach. As a result, form solutions are interwoven in the decomposition strategies of the latter two approaches, whereas in the former approach they are not; function-component solutions are stored in the design repository, but are not employed in functional model development.

3.6. Conclusions and outlook

In this paper we have surveyed the Functional Basis approach of Stone and Wood (2000), the FBS approach of Umeda et al. (1996, 1997, 2005), and the Functional Reasoning approach of Chakrabarti and Bligh (1994, 2001). The focus in this survey was on conceptual differences between these approaches, specifically on how the term function itself is understood, on how functional decomposition taken as the reasoning from overall functions to sub-functions is captured, and on how this decomposition of functions is related to engineering knowledge about design solutions to functions. It is argued that the Functional Basis approach defines functions in a form-independent fashion, that in the FBS approach representations of function include object or form solutions, and that in the approach of Chakrabarti and Bligh functions are represented as the functions of physical objects. It has further been argued that these function definitions are related to the manner in which the respective approaches employ design knowledge bases in developing functional decomposition schemes. Whereas in the Functional Basis approach the use of knowledge bases is not primary in developing functional models, these knowledge bases are an integral part in the FBS approach and the Chakrabarti-Bligh approach. As a result, form solutions are interwoven in the decomposition strategies of the latter two approaches, whereas in the former approach they are not.

The starting point for this survey is the observation that there is no consensus in engineering about the meaning of the term function itself, and that despite the broadly accepted value of a common and uniform language for functional reasoning, there is currently a series of functional reasoning models being proposed and developed that are distinct rather than convergent. Our aim was to make this lack of consensus explicit, primarily since we estimate that the existing differences between current approaches towards functional reasoning will eventually hamper the development of engineering computer tools and knowledge bases that can deal with functional reasoning.

The two steps that naturally succeed our analysis are, first, expanding our survey to other approaches towards functional reasoning, and, second, making an attempt to overcome the identified differences, either by integrating the approaches or by embracing some at the expense of others. The first step defines our next project, and also the second is one that we cannot take here. We finish instead with remarking that both options to overcome differences may in the end be one and the same. There is some room to integrate the three analyzed approaches. Functions may be taken as intended abstracted behaviors represented by mappings of input flows to output flows, thus arriving at a general form-independent notion of function. The use of the repository in the Functional Basis approach may be shifted to the early phase of the decomposition of product functions into sub-functions, say by constraining the set of sub-functions allowed by the functional basis (see table 1) to a set of sub-functions for which there are actually design solutions available in the repository (see Vermaas: 2007), thus arriving at a functional reasoning scheme that at least takes into account current engineering knowledge about design solutions to functions. Yet, even if this scheme towards integration is viable, it will amount to a new approach to functional reasoning which is embraced at the expense of the three approaches we have analyzed.

Acknowledgments

Research by Dingmar van Eck and Pieter Vermaas is supported by the Netherlands Organization of Scientific Research (NWO).

References

Bryant, C. R., McAdams, D. A., Stone, R. B., Kurtoglu, T., and Campbell, M. I. (2006) 'A Validation Study of an Automated Concept Generator Design Tool', in: *Proceedings of the 2006 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, DETC2006-99489: 1-12.

Chakrabarti, A. and Bligh, T. P. (1994) 'An Approach to Functional Synthesis of Solutions in Mechanical Conceptual Design. Part1: Introduction and Knowledge Representation', in: *Research in Engineering Design* 6: 127-141.

Chakrabarti, A. and Bligh, T. P. (2001) 'A Scheme for Functional Reasoning in Conceptual Design', in: *Design Studies* 22: 493-517.

Chittaro, L. and Kumar, A. N. (1998) 'Reasoning about Function and its Applications to Engineering', in: *Artificial Intelligence in Engineering*12: 331-336.

Deng, Y. M. (2002) 'Function and Behavior Representation in Conceptual Mechanical Design', in: *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, 16: 343-362.

Far, B. H. and Elamy, A. H. (2005) 'Functional Reasoning Theories: Problems and Perspectives', in: *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing* 19: 75-88.

Hirtz, J., Stone, R. B., McAdams, D. A., Szykman, S., and Wood, K. L. (2002) 'A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts', in: *Research in Engineering Design* 13: 65-82.

http://function.basiceng.umr.edu/delabsite/repository.html.

Hubka, V. and Eder, W. (2001) 'Functions Revisited', in: *Proceedings of the 13th International Conference on Engineering Design, Glasgow, August 21-23, 2001*: 69-76.

Ishii, M. and Tomiyama, T. (1996) 'A Synthetic Reasoning Method based on a Physical Phenomenon Knowledge Base', in: Sharpe, J. (ed.), *AI System Support for Conceptual Design*, Springer, London: 109-123.

Pahl G. and Beitz, W. (1996) *Engineering Design: A Systematic Approach*, Springer, Berlin.

Pahl, G. and Wallace, K.(2002) 'Using the Concept of Functions to Help Synthesize Solutions', in: Chakrabarti, A (ed.), *Engineering Design Synthesis*, Springer, London: 109-119.

Stone, R. B. and Wood K. L. (2000) 'Development of a Functional Basis for Design', in: *Journal of Mechanical Design* 122: 359-370.

Szykman, S., Racz, J. W., and Sriram, R. D. (1999) 'The Representation of Function in Computer-based Design', in: *Proceedings of the 1999 ASME Design Engineering Technical Conferences*, DETC99/DTM-8742: 1-14.

Umeda, Y. and Tomiyama, T. (1997) 'Functional Reasoning in Design', in: *IEEE Expert: Intelligent Systems and Their Applications* 12: 42-48.

Umeda, Y., Takeda, H., Tomiyama, T., and Yoshikawa, H. (1990) 'Function, Behaviour, and Structure', in: Gero, J.S. (ed.), *Applications of Artificial Intelligence in Engineering, Proceedings of the fifth International Conference, Volume 1: Design*, Springer, Berlin: 177-193.

Umeda, Y., Kondoh, S., Shimomura, Y., and Tomiyama, T. (2005) 'Development of Design Methodology for Upgradable Products based on Function-Behavior-State Modeling', in: *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing* 19: 161-182.

Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y., and Tomiyama, T. (1996) 'Supporting Conceptual Design based on the Function-Behavior-State-Modeler', in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 10: 275-288.

Vermaas, P.E. (forthcoming 2007) 'The Functional Modelling Account of Stone and Wood: Some Critical Remarks', in: *Proceedings of the 15th International Conference on Engineering Design, Paris, August 28-31, 2007*.

Yoshioka, M., Umeda, Y., Takeda, H., Shimomura, Y., Nomaguchi, Y., and Tomiyama, T. (2004) 'Physical Concept Ontology for the Knowledge Intensive Engineering Framework', in: *Advanced Engineering Informatics* 18: 95-113.

4 Explaining and Relating Different Engineering Models of Functional Decomposition

This chapter appeared as an article in the proceedings of the Design Research Society (DRS) International Conference:

Van Eck, D. (2010) 'Explaining and Relating Different Engineering Models of Functional Decomposition', in: *Proceedings Design Research Society (DRS) International Conference, 07-09 July 2010, Montréal, Canada*. 122.1-122.11.

Abstract

In this paper I analyze the use of different models of functional decomposition in engineering design. I consider models that refer to sets of desired behavior functions, to sets of desired effect functions, and ones that refer to sets of purpose functions. It is argued that the choice for a particular model is affected by whether or not its construction will be based on known function-structure connections for the functions in the model or on known behavior-structure relations that implement the functions in the model. It is then argued that whether or not such knowledge is taken into account is affected by specific design objectives. Finally, I thus argue that the choice for and suitability of particular models of functional decomposition depends on the design objectives for which these models are employed. Based on this result, it is concluded that the co-existence of different functional decomposition models has engineering value, defining the remaining task to relate them. To this end, a strategy is proposed for relating different models. The above analysis is focused on three approaches that advance particular models of functional decomposition: the Functional Basis approach in which models refer to sets of desired behavior functions, the Functional Interpretation Language approach in which models refer to sets of desired effect functions, and the Dual Stage approach in which models refer to sets of purpose functions.

Keywords: Functional decomposition model, design objective, design knowledge.

4.1. Introduction

As evidenced by a recent review of Erden et al. (2008), engineering design research has produced an impressive number of functional modeling approaches. In these approaches a variety of definitions of functions, representation schemes for functions, and strategies and representation schemes for the decomposition of functions into sub functions are formulated. It is however acknowledged in the literature that this richness of different conceptualizations has its price: it can lead to cross-communication problems between engineers working with different functional frameworks (see Rosenman and Gero: 1999; Deng: 2002). In the review given by Erden et al. (2008) the authors state, for instance, that not all reviewed approaches are compatible with one another. This sets a research challenge for establishing human and automated communication across functional frameworks. Different responses are given in the engineering literature to the diversity of functional frameworks. Erden et al. (2008) suggest that incompatibilities between approaches are due to different educational backgrounds and different application domains they are aimed at. Yet, at the same time, these authors aim with their review to develop a common framework for functional modeling that rises above the domains. Achieving this aim however suggests the discarding of a number of approaches for, otherwise, it seems that due to these incompatibilities the envisioned framework cannot provide the desired common frame that transcends particular application domains, and functional modeling thus will remain domain-specific (see Vermaas: 2009). Other authors seem to acknowledge the worth of keeping different functional frameworks side-by-side, considering them useful for different applications, yet at the same time they voice preferences for particular ones (see Umeda et al.: 1996; Deng: 2002).

In this paper I evaluate the merits of adopting a single and common framework for functional modeling. The benefits of adopting such a common framework are immediately obvious: cross-communication problems will presumably be solved. However, adopting a common framework may at the same time narrow down the application scope of functional modeling. I will assess by means of a case study how a single and common modeling framework fares when employed in different application domains. Focusing this analysis on models of functional decomposition, graphical representations of organized sets of functions, I identify three particular notions of functional decomposition model and three specific engineering objectives that are advanced in the functional modeling literature. These models and objectives are derived from: the Functional Basis (FB) approach (see Stone and Wood: 2000), the Functional Interpretation Language (FIL) approach (see Price: 1998; Bell et al.: 2007), and the Dual Stage (DS) approach (see Deng et al.: 2000a; Deng et al.: 2000b; Deng: 2002). Framed in the context of this case, this paper addresses the research question whether the use of one of these three models of functional decomposition is suited for achieving each of the three objectives. This case study shows that, rather than favoring a single framework proposal (and displacing a number of models), particular models are suited for specific objectives, implying the engineering value of keeping different models of functional decomposition side-by-side. Given this result, the challenge then becomes — with an eye to the cross-communication context mentioned earlier — to relate different models of functional decomposition. This paper also briefly outlines a strategy to meet this challenge.

I start my investigation of the different notions of functional decomposition model in terms of an analysis advanced by Vermaas (2009). His analysis shows that specific meanings of the concept of technical function are used in engineering to advance specific descriptions of technical devices. Since these descriptions are all useful to engineering, he thus explains why the concept of function is used with more than one meaning in the field. He identifies three archetypical meanings of the concept of technical function: desired behavior, desired effect of behavior, and purpose. Using this analysis, I argue that FB models refer to sets of behavior functions, FIL models refer to sets of effect functions, and that DS models refer to sets of purpose functions. In the research of Vermaas (2009), the choice for advancing a specific meaning of the concept of function, apart from the connection between a specific function meaning and a specific description of a technical device, is a question left implicit. In the case of functional decomposition, it is argued here that (i) the choice for a particular model is affected by whether or not its construction will be based on known function-structure connections, as laid down in engineering knowledge bases, for the functions in the model, and that (ii) whether or not such knowledge is considered is affected by specific design objectives that engineers aim to achieve with their models of functional decomposition.

This research is conceptual and example-based. It focuses on the internal structure of the FB, FIL, and DS approaches, in particular the use of knowledge bases. Empirical examples of functional decomposition models as specified in these approaches are analyzed, compared, and used as demonstration. This paper is organized as follows. The account of Vermaas (2009) is introduced in section 1. Different models of functional decomposition are discussed in section 2. Design objectives and the use of design knowledge bases are analyzed in section 3. Conclusions are given in Section 4.

4.2. Simplifying full descriptions of technical devices: relating goal to behavior and/or structure in different ways

Vermaas (2009) has presented an analysis of the flexible meaning of the concept of function as it is used in engineering. This analysis is developed in terms of the notions of a full and a simplified description of a technical device. Vermaas identifies five key concepts in full descriptions of technical devices (see Figure 1): *goals* of agents that refer to states in the world that agents desire to realize by using devices; *actions* that refer to intentional behaviors that agents carry out when using devices; *functions* that refer to desired roles played by devices; *behaviors* that refer to physicochemical state changes of devices; and *structures* that refer to the physicochemical materials and fields of devices, their configurations, and their interactions.

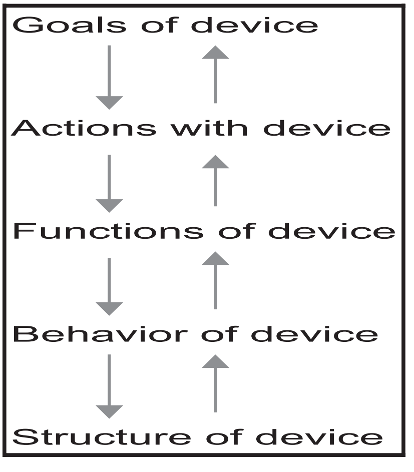

Figure 1: Full description of a technical device in terms of five key concepts (adopted from Vermaas: 2009)

Vermaas asserts that the concept of function is used with different meanings and that this flexibility affords different ways in which such full descriptions of technical devices can be simplified. Full descriptions in terms of the five key concepts are elaborate, and in particular engineering settings it makes sense to simplify them by "by-passing" one or more of the key concepts (see Vermaas: 2009, 2.119). Vermaas demonstrates this by-passing of certain key-concepts in terms of three approaches toward the modeling of or reasoning with functions, each advancing a different meaning of the concept of function: the FB approach of Stone and Wood (2000), the Multilevel Flow (MFM) approach of Lind (1994), and the Function-Behavior-Structure (FBS) approach of Gero (1990).

Vermaas (2009) argues that in the FB approach, the concepts of action and behavior are "by-passed" and that the concept of function is used in its meaning of *desired behavior* (by specifying the role a device should play in terms of its behavior) to relate goals to structure (see Figure 2). FB functions are modeled as operations-on-material, energy, or signal flows. Vermaas argues that these descriptions refer to physical behaviors since they represent conversions of matter and/or energy in which the input quantity matches the output quantity, meeting physical conservation laws. A function of an electric screwdriver, for instance, that is described as 'converting electricity into torque and heat' (see Stone and Wood: 2000, 364) in which the energy of the electricity equals the sum of the energies of heat and torque. In the FB approach, the concept of behavior is thus bypassed and the concept of function is instead used to refer to behavior(s). Vermaas asserts that in the MFM approach the key concept of action is by-passed but not the concept of behavior (see Figure 2). And he argues that in this approach the concept of function is used in its meaning of *desired effect of behavior* (by specifying the role of the device in terms of the effects of the device's behavior) to relate goals to behavior. Functions in MFM are represented in terms of operations and flows, and may be represented in terms of only input or output flows. A function example may be, say, 'producing torque'. This description also refers to (features of) behavior but does not meet conservation laws, referring only to the desired effects of behavior (which makes good sense, since the concept of behavior is used to account for the conservation of matter and energy). His analysis of Gero's FBS approach further broadens the spectrum of engineering meanings of the concept of function. His analysis of the simplified descriptions advanced in this FBS framework shows that the concept of function may also be used to refer to a *goal* desired by an agent. A function example may be, say, 'having a rotational force down a shaft'. This description refers to a state of affairs in the world, intended by an agent. Vermaas (2009) considers two ways in which simplified descriptions in this FBS framework may be interpreted, due to the shifting position of Gero on the meaning he ascribes to the concept of function: either as functions as goals to behavior, and then structure, by-passing the concept of action (see Figure 2), or, alternatively, as side-stepping both the concepts of goal and action, and reasoning from functions as desired effects to behavior, and then to structure.

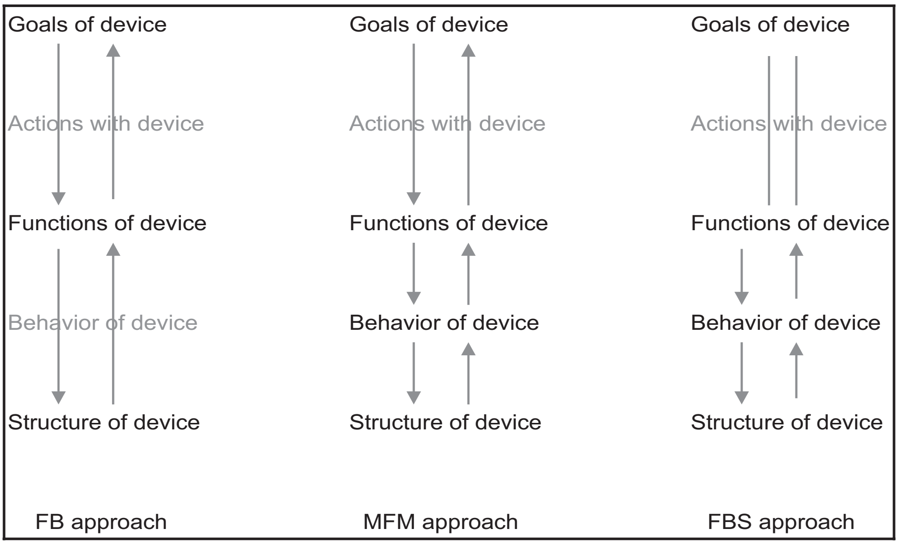

Figure 2: Simplified descriptions advanced in the FB, MFM, and FBS approaches (adopted from Vermaas: 2009)

4.3. Engineering models of functional decomposition

The foregoing analysis shows that different meanings of the concept of function are employed in relating goals to structure and/or behavior: desired behavior, desired effect, and goal. To distinguish goals of users from goals of designers, I coin the latter purposes. Purpose function descriptions hence refer to states of affairs in the world, intended by designers. These different meanings of function are given in Table 2.

Behavior function: desired behavior of a device
Effect function: desired effect of behavior of a device
Purpose function: purpose for which a device is designed

Table 1: Three meanings of the concept of function

These three types of functions are also described in models of functional decomposition, graphical representations of organized sets of functions, and the flexibility in the way goals (or designer purposes) are related to structure and/or behavior is also in play in the functional decomposition case. Often, in engineering design, models of functional decomposition (that make up an overall function) are advanced to relate goal (or purpose) to structure (see Stone and Wood: 2000; Deng et al.: 2000a, 2000b; Chakrabarti and Bligh: 2001). In this section I give an analysis of 3 approaches toward functional modeling, each advancing a different model of functional decomposition (fm_D) by which goals (or purposes) are related to structure and/or behavior. These three models are depicted (and abbreviated) in Table 2. Behavior function fm_Ds are advanced in, for instance, the FB approach, the Systematic approach (see Pahl and Beitz: 1988), and the Functional Reasoning approach (see Chakrabarti and Bligh: 2001). Effect function fm_Ds are advanced in, for instance, the FIL approach and the MFM approach. The third notion of purpose function fm_D is advanced in the DS approach. (I do not consider here the use of the concept of function to refer to a 'user action', nor the description of such functions in models of functional decomposition. See Van Eck (2010a) for these details).

Functional decomposition model of organized set of behavior functions (behavior function fm_D)
Functional decomposition model of organized set of effect functions (effect function fm_D)
Functional decomposition model of organized set of purpose functions (purpose function fm_D)

Table 2: Three models of functional decomposition

Based on this analysis, I then develop the position in section three that the choice for particular models of functional decomposition is affected by particular design objectives that engineers aim to achieve with them.

4.3.1. Functional Basis approach

The Functional Basis (FB) approach, developed by Stone and Wood (2000), is an approach to functional modeling that is aimed at supporting the engineering designing of new products in the electro-mechanical domain, as well as the archiving, and communication of functional descriptions of existing products. In the FB approach, an overall product is described in a verb-object form and represented by a black-boxed operation on flows of materials, energies, and signals. A sub function, describing a part of the product's overall task, is also described in a verb-object form but represented by a well-defined basic operation on a well-defined basic flow of materials, energies, or signals. The black-boxed operations on general flows representing product functions are derived from customer needs, and the basic operations and basic flows representing sub functions are laid down in libraries of operations and libraries of flows, together called a *functional basis*.

To support engineering designing, Stone and Wood (2000) present a three-step methodology to develop functional decomposition models. The method starts with describing a product function in a verb-object form, derived from customer needs and represented by a black-boxed operation on flows of materials, energies, and signals. A chain of operations-on-flows is then specified for each black box input flow, transforming that flow step-by-step into an output flow. These operations-on-flows are to be selected from the FB libraries. Finally, these chains of operations-on-flows are aggregated, completing the model of functional decomposition. Such models are intended to provide a "form-independent blueprint" (see Stone and Wood: 2000, 359) of the functions of a product-to-be-designed, meaning that no known technical solutions for sub functions – structures – are taken into account during its specification. Not taking such existing function-structure connections into account during specification of a model is intended to support creative, and innovative designs (see Stone and Wood: 2000). And in order to support such mappings after completion of a model, the sub functions in it should be small and easily solvable ones. The FB approach currently includes a web-based repository in which functional decompositions of existing products are archived, as well as components counting as design solutions for the sub functions that are part of these decompositions, supporting such mappings systematically.

Functional decomposition model

Relative to the behavior, effect, and purpose meaning of technical functions, FB product functions and sub functions can be taken to refer to desired behaviors (which may include their effects) since they represent conversions of matter and/or energy in which the input quantity matches the output quantity, meeting physical conservation laws (see Vermaas: 2009; Van Eck: 2009). For instance, the sub function 'converting electricity into torque and heat' (see section 1 and Figure 3). FB models thus are behavior function *fm*$_D$s, organized such that the output flows of preceding behavior functions constitute the input flows of succeeding behavior functions (Figure 3).

Relative to the five key concepts, the concepts of action and behavior are bypassed and sub functions in FB *fm*$_D$s relate a goal – customer need – to structures (components as archived in the FB repository).

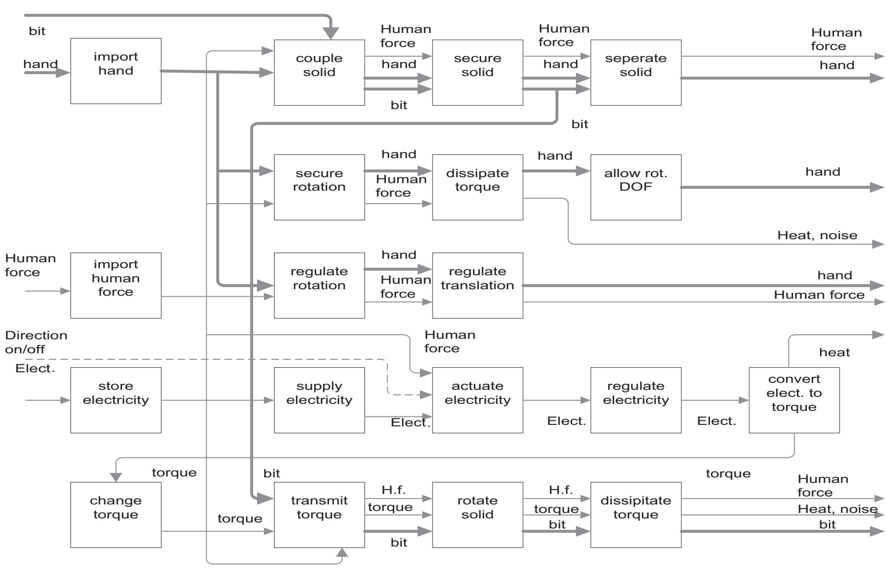

Figure 3: FB behavior function *fm*$_D$ of a power screwdriver (see Stone and Wood: 2000)

4.3.2. Functional Interpretation Language approach

The Functional Interpretation Language (FIL) approach (see Price: 1998; Bell et al.: 2007) is an approach to functional modeling that is aimed at supporting design analysis tasks, such as failure analysis and design verification, mainly in the electro-mechanical domain. In this approach functions of technical devices are taken as and represented by trigger-effect pairs. An overall function is represented in terms of three elements: the "purpose" achieved by the function, the "trigger" of the function, and the "effect" of the function (see Bell et al.: 2007, 400). Purposes in FIL refer to goals that agents aim to achieve when using devices. Triggers and effects in FIL describe the boundaries of a technical device, and are intended as labels that allow linking to relevant properties of its behaviors (in a design analysis context). Sub functions are either represented in terms of these three elements or as combinations of two out of these three elements, depending on the type of device analyzed.

In a design analysis setting, an overall function is decomposed into sub functions when its achievement depends on more than one trigger and effect, or when different trigger-effect pairs can achieve the overall function. In a model of functional decomposition that results, the triggers and effects of the sub functions then replace the trigger and effect (originally) associated with the overall function. Such models allow linking (in a design analysis context) to relevant properties of the behaviors of a technical device, both for tracing the cause of failures and for verifying whether a device's behavior implements the effects that are desired. This is done by checking the "on/off" states of triggers and effects. For instance, in a functional decomposition model of a room light-function a sub function is represented by the trigger-effect pair "switch on-light on". Now, say, if the lamp switch position is "on" (trigger) and the effect "light on" is absent, this sub function has failed (see Bell et al.: 2007, 405). This trigger-effect relation allows tracing those behavioral properties that cause this failure, say, an electrical short circuit. Such functional decomposition models describe the (sub) functions of devices of which its required behaviors and structures are known.

Functional decomposition model

Relative to the behavior, effect, and purpose meaning of technical functions, FIL overall functions and sub functions can be taken to refer to desired effects of behaviors. For instance, the sub function of the room light-function above only refers to the desired effect of the light being on, and not to the behavior due to which this effect is displayed, say, the conversion of electrical energy into light and heat. FIL models thus are effect function fm_Ds (Figure 4).

Relative to the five key concepts, the concept of action is bypassed and sub functions in FIL fm_Ds relate a goal – FIL-purpose – to behaviors.

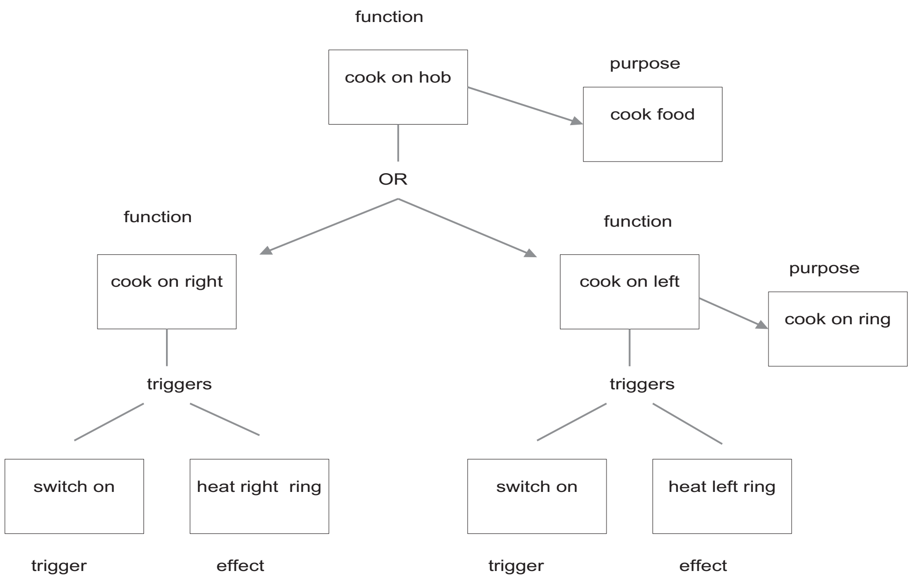

Figure 4: FIL effect function fm_D of a two ring-cooking hob (see Bell et al.: 2007)

4.3.3. Dual Stage approach

The Dual Stage (DS) approach, developed by Deng, Tor, and Britton (2000a, 2000b, 2002), is an approach to functional modeling that is aimed at supporting the engineering designing of products in the mechanical domain. In this approach two types of functions are defined: purpose functions and action functions (see Deng: 2002). A purpose function refers to a designer's intention or purpose of a design. An action function is defined as an abstraction of intended behavior. Both types of function are represented by verb-noun descriptions.

To support engineering designing, Deng, Tor, and Britton (2000a, 2000b, 2002) present a knowledge base-assisted method to develop functional decompositionmodels of an overall purpose function. First, an overall purpose function is decomposed into purpose sub functions, using a function-library in which existing functional decomposition-design knowledge is stored. This library archives descriptions of purpose functions that have "pointers" added to them, linking them to sub functions and to functions of which they are a functional element (see Deng et al.: 2000b, 352). An overall purpose function is decomposed into those sub functions to which it is linked in the library. Then, these purpose sub functions are mapped onto structures using a physical structure-library, in which descriptions of commonly used structures are archived. The purpose sub functions stored in the function-library also have pointers to the structures housed in the physical structure-library that are suitable to implement them, thus supporting function-structure mapping. These steps constitute the first stage of the DS modeling framework. When functions from the function-library do not have pointers to physical structures as housed in the physical structure-library, hence cannot be mapped onto structures, a physical phenomena-library is then employed to carry out function-structure mapping. This library stores descriptions of commonly used physical behaviors and their effects, which have pointers added to them, linking them to structures in the physical structure-library. Action functions refer to behavioral effects (see Deng: 2002). This physical phenomena-library is searched to retrieve those behavioral effects – action functions – that are deemed suitable to achieve an unmapped purpose sub function. By linking a purpose sub function to a behavioral effect, which has a pointer added to a physical structure, purpose function-structure mapping is supported. These steps constitute the second stage.

The usage of these libraries in specifying models of functional decomposition (and supporting function-structure mapping) is aimed at employing past design knowledge in a systematic way to assist engineering designing (see Deng: 2002). Models of functional decomposition are constructed that consist of sub functions for which structures are known. For instance, Deng et al. (2000a, 43) specify a purpose sub function of the overall purpose function of a rivet setting device as "to exert certain force on the rivet by a working head, during the process the working head moves down a specified distance", which contains a pointer to the structures of "working head" and "rod". This type of designing in which known function-structure relations (and function-behavior-structure relations) are employed in constructing functional decomposition models is also referred to as design-by-analogy or analogy-based-design (see Goel and Bhatta: 2004).

Functional decomposition model

Relative to the behavior, effect, and purpose meaning of technical functions, DS-purpose functions and sub functions can be taken to refer to states in the world desired by an agent-as-designer. For instance, the sub function above refers to the desired state that a rivet has force applied to it, come about by a sequence of states pertaining to the position of the working head. DS models thus are purpose function fm_D s (Figure 5).

Relative to the five key concepts, the concepts of action and behavior are bypassed in the first stage and sub functions in DS fm_Ds directly relate designer purposes to structures. (In the second stage, the concept of behavior is not bypassed and effect functions are used to relate designer purposes to behavior. In this stage the step from goal to behavior is taken via a single function, not via a model of functional decomposition).

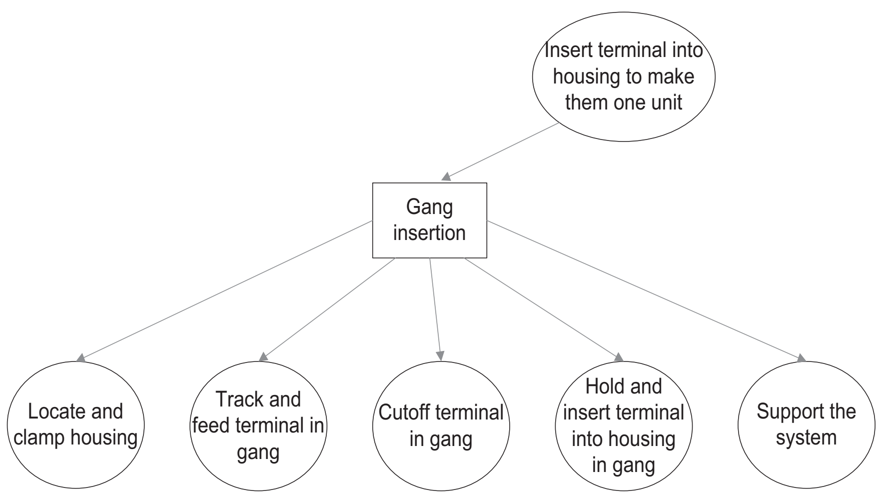

Figure 5: DS purpose function fm_D of a terminal insertion device (part of an automatic assembly system for manufacturing electronic connectors; the block "gang insertion" refers to knowledge about physical structures that implement the functions, depicted in the oval nodes) (see Deng: 2002)

4.4. Choosing functional decomposition models: design knowledge employment and design objectives

Returning to the notion to discard a number of functional modeling approaches and settle for a single and common framework for functional modeling, here an alternative position is developed. I will argue that the choice for constructing particular models of functional decomposition (behavior function fm_Ds, effect function fm_Ds, or purpose function fm_Ds) is based on their suitability for achieving particular design objectives. And that none of these three models alone is (best) suited for achieving the three considered objectives of innovative design, design analysis, and design-by-analogy. Given the suitability of particular models for particular objectives, one can both explain and defend the keeping of different models of functional decomposition side-by-side (and the approaches in which they are advanced) in engineering design.

Consider that, due to particular design objectives, particular design knowledge is or is not used in the construction of models of functional decomposition: construction of a model can be based on known function-structure connections for the functions in the model (DS), known behavior-structure relations that implement the functions in the model (FIL), or, instead, not based on such (types of) knowledge (FB). Precisely the type of design knowledge that is or is not employed, as due to design objectives, makes the models suitable for achieving the objectives for which they are advanced.

Consider FB models that are used to support the objective of innovative design: relating goals to structures by fm_Ds without employing known function-structure connections or behavior-structure relations during construction of these fm_Ds. Since behavior and structure are not taken into account in the construction phase, behavior function fm_Ds are suited for relating goals to structures since behavioral descriptions (which may include effects) are detailed enough to support the selection of structures after the model is constructed. Purpose function fm_Ds and effect function fm_Ds, instead, are too coarse-grained to allow the selection of structures in any precise way, when existing knowledge on behaviors and structures is not considered in the construction phase of such models. The use of such models, skipping reference to behaviors and effects in purpose function fm_Ds and to behaviors in effect function fm_Ds, does not give (in a precise manner) those structures that exercise certain behaviors, resulting in certain effects that are suitable to achieve the goals one wants realized. In the case of purpose function fm_Ds, the designer may choose to select structures already known to him/her to achieve the purpose functions in the model, but this changes the objective of innovative design into design-by-analogy (precisely the objective for which models of functional decomposition are employed in the DS approach). The use of effect function fm_Ds to relate goals to structures, skipping reference to behaviors, also seem to provide insufficient precision for selecting (potentially innovative) structures that exercise behaviors which result in the effects desired (although more precision is gained than using purpose function fm_Ds). For instance, a car's headlight effect-function "light on" may be sufficient to select well-known structures of an incandescent lamp or halogen one, but without a desired behavioral specification, the choice for, say, a more recent LED lamp (which differs in its behaviors by which the effect "light on" results) is not obvious (again the design objective would shift from innovation to analogy).

Now consider FIL models that are used to support the objective of design analysis: relating goals (FIL-purposes) to behaviors (of structures) by fm_Ds that are constructed based on known (and required) behavior-structure relations of an existing design. Since behavior and structure are known, effect function fm_Ds are suited for relating goals to behavior, since they allow verifying whether the behaviors exercised by structures display (in the intended fashion) the effects that are desired for contributing to the goals of the device. Using a purpose function fm_D, instead, skipping reference to effects, does not give the precision to ascertain whether or not the desired effects are indeed manifested *in the intended way* by the behaviors of the device. For instance, the purpose function "illumination in a room" seems sufficient for determining whether the behavior of the device implements the effect function "light on". Yet, only an effect function description, say "switch on-light on", is suited for verifying whether the behavior of the device implements this effect in the intended way: say, the switch might be "off" while the light is still on. The device's behavior, in this case, implements a desired effect but not in the intended fashion. This goes undetected with the purpose function description "illumination in a room" (More elaborate behavior function fm_Ds may also do the trick, but are unnecessarily complex in this setting).

Consider, finally, DS models that are used to support the objective of design-by-analogy: relating purposes to structures by fm_Ds that are constructed based on known (purpose) function-structure connections for the functions in the model. Since these connections are known, purpose function fm_Ds are suited for directly relating purposes to structures. Constructing more elaborate behavior function

fm_Ds or effect function fm_Ds is unnecessary for this objective, only adding additional complexity to the design task and decreasing efficiency (if, however, there are no structures available for the purpose functions, behavior function fm_Ds or effect function fm_Ds do become suited for relating purposes to structures. See, for instance, the use of effect functions in the DS approach for relating purpose to behavior and then structure).

In sum, different models of functional decomposition are suited for different objectives (and as the "switch on-light on" example above shows, particular representational frameworks are suitable for particular objectives as well). Therefore, I submit that the co-existence of different approaches, advancing specific fm_Ds, has engineering value and is to be preferred above a single and common framework for functional modeling. A task remaining is then to relate different fm_Ds.

This step of relating different fm_Ds is developed in more detail in (Van Eck: 2010a, 2010b). The idea behind it is that in order to relate behavior function fm_Ds to effect function fm_Ds or to purpose function fm_Ds, the information expressed in the effect function fm_Ds or purpose function fm_Ds must be expanded in order to relate them to behavior function fm_Ds. For instance, whereas an effect function fm_D only represents desired outputs such as 'producing torque', a behavior function fm_D contains more elaborate descriptions such as 'conversions of electricity (input) into torque and heat (output)'. By expanding the desired effect (or purpose) descriptions with input and (other) effect descriptions (such as 'electricity' and 'heat'), the descriptions become behavior function descriptions that meet physical laws. Such descriptions are the ones advanced in behavior function fm_Ds. By rephrasing effect function (or purpose function) descriptions as behavior function descriptions by expanding them one can thus relate different fm_Ds. Vice versa, one can move from behavior function fm_Ds to effect function fm_Ds or purpose function fm_Ds by selecting and describing only specific elements of behavior function descriptions, namely their desired effects.

4.5. Concluding remarks

In this paper I have analyzed the use of different models of functional decomposition in engineering design. I considered models that refer to sets of desired behavior functions, to sets of desired effect functions, and ones that refer to sets of purpose functions. It is shown that the choice for and suitability of particular models of functional decomposition depends on the design objectives for which these models are employed. Based on this result, it is concluded that the co-existence of different models has engineering value and is to be preferred above a single and common framework for functional modeling.

Acknowledgements

Thanks to Pieter Vermaas and two anonymous reviewers for their useful feedback. This research is supported by the Netherlands Organization of Scientific Research (NWO).

References

Bell, J., Snooke, N., and Price, C. (2007) 'A Language for Functional Interpretation of Model Based Simulation', in: *Advanced Engineering Informatics* 21: 398-409.

Chakrabarti, A. and Bligh, T. P. (2001) 'A Scheme for Functional Reasoning in Conceptual Design', in: *Design Studies* 22: 493-517.

Deng, Y.M. (2002) 'Function and Behavior Representation in Conceptual Mechanical Design', in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 16: 343-362.

Deng, Y.M., Tor, S.B., and Britton, G.A. (2000a) 'Abstracting and Exploring Functional Design Information for Conceptual Mechanical Product Design', in: *Engineering with Computers* 16: 36-52.

Deng, Y.M.,Tor, S.B., and Britton, G.A. (2000b) 'A Dual-Stage Functional Modelling Framework with Multi-Level Design Knowledge for Conceptual Mechanical Design', in: *Journal of Engineering Design* 11(4): 347-375.

Erden, M.S., Komoto, H., Van Beek, T.J., D'Amelio, V., Echavarria, E., and Tomiyama, T. (2008) 'A Review of Function Modelling: Approaches and Applications', in:*Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 22: 147-169.

Gero, J.S. (1990) 'Design Prototypes: A Knowledge Representation Schema for Design', in: *AI Magazine* 11(4): 26-36.

Goel, A.K. and Bhatta, S.R. (2004) 'Use of Design Patterns in Analogy-Based Design', in: *Advanced Engineering Informatics* 18: 185-194.

Lind, M. (1994) 'Modeling Goals and Functions of Complex Industrial Plants', in: *Applied Artificial Intelligence* 8: 259-283.

Pahl, G. and Beitz, W. (1988) *Engineering Design: A Systematic Approach*, Springer, Berlin.

Price, C.J. (1998) 'Function-Directed Electrical Design Analysis', in: *Artificial Intelligence in Engineering* 12 (4): 445-456.

Rosenman, M.A. and Gero J.S. (1999) 'Purpose and Function in a Collaborative CAD Environment', in: *Reliability Engineering and System Safety* 64: 167-179.

Stone, R.B. and Wood, K.L. (2000) 'Development of a Functional Basis for Design', in: *Journal of Mechanical Design* 122: 359-370.

Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y., and Tomiyama, T. (1996) 'Supporting Conceptual Design Based on the Function-Behavior-State-Modeler', in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 10: 275-288.

Van Eck, D. (2009) 'On Relating Functional Modelling Approaches: Abstracting Functional Models from Behavioral Models', in: *Proceedings of The 17th International Conference on Engineering Design (ICED 09), August 24-27, 2009, Stanford, California, USA*: 2.89-2.100.

Van Eck, D. (2010a) 'On the Conversion of Functional Models: Bridging Differences Between Functional Taxonomies in the Modeling of User Actions', in: *Research in Engineering Design* 21 (2): 99-111. DOI 10.1007/s00163-009-0080-7.

Van Eck, D. (2010b), 'Supporting Functional Knowledge Exchange Between Functional Taxonomies by Establishing Function-Behaviour Links', in: *Proceedings of the eight international symposium on tools and methods of competitive engineering-TMCE 2010, April 12-16, 2010, Ancona, Italy*: 1583-1595.

Vermaas, P.E. (2009), 'The Flexible Meaning of Function in Engineering', in: *Proceedings of the 17th International Conference on Engineering Design (ICED 09), August 24-27, 2009, Stanford, California, USA*: 2.113-2.124.

5 Incommensurability and Rationality in Engineering Design: The Case of Functional Decomposition

This chapter is forthcoming as an article in *Techné: Research in Philosophy and Technology*.

Abstract

In engineering design research different models of functional decomposition are advanced side-by-side. In this paper I explain and validate this co-existence of models in terms of the Kuhnian thesis of methodological incommensurability. I advance this analysis in terms of the thesis' construal of (non-algorithmic) theory choice in terms of values, expanding this notion to the engineering domain. I further argue that the (by some) implicated threat of the thesis to rational theory choice has no force in the functional decomposition case: co-existence of different models of functional decomposition is rational from an instrumental point of view. My explanation covers cases in which different models are advanced as means for the same objective. Such cases cannot be explicated with the explanatory construct of variety in objectives, as advanced in other analyses of co-existing conceptualizations in engineering.

Keywords: Functional decomposition, co-existence, methodological incommensurability, rationality

5.1. Introduction

Engineering design research on the modeling of technical functions provides what seems to be a striking contrast with research in the sciences. Whereas research in the sciences exhibits an orientation toward establishing (increasingly) unambiguous and commonly shared concepts[1] – and engages in debate about the adequacy of the conceptualizations proposed – functional modeling research does not, by and large, strive toward such conceptual uniformity nor engages in such debate. For instance, a key concept such as function lacks a uniform meaning in functional modeling research but, instead, is specific to particular modeling frameworks (see Erden et al.: 2008). Models of functional decomposition, i.e., graphical representations of organized sets of functions, likewise come in a variety of flavors. And the majority of authors in functional modeling research accept this status quo. Some authors do aim to develop a common framework for functional modeling by means of specific functional conceptualizations (see Erden et al.: 2008; see also Chandrasekaran; 2005). Yet, most authors merely stick to advancing their favored frameworks without superiority claims over other ones.

Philosophical analyses of the co-existence of different conceptualizations in engineering explain the maintaining of this status quo as having instrumental value to engineering (see Bucciarelli: 1994, 2003; Vermaas: 2009; Van Eck: 2010a). The analyses of Bucciarelli (1994, 2003) and Vermaas (2009) relate specific conceptualizations (as suitable means) to specific objectives, thus explaining (and validating) co-existence in terms of a variety of engineering ends. I have argued that the choice for and suitability of particular models of functional decomposition for particular objectives is influenced by whether or not specific design knowledge is employed in building these models, thus explaining (and validating) co-existence of different models in terms of variation in design knowledge employment (see Van Eck: 2010a). This latter analysis explains co-existence in informative fashion insofar as the knowledge used in building models does not contain or refer to a specific notion of function or specific functional decomposition model. When the knowledge used does already refer to a specific notion of function or specific model of functional decomposition, the choice for (the construction of) specific models is obvious, but explication of such choices in terms of knowledge employment would become circular.

In this paper I further expand on the above analyses. Focusing on research on the modeling of functional decompositions, I argue that *different* functional

[1] For instance, the efforts spend in psychiatry and clinical psychology to arrive at unambiguous and shared classification criteria for psychiatric disorders as laid down in the Diagnostic and Statistical Manual of Mental Disorders (DSM-IV) (see American Psychiatric Association: 2000)

decomposition models are also advanced in the engineering literature as means for the *same* objective. What to think of co-existence in this case? The questions that I will address are (i) why co-existence of different models obtains when the objective for which they are constructed is the same (rather than fixing a single best and commonly shared model) and (ii) whether the above implicated value of co-existence also holds in the case of co-existing and different models that are constructed for the same objective?

The previous analyses of co-existence that put forward variety in objectives as a central explanatory construct do not cover the above case(s) in which different functional decomposition models are used side-by-side as means toward the same objective. And neither does an explanation in terms of variation in design knowledge employment, since the models advanced in the above case are built using knowledge that already refers to specific models of functional decomposition. I will therefore follow a different tack to explain this case. I explain co-existence in this functional decomposition case in terms of the Kuhnian thesis of *methodological incommensurability*.[2] Key to this thesis is the notion that there is no neutral algorithm that governs scientific theory choice. Kuhn (1977) argued that theoretical disputes between advocates of rival frameworks cannot be solved by recourse to a neutral algorithm that dictates theory choice, since there is no commonly shared set of criteria or standards available on the basis of which such a choice can be forged. Kuhn (1977) pressed the point that such standards do not function as algorithmic rules by which one is able to determine theory choice but, rather, as *values* guiding such choices. I explain co-existence in terms of (and by expanding on) Kuhn's notion that one can explain divergence of theory choice in terms of variation in values. I argue that the choice for particular models of functional decomposition depends on the engineering values that are employed in choosing/evaluating them, and that these engineering values vary (and conflict) between modeling accounts.

---

2 I use the term "Kuhnian" since the thesis is labeled by some of Kuhn's commentators as methodological incommensurability but not so by Kuhn himself. Whereas Kuhn's earlier (1970) incommensurability thesis contains both methodological and semantic aspects, he focused in later work more exclusively on semantic notions such as translation-failure and taxonomic structure (e.g. 1991). Due to this more specific focus, some commentators began to distinguish semantic incommensurability from methodological incommensurability (e.g., see Sankey: 1999). Kuhn's most explicit treatments of methodological incommensurability can be found in (Kuhn 1977, 1983). The thesis is currently more frequently discussed under such headings as 'rationality of theory choice' and 'epistemological relativism' (see Sankey: 1999).

I conclude that the functional decomposition case exemplifies methodological incommensurability in the engineering domain.

Kuhn's analysis of values, in addition, led him to conclude that scientists' choice of (competing) theories can be considered rational. This conclusion has spawned extensive debate in philosophy of science. A key issue is whether in the absence of a commonly shared algorithm scientists' choice of theories can in fact be considered rational (see Kuhn: 1977). Thus, authors that accept variation in values are pressed to show that theory choice by means of values ensures the rationality of scientists' choice of theories (see Worrall: 1988; Sankey: 1995, 2002). I will address this issue in the functional decomposition case. I argue that the choice and usage of different models by different engineers is rational from a practical point of view. I construct this argument along the lines of a position developed by Sankey (2002) in which he combines variation of values with a means-end analysis of values.

This paper is organized as follows. In section 2 I briefly present an overview of engineering notions of function and models of functional decomposition. Section 3 introduces the earlier analyses of co-existence. In section 4 I explain co-existence in terms of the thesis of methodological incommensurability. I argue for the rational grounds of co-existence in section 5. Section 6 concludes the paper.

5.2. Engineering notions of function and functional decomposition

The concept of function has a flexible meaning in the engineering domain. An analysis of Vermaas (2009) established three archetypical ones: behavior functions that refer to the desired behaviors of a device, effect functions that refer to the desired effects of the behaviors of a device, and purpose functions that refer to the purposes for which a device is designed. And, in addition, I identified a fourth (see Van Eck: 2009/2010): action functions that refer to intentional behaviors carried out by an agent using a device.

Behavior function descriptions characterize conversions of matter and/or energy in which physical conservation laws are taken into account, thus referring to physical behaviors. An electric screwdriver's function of 'converting electricity into torque and heat' (see Stone and Wood: 2000, 364), for instance, in which the energy of the electricity is supposed to equal the sum of the energies of heat and torque. Effect function descriptions also characterize (features of) behavior but do not take conservation laws into account, referring only to the desired effects of behavior. In case of a screwdriver's function, say, 'producing torque'. Purpose function descriptions refer to states of affairs intended by designers that are the final result(s) of behaviors. In the screwdriver case, say, 'having a rotational force down a shaft'.[3] Finally, action function descriptions are used to characterize user actions with a device; again in the screwdriver case, say, 'manually inserting a screw in a screw bit'.

These four notions of function are also described in engineering models of functional decomposition, i.e., graphical representations of organized sets of functions. Engineers put such models to a variety of uses. They use them, among others, in the conceptual phase of engineering designing to specify the desired functions of some artifact-to-be (see Stone and Wood: 2000; Chakrabarti and Bligh: 2001); in the reverse engineering of existing artifacts to identify their functions (see Otto and Wood: 2001); and engineers use functional decomposition models to identify malfunctions of artifacts (see Bell et al.: 2007).

In this paper I consider three distinct engineering models of functional decomposition: a functional decomposition model of an organized set of behavior functions (behavior function fm_D), a functional decomposition model of an organized set of effect functions (effect function fm_D), and a functional decomposition model of an organized set of purpose functions (purpose function fm_D).[4] Examples of such models are given in Figures 1, 2, and 3.

---

3 The distinction between effect and purpose function is not completely clear-cut: both relate to features of behavior. Yet, purpose function descriptions, such as 'having a rotational force down a shaft', are, typically, phrased in terms of a result of behavior in the environment of a technical artifact. Effect function descriptions, such as 'producing torque', are phrased in terms of behavioral features of a technical artifact (this distinction originates from Chandrasekaran and Josephson (2000) who distinguish between device-centric and environment-centric descriptions of functions). Behavior functions can be distinguished clearly from effect and purpose functions: in behavior function descriptions physical conservation laws are taken into account, whereas this is not the case in effect and purpose function descriptions. For instance, in the description 'producing torque', the conservation of energy is not taken into account. In 'converting electricity into torque and heat' the input energy of electricity is supposed to equal the output energies of torque and heat, thus taking physical law into account.

4 In (Van Eck: 2009, 2010b) I analyze a number of accounts in terms of the notions of function and models of functional decomposition that they advance.

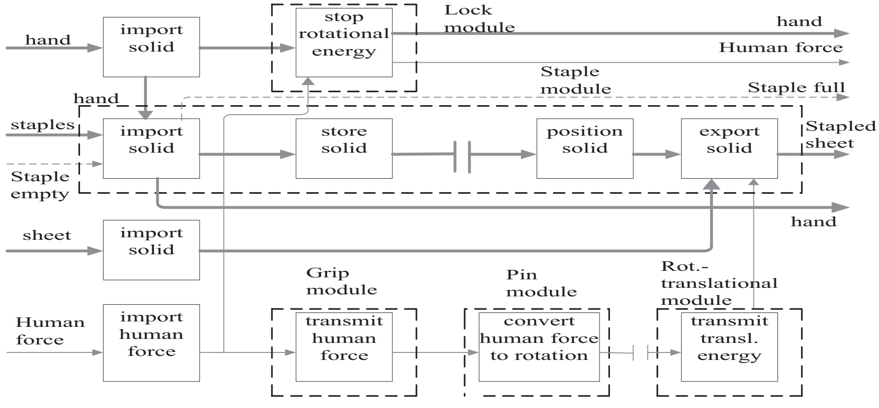

Figure 1: Behavior function fm_D of a stapler (adapted from Stone et al.: 2004)

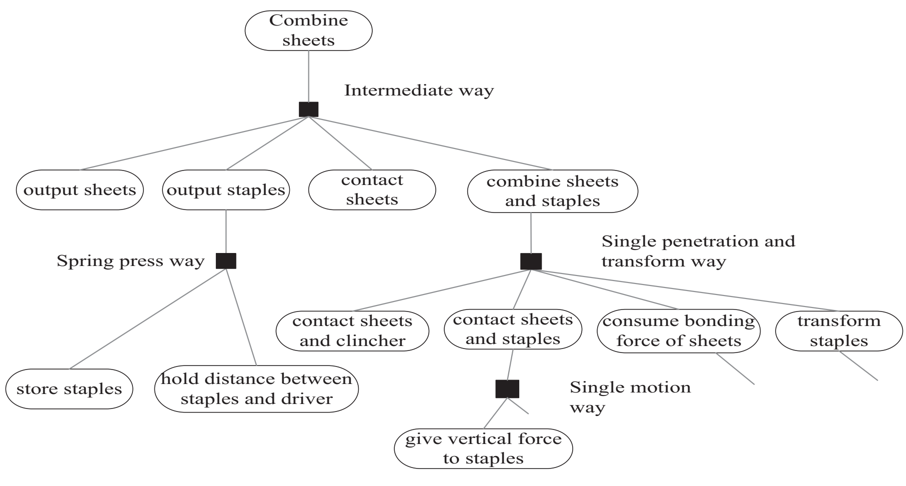

Figure 2: Effect function fm_D of a stapler (adopted from Ookubo et al.: 2007)

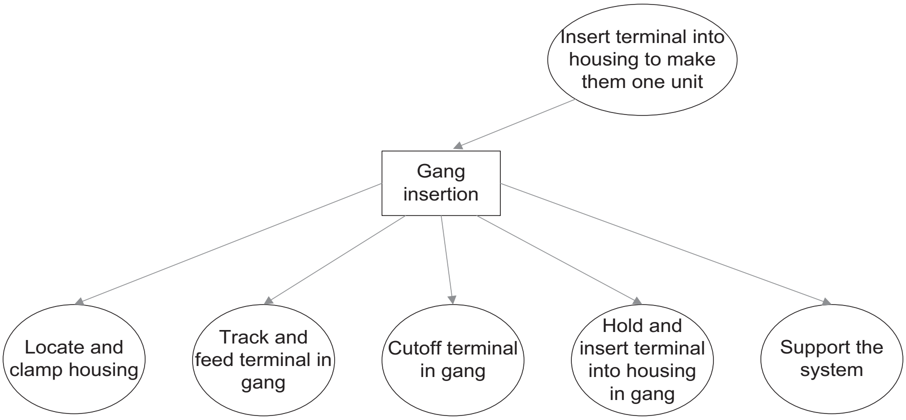

Figure 3: Purpose function fm_D of a terminal insertion device (adapted from Deng: 2002) [5]

5.3. Explaining co-existence by variation of objectives or design knowledge employment

Philosophical analyses of the usage, side-by-side, of different conceptualizations explicate this co-existence as having instrumental value to engineering (see Bucciarelli: 1994, 2003; Vermaas: 2009; Van Eck: 2010a). These analyses either relate specific conceptualizations (as suitable means) to specific objectives, or explicate the suitability of specific conceptualizations (as suitable means) to specific objectives in terms of design knowledge usage.

5.3.1. Object worlds

Based on analyses of several cases of actual engineering design practice, Bucciarelli advances the argument that engineers practice their trades in different "object worlds" (see Bucciareli: 1994, 62; Bucciarelli: 2003, 99). The notion of an object world(s) conveys:

---

5 These devices are a part of automatic assembly systems for manufacturing electronic connectors. They are used to insert terminals into a housing in order to make a conductor and an insulator one unit (see Deng: 2002)

> the idea that different participants in design see the object of design differently depending upon their competencies, responsibilities and their technical interests (Bucciarelli: 2003, 99).

Engineers from specific technical disciplines use conceptual frameworks in designing that are specific to their specialization; between technical disciplines there are differences in, amongst others, standards, regulations, mathematics, computer tools, and sketching and modeling tools (see Bucciarelli: 2003). Exponents from different disciplines hence will conceptualize an object of design in different ways. And they may also interpret a concept or notion that is shared across object worlds in different ways. As Bucciarelli (2003) illustrates:

> the same object, say a prismatic bar, to the structural engineer is a cantilever beam while to the person responsible for ensuring that the system does not overheat, it is a radiating appendage (Bucciarelli: 2003, 99).

These different conceptualizations have value since engineers from different object worlds work on different features of the object of design for which the adopted conceptualizations are useful (see Bucciarelli 2003: 9). Phrased differently, these conceptualizations are useful for achieving specific objectives. For instance, the above conceptualization of a prismatic bar as a radiating appendage is useful when one's objective is preventing a system to overheat, whereas a cantilever beam-conceptualization serves other ends.

Although conceptualizations between object worlds may be at variance in a given case of designing, requiring negotiation to arrive at design decisions (see Bucciarelli 2003: 101), these co-existing conceptualizations thus can be validated in terms of their being useful means to achieve a variety of objectives.

Whereas the above work of Bucciarelli validates co-existing conceptualizations between distinct engineering disciplines, the analyses of Vermaas (2009) and myself (2010a) support such conceptual divergence within an engineering discipline, to wit: functional modeling in electro-mechanical engineering.

5.3.2. Simplifying full descriptions of technical artifacts

Vermaas (2009) argues that specific meanings of the concept of technical function are used in engineering to advance specific descriptions of technical artifacts. Since these descriptions are all useful to engineering, he thus explains why the concept of function is used with more than one meaning in the field. He develops his analysis in terms of the notions of a full and a simplified description of a technical artifact. Vermaas identifies five key concepts in full descriptions of technical artifacts: *goals* of agents that refer to states in the world that agents desire to realize by using artifacts; *actions* that refer to intentional behaviors that agents carry out when using artifacts; *functions* that refer to desired roles played by artifacts; *behaviors* that refer to physicochemical state changes of artifacts; and *structures* that refer to the physicochemical materials and fields of artifacts, their configurations, and their interactions.

Vermaas (2009) argues that the flexible meaning of the concept of function affords different ways in which such full descriptions of technical artifacts can be simplified. Full descriptions in terms of the five key concepts are elaborate, and in particular engineering settings it makes sense to simplify them by "by-passing" one or more of the key concepts (see Vermaas: 2009, 2.119). Key to the analysis is that specific meanings of function are advantageous to specific by-passing simplifications. For instance, relative to the five key concepts, the concepts of action and behavior are by-passed in the account of Stone and Wood (2000) and the concept of function is used in its meaning of *desired behavior* (by specifying the role an artifact should play in terms of its behavior) to relate goals to structure. The concept of behavior is thus bypassed and the concept of function is instead used to refer to behavior(s). On the other hand, in the account of Lind (1994) the key concept of action is by-passed but not the concept of behavior. In this approach the concept of function is used in its meaning of *desired effect of behavior* (by specifying the role of the artifact in terms of the effects of the artifact's behavior) to relate goals to behavior.

This analysis thus explains co-existing meanings of the concept of function (and the accounts in which these meanings are advanced) as useful for the advancement of different simplified descriptions of technical artifacts (objectives), all valuable to engineering.

5.3.3. fm_D choice and knowledge usage

In a similar vein, but focusing on the notion of design knowledge usage, I have argued that the choice for (constructing) particular fm_Ds (behavior function fm_Ds, effect function fm_Ds, and purpose function fm_Ds) is influenced by whether or not particular design knowledge is employed in their construction. And that depending on the particulars of such design knowledge employment, particular models are best suited for achieving particular design objectives (see Van Eck: 2010a). I thus explained and defended the keeping of different fm_Ds side-by-side (and the accounts in which they are advanced) in engineering design in terms of variation in design knowledge employment.

Among others, I considered *fm*$_D$s that are used to support the objective of innovative design, characterized as the designing of new artifacts that have potentially novel (combinations of) function-structure connections (e.g., see Pahl and Beitz: 1988; Stone and Wood: 2000). Pahl and Beitz (1988) and Stone and Wood (2000) explicitly do not employ known function-structure connections (nor behavior-structure connections) in the construction of *fm*$_D$s, but establish such function-structure mappings after the completion of *fm*$_D$s. I argued that since known function-structure connections are not taken into account in the construction phase, behavior function *fm*$_D$s are best suited for the above objective since behavior function descriptions (which may include effects) are detailed enough to support the selection of (potentially novel) structures after the model is constructed. Purpose function *fm*$_D$s and effect function *fm*$_D$s, instead, are too coarse-grained to allow the selection of (potentially novel) structures in any precise way, when existing knowledge on function-structure connections is not considered in the construction phase of such models. The use of such models, skipping reference to behaviors and effects in purpose function *fm*$_D$s and to behaviors in effect function *fm*$_D$s, does not give (in a precise manner) those (potentially novel) structures that are suitable to achieve the functions of an artifact-to-be. For instance, a car's headlight effect-function "light on" may be suitable to select well-known structures such as an incandescent lamp or halogen one, but without a desired behavioral specification, the choice for, say, a more recent LED lamp (which differs in its behaviors by which the effect "light on" results) is less obvious.

On the other hand, I argued that other *fm*$_D$s get favored when their construction is based on employing known (and required) behavior-structure relations of an existing artifact. For instance, for the objective of design analysis, characterized as verifying whether the functions of an artifact are achieved by the behaviors of the artifact in the intended manner, *fm*$_D$s that are constructed based on known behavior-structure relations are used (e.g., see Lind: 1994; Bell et al.: 2007). Given that both behavior and structure of an artifact are known, effect function *fm*$_D$s are suited for verifying whether the behaviors exercised by structures achieve (in the intended fashion) the functions that are desired. Using a purpose function *fm*$_D$, instead, skipping reference to effects, does not give the precision to ascertain whether or not the functions are indeed manifested *in the intended way* by the behaviors of the artifact. For instance, the purpose function "illumination in a room" may be sufficient for determining whether the behavior of the artifact implements this function. Yet, it is not suited to verify whether the behavior of the artifact implements this function in the intended way. In contrast, an effect function description, say, "switch on-light on", is suited for verifying whether the behavior of the artifact implements this function in the intended way: say, the switch might be "off" while the light is still on. Such a failure goes undetected with the purpose function description "illumination in a room" (more elaborate behavior function fm_Ds may also do the trick, but seem unnecessarily complex). Hence, given that behavior and structure of an artifact are known, effect function fm_Ds are best suited for verifying whether the functions of an artifact are achieved by the behaviors of the artifact in the intended manner.

My analysis, like the ones of Bucciarelli (1994) and Vermaas (2009), shows the instrumental value of maintaining co-existence, in casu of different fm_Ds: depending on the specifics of the design knowledge employed, particular models are best suited for achieving particular objectives.[6] This explanation however holds (in informative fashion) for certain cases only. That is, insofar as the knowledge used in building models does not contain or refer to a specific notion of function or fm_D, co-existence of models can be understood in terms of variation in knowledge usage. Yet, when the knowledge used does already refer to a specific notion of function or fm_D, the choice for (the construction of) specific models is obvious, but explicating such choices in terms of knowledge employ-

6 To avoid misunderstanding, I am not pressing the claim that an objective fixes a specific knowledge usage, which in turn fixes what counts as the most adequate model. An objective then would fix the most adequate model. I am, rather, advancing the claim that (the choice for) a specific knowledge usage impacts the suitability of a model. The choice to employ specific knowledge may differ between modeling accounts, whereas the objective they target is the same. For instance, one may envision the design strategy to use known function-structure connections for building models in innovative design under the assumption that it reveals when such knowledge is insufficient to take care of all required functionalities, indicating that new knowledge on (novel) function-structure connections is required (see Chakrabarti and Bligh: 2001). This differs from the design strategy of Stone and Wood (2000) in which knowledge of function-structure connections is explicitly not employed during the construction of models, but only after models are built. Both these choices with respect to knowledge use seem sensible ones for achievement of the objective of innovative design. So, knowledge usage in the construction of models is the crucial parameter in my analysis to explicate the choice for and co-existence of models.

ment would become circular. For instance, in the case of the objective of routine designing (characterized as the designing of new artifacts by using knowledge of function-structure connections of existing types of the to-be-designed artifact). Deng (2002) puts forward purpose function fm_Ds that are build using known connections between purpose functions and structures as means toward this objective. Since these connections are known and employed, purpose function fm_Ds are obviously opted for. However, explicating the choice for these fm_Ds in terms of the usage of known purpose function-structure connections would introduce circularity in the explanation. Moreover, as I will argue in the next section, effect function fm_Ds and behavior function fm_Ds that are also built using connections between these notions of function and structure, respectively, are advanced as well as means toward this objective of routine designing. Hence, different explanatory constructs than variety in objectives and variation in knowledge usage are needed to explain cases such as these. These constructs do not provide the requisite explanatory leverage when one wants to explain why different fm_Ds are used side-by-side as means for achievement of the same objective, and the knowledge used to build them already refers to specific notions of function or fm_D. In such cases, other explanatory constructs are needed. The work of Kuhn and others on methodological incommensurability and the dynamics of theory choice provide concepts suited to explicate such cases, as I will argue in the next section.

5.4. Methodological incommensurability in engineering

Kuhn (1970, 148-150) initially used the term incommensurability in a holistic fashion to capture methodological, observational, and conceptual incompatibilities between successive scientific paradigms. In later work (e.g., see Kuhn: 1991) he narrowed down and specified his notion of incommensurability further in terms of differences in the taxonomic structure of successive scientific theories. On this "semantic" reading of incommensurability, translation failure occurs between kind terms of competing theories due to the unmatchable classificatory schemes/taxonomic structures underlying these theories (see Kuhn: 1991). In such cases, theories classify the same objects into different kinds, the members of which are (taken to be) governed by different natural laws. Translation of kind terms between theories then will fail since the nomic expectations attached to these terms are incompatible between theories. For instance, Ptolemy's theory classifies the sun as a planet, where planets orbit around the earth, whereas

Copernicus' theory classifies the sun as a star, where planets orbit stars. A Copernican claim such as planets orbiting the sun is incompatible with Ptolemy's framework, hence translation of the kind term 'sun' between these theories will fail (see Kuhn: 1991, 94).

As Kuhn's later treatment of incommensurability focused mainly on semantic aspects, some commentators began to distinguish two different notions of incommensurability: on the one hand the above-mentioned semantic incommensurability and on the other "methodological" incommensurability, which involves epistemic standards that are used to evaluate competing theories (see Kuhn: 1970, 1977; cf. Sankey: 1999; Carrier: 2008; Soler: 2008; Oberheim and Hoyningen-Huene: 2009).

5.4.1. Methodological incommensurability

The development of the thesis of methodological incommensurability is traced back to Kuhn's (as well as Feyerabend's) rejection of the view held by both the Logical Positivist movement and Popper that a distinguishing feature of science is the use of a uniform scientific method that remains fixed throughout scientific development, and on the basis of which theory choice can be determined unambiguously (see Oberheim and Hoyningen-Huene: 2009; cf. Kuhn: 1970, 94, 103).[7] Kuhn challenged the view of an invariant scientific method that is capable of governing theory choice in such unambiguous fashion. He argued, instead, that standards or criteria of theory appraisal, such as accuracy, consistency, fruitfulness, scope, and simplicity (1977, 322) depend on and vary between paradigms. Kuhn pressed the point that such standards do not function as algorithmic rules that are able to determine theory choice but rather as values that only guide it (1977, 331). Epistemic values refer to characteristics or properties of scientific theories that are considered desirable by scientists relative to their objectives. The history of science shows that disputes between advocates of rival theoretical frameworks are not solved by recourse to a neutral algorithm that is capable of dictating theory choice, since there is no commonly shared set of criteria or standards available on the basis of which such a choice can be

7 See Worrall (1988) for a more recent defense of a fixed scientific method.

forged (see Kuhn: 1970, 1977).[8] Based on this construal of theory choice in terms of values and the observation that scientists (can) differ in the values they employ, Kuhn (1977) concluded that scientists may rationally disagree in theory choice. This disagreement may have different sources. First, advocates of rival scientific frameworks may differ in the values they employ in theory choice and appraisal. Second, values may conflict when applied to concrete cases of theory choice. For instance, scope may favor one theory, yet simplicity another. Theory choice then entails assigning weight/relevance to such values, which advocates of rival frameworks may do so in different fashion. Third, advocates of rival scientific frameworks may also interpret the content of values differently. What is, for instance, precisely meant when one speaks about accuracy? Based on these considerations, Kuhn (1970, 1977) concluded that there is no commonly shared algorithm available for theory choice.

Summing up, key elements of this position are the closely related notions of "non-algorithmic theory choice" and "methodological variation" (see Sankey: 1995, 2002), that is, variation in how and/or which (set of) values are employed in theory choice. Furthermore, the theories in question are advanced to meet what we may call a 'common objective': they purport to explain the same (or substantially overlapping) range of phenomena (see e.g., Soler: 2008). I use (and expand on) the notion of variation in values in section 4.2 to explain co-existence of different engineering fm_Ds that are advanced as means to achieve a *common objective*.[9]

My earlier explanation of co-existence in terms of variation in knowledge usage (section 3.3) also hinges upon (though not phrased as such), in an engineering-modeling rather than a scientific-theoretical context, the idea of variation in values: knowledge usage specifics, such as employing known function-structure connections or behavior-structure relations during the construction of fm_Ds

8 Rival in the sense that these theories purport to explain the same or (overlapping) range of phenomena (see e.g., Soler: 2008). Otherwise, incommensurability issues do not arise of course.

9 Semantic incommensurability does not provide the relevant footing for explicating co-existence: translations between different fm_Ds are, after doing some conceptual groundwork, possible. Behavior function fm_Ds can, for instance, be translated into physical behavior models after which the relevant information can be extracted from these behavior models to construct effect function fm_Ds (see Van Eck: 2009/2010, 2010c).

correspond to values engineers have that influence their choices for particular models. These values are not ones that are operative in a scientific-theoretical context (epistemic values) but they do function similarly, in an engineering-modeling context, as factors that influence engineers their choices for particular *fm*$_D$s. Let us capture this similarity by calling such factors engineering-values, or "e-values" for short. I define an engineering value as a characteristic or property of a functional decomposition model or a functional decomposition strategy that is considered desirable by an engineer relative to an objective.[10] [11] However, as indicated in section 3.3, e-values relating to knowledge usage do not provide the requisite explanatory leverage in the case of routine designing. I consider other e-values to explicate this case.

5.4.2. Incommensurability in engineering: The case of functional decomposition and routine designing

In the engineering literature, in the electro-mechanical domain, different *fm*$_D$s are advanced as means for achieving the (common) objective of "routine design": behavior function *fm*$_D$s (see Chakrabarti and Bligh: 2001), effect function *fm*$_D$s (see Kitamura and Mizoguchi: 2003), as well as purpose function *fm*$_D$s (see Deng et al.: 2000a, 2000b; Deng: 2002) are put forward as means for achieving this objective.[12] In the above accounts in which these particular models are advanced this objective is characterized as the designing of new artifacts by using knowledge of function-structure connections of existing types of the to-be-designed artifact (references see above).

---

10 Applying the notion of variation in values to engineering rather than science is unproblematic. Values are not specific to science (e.g., see McMullin: 1983). In different contexts, in casu science and engineering, values convey the (same) idea that a characteristic or property of an item or entity is considered desirable. The more discriminative notions of epistemic value and engineering value, of course, are specific to these contexts and relate to different items: epistemic values relate to scientific theories, and engineering values relate to *fm*$_D$s or strategies.

11 E-values relating to knowledge use refer to the process of building *fm*$_D$s, i.e., to functional decomposition strategies. The e-values that I consider in the next section refer to features of *fm*$_D$s. One can rephrase this process feature as a model feature. For instance, not employing known function-structure connections in model building can be rephrased as, say, 'function-structure independency'. In the engineering literature, the term "form-independent" (see Stone and Wood: 2000, 359) is often employed.

12 In (Van Eck: 2009) I spell out the claim that these *fm*$_D$s are put forward in these accounts.

5.4.2.1. Variation of e-values

Analysis of these accounts shows that their developers advance different e-values that their proposed fm_Ds are to satisfy. These e-values are given in Table 1.

Structural compatibility	The spatial organization that an fm_D provides must be such that all functions of the structures contained in the spatial organization are achieved
Function-behavior independency	Descriptions of functions in an fm_D should be such that they do not describe their underlying behavior and are organized in sets in terms of knowledge of their underlying behavior
Function-to-function independency	The functions in an fm_D must be independent from one another in the sense that realization of a given function by a structure is (considered to be) independent from realization of other function(s), and vice versa

Table 1: e-values employed in fm_D choice

In the Chakrabarti-Bligh (CB) account fm_Ds must satisfy the e-value of what we may call 'structural compatibility'. The (input-output) organization of functions in an fm_D also provides a spatial organization of the structures that achieve them.[13] And the spatial organization that a model provides must be such that any negative interactions between structures (as a result of which structures would fail to achieve their functions) do not occur, so that all the functions of the structures contained in the spatial organization are achieved (see Chakrabarti

[13] I borrow the term spatial organization from the mechanistic explanations-literature (e.g., see Machamer et al.: 2000).

and Bligh: 2001). In other words, structures contained in the spatial organization provided by an fm_D must be compatible with one another.[14]

In the Functional Concept Ontology (FCO) account (see Kitamura and Mizoguchi: 2003; Kitamura et al.: 2005/6; Kitamura et al.: 2007) another e-value, which we may call 'function-behavior independency' is emphasized. This e-value prescribes that descriptions of functions in fm_Ds should be such that they do not describe their underlying behavior and are organized in sets in terms of knowledge of their underlying behavior (descriptions that do refer to underlying behavior are coined "quasi-functions"). These authors distinguish the concept of function from the concept of behavior. And in fm_Ds those functions that make up another function are grouped together (organized) in sets based on knowledge of their underlying behavior and structure, thus distinguishing functional from behavioral descriptions.

Yet another e-value is emphasized in the Dual Stage (DS) account (see Deng et al.: 2000a, 2000b; Deng: 2002), which we may call 'function-to-function independency'. This e-value prescribes that the functions in an fm_D must be independent from one another in the sense that realization of a given function by a structure does not depend on the (prior) realization of another function by another structure (see Deng: 2002). For instance, a washing machine's function of 'washing laundry' can be independent (for its realization) from its function of 'drying laundry' (see Deng: 2002). (In this account, the behaviors underlying the functions in an fm_D are not considered to be independent, but causally related).

As can be seen, different e-values for fm_Ds hold in these accounts. This variation in e-values provides means to explain the choice for/construction of different fm_Ds in these accounts, as I will argue below.

14 fm_Ds are the end result or product of a series of design reasoning steps. They are constructed in step-by-step fashion by first selecting (from a knowledge base) a function and an associated structure, then it is assessed which functionality is solved and which still remains to be solved, then another function and associated structure (that is compatible with the first selected structure) are selected, then again an assessment is made of the solved and still unsolved functionalities, after which again a function and structure are selected (compatible with the already selected structures) etcetera, until these selected functions jointly achieve an overall function/achieve all required functionalities (due to the spatial organization of their associated structures). The end result of this process is an fm_D that satisfies the structural compatibility e-value.

5.4.2.2. Explaining fm_D choice by e-values

Given the emphasis in the CB account on the e-value of structural compatibility, one can understand why behavior functions fm_Ds are (chosen to be) developed. When the spatial organization that an fm_D provides must be such that any negative interactions between structures (as a result of which structures would fail to realize their functions) do not occur, behavior function fm_Ds are best equipped to provide such a spatial organization. Such fm_Ds contain the details needed for assessing whether the output characteristics of one structure's function match/are compatible with the input characteristics of another structure's function. Say, the heat generated when energy is converted into torque by an electric screwdriver's motor may negatively interact with the electrical wiring connected to the motor, possibly leading to failure of their 'transmitting electricity' function (and hence the motor's function as well). Purpose and effect function fm_Ds seem too course-grained to satisfy this e-value of structural compatibility. For instance, the effect function 'produce torque' of a screwdriver's motor does not contain the information required to assess its compatibility with the electrical wiring

In the FCO account, on the other hand, structural compatibility is already assumed to be in place. In these fm_Ds, those functions that make up another function are grouped together (organized) in sets based on knowledge of their underlying behavior and structure (see Kitamura et al.: 2005/6). One needs to assume that the structures (and behaviors) underlying the functions in fm_Ds are compatible for otherwise sets of functions making up/achieving other functions would fail to do so (these authors make this assumption: fm_Ds are models of existing and working artifacts).

In the DS account, structural compatibility is not something that fm_Ds should satisfy. Rather, in this account, both the assembly of structures and the verification of whether assembled structures meet the design requirements take place in later design phases after fm_Ds are constructed (see Deng: 2002).[15]

[15] To be sure, the constraint that all structures are compatible with one another (and, hence, all functions realized) is of course a crucial constraint that is valued in all functional modeling frameworks. Modeling frameworks differ, however, in which design phase this value is to be satisfied. In some accounts fm_Ds should satisfy this value (see Chakrabarti and Bligh: 2001), whereas in others it should be satisfied in later design phases (see Deng et al.: 2000; Deng: 2002). Thus, only in some accounts is it a value that applies to fm_Ds.

Next to the structural compatibility assumption, *fm*$_D$s in the FCO account must satisfy the e-value of function-behavior independency. Given this e-value one can understand why effect function *fm*$_D$s are developed. When descriptions of functions in *fm*$_D$s must be such that they do not describe their underlying behavior and are organized in terms of their underlying behavior (and structure), behavior function *fm*$_D$s will (obviously) not be opted for since the functions in such models describe behaviors. And since functions in *fm*$_D$s are grouped together based on knowledge of their underlying behavior and structure one can also understand why purpose function *fm*$_D$s are not chosen. By using purpose function *fm*$_D$s, in which functions refer to states of affairs that are the final result(s) of behavior, one skips reference to the more immediate effects of behaviors and structures. Compared with effect function *fm*$_D$s, the grouping of functions in sets based on their underlying behavior and structure is less straightforwardly established with such purpose function *fm*$_D$s. In the latter case, the connection between function-behavior-structure is less straightforward. For instance, the purpose function "to tell time" can be achieved by a wide variety of behaviors and structures. The effect function description "rotate arms in clockwise direction" on the other hand is more easily connectable to specific behaviors and structures (and sets of such functions thus more straightforwardly organized in terms of behavioral and structural knowledge).

In the CB account, this function-behavior independency is not an e-value that *fm*$_D$s must satisfy. On the contrary, as we saw, in these *fm*$_D$s functions refer to behaviors. Neither do *fm*$_D$s satisfy this e-value in the DS account. Functions in purpose function *fm*$_D$s are not organized in terms of knowledge of their underlying behavior (what they do have in common with *fm*$_D$s in the FCO account is that functions in DS *fm*$_D$s do not describe their underlying behavior since they characterize states of affairs that are the final results of behaviors).

Given the third e-value of function-to-function independency that *fm*$_D$s in the DS account must satisfy, one can understand why purpose function *fm*$_D$s are developed. When functions in an *fm*$_D$ are required to be independent from one another in the sense that realization of a given function is independent from the realization of other function(s), and vice versa, purpose function *fm*$_D$s seem most suited. Such models allow one to conceive most clearly of the realization of functions as being independent from the realization of other functions. For instance, realization of the behavior function 'transmitting torque' of an electric screwdriver requires, say, prior realization of the behavior function 'converting electricity into torque'. Similarly, realization of the electric screwdriver's effect function 'produce torque' requires, say, prior realization of the effect function 'generate electricity'. In contrast, realization of the purpose function of, say, 'having a rotational force' is more easily conceived as independent from the realization of other functions. Hence, models of purpose functions satisfy this e-value best.

In contrast, function-to-function independency is not an e-value in the FCO account since functions in fm_Ds that jointly achieve another function are grouped in sets (based on knowledge of their underlying behavior and structure) and hence not (considered to be) independently realized. Fm_Ds in the CB account also do not satisfy this e-value. Functions in fm_Ds are organized in terms of their input-output characterizations and thus for their realization dependent on one another (and on the structural compatibility of their underlying structures). This analysis is summarized graphically in Table 2.

	Structural compatibility	Function-behavior independency	Function-to-function independency	
CB account	+	-	-	Behavior function fm_D
FCO account	+	+	-	Effect function fm_D
DS account	-	-	+	Purpose function fm_D

Table 2: e-values and fm_D choice

5.4.2.3. Co-existence and methodological incommensurability

Summing up, in these accounts different e-values are considered important, due to which different fm_Ds are chosen for routine designing: behavior function fm_Ds in the CB account, effect function fm_Ds in the FCO account, and purpose function fm_Ds in the DS account. And, in addition, some of these e-values conflict: function-to-function independency applies to DS purpose function fm_Ds but not to CB behavior and FCO effect function fm_Ds; and function-behavior independency applies to FCO effect function fm_Ds but not to DS purpose function and CB behavior function fm_Ds. Of the three sources that each leads to methodological incommensurability (see section 5.1), two engineering variants can be identified in this case: (1) variation in and (2) conflict between e-values. Due to both this variety of and conflict between e-values, there is in this case no commonly shared algorithm available that governs engineers' their choices of fm_Ds. This choice, rather, is seen to be dependent on the e-values that engineers adopt. This divergence of e-values thus explains the co-existence of different fm_Ds that are advanced as means to achieve a common objective. I submit that this functional decomposition case exemplifies an instance of methodological incommensurability in the engineering domain.[16]

In a similar vein as Kuhn (1970, 1977, 1983) explained scientists' choices for different theories in terms of differences in epistemic values, I thus offer an explanation why different fm_Ds are used side-by-side in engineering in terms of variation in e-values. Kuhn's analysis of values, in addition, led him to conclude that scientists' choice of (competing) theories can be considered rational. This conclusion has spawned extensive debate in philosophy of science (see Kuhn: 1977; McMullin: 1983; Laudan: 1987; Worrall; 1988: Sankey; 1995, 2002). Initially, a key issue was whether in the absence of a commonly shared algorithm scientists' choice of theories can in fact be considered rational. More recently, this debate has shifted in orientation: both advocates of a single method for theory choice and authors that accept variation in values are pressed to show that their preferred single method or spectrum of values ensure the rationality of scientists' choice of theories (see Worrall: 1988; Sankey: 1995, 2002).

I will address this issue in the engineering functional decomposition case: can engineers' choices for different fm_Ds be considered rational from an instrumental point of view? I will argue that variation in e-values ensures that the choice and usage of different fm_Ds by different engineers is rational from a practical point of view.

---

16 Perhaps someone might object to this latter conclusion/existence proof on the grounds that the term incommensurability should be reserved to a scientific-theoretical context, period. I disagree but if this causes too much cognitive dissonance, let us not skirmish over words. My purpose in this paper is to understand co-existence of distinct fm_Ds for the same objective and (expansion into engineering of) the thesis of methodological incommensurability allows me to do so.

5.5. Rationality in engineering

5.5.1. Values and theoretical rationality

Kuhn (1983) took the position that the rationality of scientists' choice of theories is ensured by the concept of science itself (see also Sankey: 1999). Kuhn's position has however been criticized on the grounds that he never satisfactorily addressed the challenge to explicate how variation in epistemic values ensures rational theory choice (see Hempel: 1983; Sankey: 1999). In the case of values, the challenge is to show that the values one considers are appropriate ones for the evaluation and choice of scientific theories. A value is considered appropriate for theory choice if a theory that satisfies a particular value contributes to the attainment of a scientific objective (that one aims to achieve with the theory) precisely because the theory satisfies that value (see Hempel: 1983; McMullin: 1983; Sankey: 2002). Stated differently, that the (desired) characteristics or properties of the chosen theory indeed are the features by means of which the theory contributes to attainment of an objective that one aims to achieve with the theory. Insofar as values are appropriate, maintaining variation of these values in theory choice is considered rational. Advocates of value variation consider such means-end interpretations of values an asset (see Laudan: 1987; Teller: 2008). It allows for the possibility to rationally compare the merits of competing theories (or scientific models): this theory/model is better with respect to this value, that theory/model is better with respect to that value.

Several interpretations of such means-end relationships between epistemic values and scientific objectives are given in philosophy of science. Some assert that appropriate values contribute to the attainment of a main or ultimate objective of science, such as empirical adequacy or truth (see McMullin: 1983). Others do not invoke the notion of an ultimate objective and argue that specific values contribute to more specific objectives (see Laudan: 1987; Teller: 2008). Sankey (2002) gives a third interpretation by combining the two interpretations above. Sankey views these more specific objectives as subordinate to a main or ultimate objective of science, which in his book is advancement on the truth. He takes the achievement of subordinate objectives as sub serving this main or ultimate objective of science. In Sankey's scheme on thus finds epistemic values, their related subordinate objectives, and a common ultimate objective

Based on this means-end interpretation of epistemic values, Sankey (2002) defends methodological variation: insofar as values are conducive to the realization of their related subordinate objectives, maintaining variation of these values in theory choice is rational.[17]

I use and expand on Sankey's means-end analysis of epistemic values, specifically his distinction between subordinate objectives and a main objective, to show that the e-values that I consider are appropriate ones for the evaluation and choice of fm_Ds. In the engineering case I speak of sub objectives rather than subordinate ones. As I will argue in the next section, this analysis indicates that engineers' choices for different fm_Ds are rational from an instrumental point of view.

5.5.2. E-values and practical rationality

I will demonstrate in the following that an fm_D that satisfies a particular e-value contributes to the attainment of the objective for which the fm_D is used precisely because it satisfies that e-value, i.e., that the (desired) characteristics or properties of the chosen fm_D indeed are (among) the features by means of which the model contributes to attainment of the objective for which it is used. To demonstrate that fm_Ds satisfying the e-values that I consider are suitable means to achieve the objectives for which they are used, I distinguish between main and sub objectives of engineers. I argue that different fm_Ds, precisely because they satisfy particular e-values, directly contribute to the attainment of particular sub objectives and indirectly, via the achievement of sub objectives, to main objectives. This analysis in terms of sub objectives makes it insightful how different fm_Ds that satisfy different (and conflicting) e-values all contribute to a common main objective. This analysis thus also indicates that specific models have specific advantages: depending on the e-values and sub objectives that engineers have, specific fm_Ds are better than others. By implication, my analysis shows that the usage of different fm_Ds by different engineers is rational from a practical point of view.

---

[17] Hoyningen-Huene (1992) endorses a similar position, arguing that values are "something like execution procedures" (498) for the ultimate goal of science "to produce general, explanatory theories about the world" (499), and that they "concretize this goal in an operationally meaningful way" (499).

Returning to the first e-value of structural compatibility that is satisfied by *fm*$_D$s in the CB account, we can explicate these *fm*$_D$s as contributing to a sub objective of what we may call "accuracy", to wit: that all the functions in an *fm*$_D$ are realized. In order for this sub objective to be achieved an *fm*$_D$ must satisfy structural compatibility: the spatial organization that an *fm*$_D$ provides must be such that any negative interactions between structures do not occur, so that all the functions of the structures contained in the spatial organization are realized.[18] Since this e-value is already assumed to be satisfied in the FCO account, so is its related sub objective. In the DS account, this e-value and sub objective are addressed in later design stages after *fm*$_D$s are constructed.

In similar fashion we can interpret *fm*$_D$s satisfying the e-value of function-behavior independency, as endorsed in the FCO account, as contributing to a sub objective of what we may call "knowledge management of design rationale". This account aims to capture (rather ambitiously) the rationale of engineers that lies behind their construction of particular functional descriptions and *fm*$_D$s (for archival and cross-communication purposes in design) (see Sasajima et al.: 1996; Kitamura et al.: 2007). Capturing such "design rationale" is according to these authors in engineering done in an idiosyncratic fashion in the sense that its analysis depends on the considerations of the model builder. They aim to overcome this idiosyncrasy by developing systematic guidelines for the capturing of design rationale behind *fm*$_D$s in more explicit and re-usable fashion. Key assumption in the development of these guidelines is that of all the possible input-output relations of technical behaviors only some input, output, or input-output relations are intended in a given context and hence will be used for developing functional descriptions and *fm*$_D$s. They also define primitives to isolate those input, output, or input-output relations that are used to develop descriptions of functions and *fm*$_D$s in particular contexts (see Sasajima et al.: 1996). Given this aim to capture design intent systematically, that is, the sub objective of "knowledge management of design rationale", and this key assumption underlying it, we can interpret *fm*$_D$s satisfying the e-value of function-behavior independency as contributing to this sub objective. Given this underly-

---

18 Chakrabarti indicated – personal communication on August 26, 2009, Stanford, CA, USA – that the account he developed with Bligh is explicitly geared toward satisfying these, what I labeled, e-value and sub objective by means of the steps described in note 14. The assumption that they are already satisfied when using knowledge of existing artifacts in routine designing is in his view often negated by actual design cases.

ing assumption, fm_Ds satisfying the e-value of distinguishing function from its underlying behavior contribute to capturing design intent in systematic fashion. This e-value and sub objective are not emphasized in the CB and DS accounts.

Fm_Ds satisfying the e-value of function-to-function independency, as endorsed in the DS account, can be analyzed as contributing to a sub objective that we may call "broad scope in function-structure mapping". If functions-structure connections can be considered independent from other function-structure connections, one can search the available spectrum of design solutions to a given function. If the realization of a function by a structure would depend on the (prior) realization of another function by another structure, the range of structure-function connections would decrease. A selection of a particular design solution to a function would then constrain the possible design solutions one can choose for functions that must be realized prior to this function. By considering function-structure connections as independent, this constraint does not apply. Hence, a broad range of functions-structure connections can be considered

Achievement of each of these sub objectives, in turn, all contributes to the main (and common) objective of routine designing. The sub objective of accuracy that all the functions in an fm_D are realized is crucial to the design of any artifact, irrespective of whether it is arrived at in routine or innovative fashion. Achievement of the sub objective of establishing knowledge management of design rationale – facilitating the consistent archival and cross-communication of design knowledge – is clearly instrumental toward the designing of artifacts in collaborative settings. And achievement of the sub objective of having broad scope in function-structure mapping, i.e., keeping the range of potential structures for functions as broad as possible, may support 'innovative/creative' combinations of structures of an artifact-to-be.

We thus reach the conclusion that the e-values that I considered are appropriate ones for the evaluation and choice of fm_Ds: particular fm_Ds are suited to achieve particular sub and main objectives because these fm_Ds satisfy particular e-values. This analysis in terms of e-values shows that specific models have specific advantages: depending on the e-values (and sub objectives) that engineers deem important, specific fm_Ds are better than others. For instance, if one values compatibility of structures, then one better opts for a behavior function fm_D; if one values independence of function-structure connections, one better picks a purpose function fm_D. There is not one fm_D that satisfies all such engineering values best. Hence, I submit that the usage of different fm_Ds by different engineers is rational from a practical point of view.

A qualification is in order. From the analyzed case it does not automatically follow that functional modeling research will not eventually converge toward a single fm_D. What the analysis does show is that modeling researchers have valid reasons not to do so, and my bet is that they will not. Another issue is whether the modeling field will eventually settle on a best behavior function fm_D, effect function fm_D, and purpose function fm_D, respectively. Given the current plethora of functional modeling accounts, it may turn out at some point in the future that the current situation is then interpreted as, say, "pre-paradigmatic", and accounts will have converged toward, say, three best accounts for the modeling of behavior function fm_Ds, effect function fm_Ds, and purpose function fm_Ds, respectively. My bet is that this scenario is unlikely as well: closer scrutiny will probably reveal other e-values and sub objectives that are served especially well with particular variants of the three considered fm_Ds. For instance, effect function fm_Ds in which the functions are represented by triggers and effects (see the "switch on-light on" example in section 3.3) seem better suited for failure analysis than effect function fm_Ds in which functions are represented in term of desired output only (e.g. "light on").

5.6. Conclusion

In this paper I have explained the co-existence of different models of functional decomposition in terms of the thesis of methodological incommensurability. I advanced this analysis in terms of the thesis' construal of (non-algorithmic) theory choice in terms of values, expanding this notion to the engineering domain. I further argue that co-existence of different models of functional decomposition is rational from an instrumental point of view.

Acknowledgements

I thank two anonymous reviewers for their useful comments. In addition, comments from Maarten Franssen, Paul Hoyningen-Huene, Peter Kroes, and Pieter Vermaas proved very rewarding in shaping this paper. This research is supported by the Netherlands Organization of Scientific Research (NWO).

References

Bell, J., Snooke, N., and Price C. (2007) 'A Language for Functional Interpretation of Model Based Simulation', in: *Advanced Engineering Informatics* 21: 398-409.

Bucciarelli, L.L. (1994) *Designing Engineers*, MIT Press, Cambridge MA.

Bucciarelli, L.L. (2003) *Engineering Philosophy*, Delft University Press, Delft.

Carrier, M. (2008) 'The Aim and Structure of Methodological Theory', in: Soler, L., Sankey, H. and Hoyningen-Huene, P. (eds.), *Rethinking Scientific Change and Theory Comparison: Stabilities, Ruptures, Incommensurabilities?*, Springer, Dordrecht: 273-290.

Chakrabarti, A. and Bligh, T.P. (2001) 'A Scheme for Functional Reasoning in Conceptual Design', in: *Design Studies* 22: 493-517.

Chandrasekaran, B. (2005) 'Representing Function: Relating Functional Representation and Functional Modeling Research Streams', in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 19: 65-74.

Chandrasekaran, B. and Josephson, J.R. (2000) 'Function in Device Representation', in: *Engineering with Computers* 16: 162-177.

Deng, Y.M. (2002) 'Function and Behavior Representation in Conceptual Mechanical Design', in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 16: 343-362.

Deng, Y.M., Tor, S.B., and Britton, G.A. (2000a) 'A Dual-Stage Functional Modelling Framework with Multi-Level Design Knowledge for Conceptual Mechanical Design', in: *Journal of Engineering Design* 11 (4): 347-375.

Deng, Y.M., Tor, S.B., and Britton G.A. (2000b) 'Abstracting and Exploring Functional Design Information for Conceptual Mechanical Product Design', in: *Engineering with Computers* 16: 36-52.

Erden, M.S., Komoto, H., Van Beek, T.J., D'Amelio, V., Echavarria, E, and Tomiyama, T. (2008), 'A Review of Function Modelling: Approaches and Applications', in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 22: 147-169.

Hempel, C.G. (1983) 'Valuation and Objectivity in Science', in: Cohen, R.S., and Laudan, L. (eds.), *Physics, Philosophy and Psychoanalysis: Essays in honor of Adolf Grunbaum*, Reidel, Boston: 73-100.

Hoyningen-Huene, P. (1992) 'The Interrelations between the Philosophy, History and Sociology of Science in Thomas Kuhn's Theory of Scientific Development', in: *British Journal for the Philosophy of Science* 43: 487-501.

Kitamura, Y. and Mizoguchi, R. (2003) 'Ontology-Based Description of Functional Design Knowledge and its Use in a Functional Way Server', in: *Expert Systems with Applications* 24: 153-166.

Kitamura, Y., Koji, Y., and Mizoguchi, R. (2005/2006) 'An Ontological Model of Device Function: Industrial Deployment and Lessons Learned', in: *Applied Ontology* 1: 237-262.

Kitamura, Y., Takafuji, S., and Mizoguchi, R. (2007) 'Towards a Reference Ontology for Functional Knowledge Interoperability', in: *Proceedings of the ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE '07, September 4-7, 2007, Las Vegas, Nevada, USA*. DETC2007-35373: 1-10.

Kuhn, T. S. (1970) *The Structure of Scientific Revolutions*, University of Chicago Press, Chicago.

Kuhn, T.S. (1977) 'Objectivity, Value Judgment and Theory Choice', in: Kuhn, T.S., *The Essential Tension*, University of Chicago Press ,Chicago: 320-339.

Kuhn, T.S. (1983) 'Rationality and Theory Choice', in: *Journal of Philosophy* 80: 563-570.

Kuhn, T. S. (1991) 'The Road since Structure', in: *Proceedings of the 1990 Biennial Meeting of the Philosophy of Science Association* 2: 3-13.

Laudan, L. (1987) 'Progress or Rationality: The Prospects for Normative Naturalism', in: *American Philosophical Quarterly* 24: 19-31.

Lind, M. (1994) 'Modeling Goals and Functions of Complex Industrial Plants', in: *Applied Artificial Intelligence* 8: 259-283.

Machamer, P.K., Darden, L., and Craver, C.F. (2000) 'Thinking about Mechanisms' (pp. 249-74)., in: *Philosophy of Science* 57: 1-25.

McMullin, E. (1983) 'Values in Science', in: *Proceedings of the 1992 Biennial Meeting of the Philosophy of Science Association* 2: 3-28.

Oberheim, E. and Hoyningen-Huene, P. (2009) 'The Incommensurability of Scientific Theories', in: *Stanford Encyclopedia of Philosophy*: 1-27. Retrieved at 15-05-2009.

Ookubo, M., Koji, Y., Sasajima, M., Kitamura, Y., and Mizoguchi, R. (2007) 'Towards Interoperability Between Functional Taxonomies Using an Ontology-Based Mapping', in: *Proceedings of the International Conference* *on Engineering Design, ICED '07, August 28-31, 2007, Paris, France*: 154.1-154.12.

Pahl, G. and Beitz, W. (1988) *Engineering Design: A Systematic Approach*, Springer, Berlin.

Sankey, H. (1995) 'The Problem of Rational Theory-choice', in: *Epistemologia* 18 (2): 299-312.

Sankey, H. (1999) 'Incommensurability:Aan Overview', in: *Divinatio: Studia Culturologica Series* 10: 1-13.

Sankey, H. (2002) 'Methodological Pluralism, Normative Naturalism and the Realist Aim of Science', in: Nola, R., and Sankey, H. (eds.), *After Popper, Kuhn and Feyerabend: Recent Issues in Scientific Method*, Kluwer, Dordrecht: 211-229.

Sasajima, M., Kitamura, Y., Ikeda, M., and Mizoguchi, R. (1996) 'A Representation Language for Behavior and Function: FBRL', in: *Expert Systems with Applications* 10 (3/4): 471-479.

Soler, L. (2008) 'Introduction', in: Soler, L., Sankey, H. and Hoyningen-Huene, P. (eds.), *Rethinking Scientific Change and Theory Comparison: Stabilities, Ruptures, Incommensurabilities?*, Springer, Dordrecht:1-17.

Stone, R.B., Wood , K.L., and Crawford, R.H. (2000) 'Using Quantitative Functional Models to Develop Product Architectures', in: *Design Studies* 21: 239-260.

Stone, R.B., McAdams, D.A., and Kayyalethekkel, V. (2004) 'A Product Architecture-Based Conceptual DFA Technique', in: *Design Studies* 25: 301-325.

Teller, P. (2001) 'Twilight of the Perfect Model Model', in: *Erkentniss* 55: 393-415.

Teller, P (2008) 'Of Course Idealizations are Incommensurable!', in: Soler, L., Sankey, H. and Hoyningen-Huene, P. (eds.), *Rethinking Scientific Change and Theory Comparison: Stabilities, Ruptures, Incommensurabilities?*, Springer, Dordrecht: 247-264.

Van Eck, D. (2009) 'On Relating Functional Modeling Approaches: Abstracting Functional Models from Behavioral Models', in: *Proceedings of The 17th International Conference on Engineering Design (ICED 09), August 24-27, 2009, Stanford, California, USA*: 2.89-2.100.

Van Eck, D. (2009/2010) 'On the Conversion of Functional Models: Bridging Differences Between Functional Taxonomies in the Modeling of User Actions', in: *Research in Engineering Design* 21 (2): 99-111. DOI 10.1007/s00163-009-0080-7. Published online 21 October 2009.

Van Eck, D. (2010a) 'Explaining and Relating Different Engineering Models of Functional Decomposition', in: *Proceedings Design Research Society (DRS) International Conference, 07-09 July 2010, Montréal, Canada*: 122.1-122.11.

Van Eck, D. (2010b) 'Supporting Design Knowledge Exchange by Converting Models of Functional Decomposition'. Delft University of Technology manuscript.

Van Eck, D. (2010c) 'Supporting Functional Knowledge Exchange Between Functional Taxonomies by Establishing Function-Behaviour Links', in: *Proceedings of the Eight International Symposium on Tools and Methods of Competitive Engineering-TMCE 2010, April 12-16, 2010, Ancona, Italy*: 1583-1595.

Vermaas, P.E. (2009) 'The Flexible Meaning of Function in Engineering', in: *Proceedings of the 17th International Conference on Engineering Design (ICED 09), August 24-27, 2009, Stanford, California, USA*: 2.113-2.124.

Worrall, J. (1988) 'The Value of a Fixed Methodology', in: *British Journal for the Philosophy of Science* 39: 263-275.

6 Supporting Design Knowledge Exchange by Converting Models of Functional Decomposition

This chapter is submitted as an article.

Abstract

Different meanings of the concept of function and different models of functional decomposition are used in engineering design. This diversity on the one hand has practical benefits to engineering, yet on the other it hampers the exchange of design knowledge across functional modeling frameworks. This paper presents a strategy for the exchange and re-use of design knowledge across functional frameworks. This strategy is intended to support the exchange and re-use of functional decompositions and knowledge on (configurations of) design solutions for the functions in these decompositions. This strategy does so by relating different functional decompositions via the conversion of behavior functional decompositions into behavior models. The paper argues that this conversion enables the exchange and re-use of knowledge on (configurations of) function-structure connections, which is not adequately supported by other conversion methods.

Keywords: design knowledge exchange; technical function; functional decomposition; model conversion

6.1. Introduction

The concept of technical function is crucial to engineering (see Chandrasekaran and Josephson: 2000; Stone and Wood: 2000; Chakrabarti and Bligh: 2001). Yet, this concept is used with different meanings in engineering design. As evidenced by a recent review of Erden et al. (2008), an impressive wealth of functional modeling approaches have been developed. In different approaches, different definitions of technical functions, representational formats for functions, and strategies for and models of the decomposition of functions into sub functions are proposed. This diversity in functional modeling has practical benefits to engineering (see Vermaas 2009; Van Eck: 2010, 2011). Yet, it also has disadvantages: it hampers the exchange of design knowledge across functional modeling frameworks (e.g., see Rosenman and Gero: 1999; Deng: 2002; Kitamura et al.: 2007). The different meanings of the concept of function, and the lack of clear relationships between them, is reported as a major factor that hampers the exchange of design knowledge across modeling frameworks (e.g., see Chandrasekaran and Josephson: 2000; Deng: 2002; Kitamura et al. 2007; Kitamura and Mizoguchi: 2009). At the same time, the exchange and re-use of design knowledge is recognized as important to engineering designing (e.g., see Stone and Wood: 2000, Szykman et al.: 2001; Otto and Wood: 2001; Deng: 2002; Kitamura et al.: 2007; Kitamura et al.: 2009).

The aim of this paper is to develop a strategy for the exchange and re-use of design knowledge across functional modeling approaches. By design knowledge we here mean functional decomposition models of technical artifacts, i.e., graphical representations of organized sets of (sub) functions, and (configurations of) components/design solutions for the functions in these models.

A methodology developed by Kitamura et al. (2007, 2008) and Ookubo et al. (2007) specifically aims to establish such knowledge exchange across functional modeling frameworks. It does so by converting models of functional decomposition. This conversion methodology has been demonstrated in terms of conversions of models described in terms of the Functional Basis (FB) taxonomy of Stone and Wood (2000) into models described in terms of the Functional Concept Ontology (FCO) taxonomy of Kitamura et al. (2005/6). Previous work has shown that this methodology encounters problems in the translation of user functions across approaches and presented alternative steps to support such translations (see Van Eck: 2009a). Facilitating the translation of user functions is, however, but one research challenge that model conversions face. Another challenge concerns the transposition of the specific *organizational* features of functional decomposition models from one model to another. For instance, transposing the input-output connections between functions in term of material, energy and signal flows from one model to another. Addressing this challenge is announced as a future research step, both in the works of Ookubo et al. (2007) and in Van Eck (2009a). This paper addresses this issue. As will be argued, the transposition of the organizational features of functional decomposition models is crucial to the exchange and re-use of knowledge on (configurations of) design solutions for the functions in models of functional decomposition (as laid down in design repositories). The here proposed strategy for the exchange and re-use of design knowledge is build in terms of an extension of the conversion methodology. The paper shows that conversion steps in the conversion methodology, valuable though it is, will lead to the *removal* of organizational features of functional decomposition models. Thereby, the exchange and re-use of knowledge on (configurations) of function-structure connections across functional modeling frameworks are not adequately supported. By means of this analysis, alternative conversion steps are proposed that avoid this removal. These steps support the exchange and re-use of knowledge on (configurations of) function-structure connections.

Key to this analysis is that the conversion methodology, in the case of FB-FCO model conversions, collapses distinctions between different meanings of the concept of function and different models of functional decomposition. This clarification both highlights why organizational features of models of functional decomposition are removed and gives means to bypass this problem. The newly proposed conversion step to address this problem consists of the conversion of functional decomposition models of behavior functions into models of (organized sets of) physical behaviors. Whereas the conversion methodology solely converts functional decomposition models into one another, the strategy proposed here both advances conversions between functional decompositions as well as conversions of functional decomposition models into behavior models. The specific type of conversion carried out depends on the type(s) of functional decomposition model(s) that are advanced in the functional modelling frameworks between which design knowledge is to be transferred.

Both the conversion methodology and the proposed alternative are demonstrated in terms of conversions of FB models of functional decomposition into FCO models. The proposed alternative is in addition applied to other models of functional decomposition. This demonstration will be given in terms of empirical models as advanced in the Functional Basis (FB) approach (see Stone and Wood: 2000), the Functional Concept Ontology (FCO) approach (see Kitamura and Mizoguchi: 2003, 2004) and the Dual Stage (DS) approach (see Deng, Tor, and Britton: 2000). Demonstration of the conversion steps in the conversion methodology and the proposed alternative by means of the same case studies allows for a clear comparison and lends support to the proposal developed here. The latter supports the exchange and re-use of function-structure connections across modeling frameworks in systematic fashion.

The analysis of different models of functional decomposition, on which the proposed conversions are based, serves a second aim of clarifying different types of engineering models and their relationships (cf. Kitamura et al.: 2009). The paper also briefly comments on specific benefits of specific models, thus validating the use of different models in engineering design and research into conversions between them.

The paper is organized as follows. It starts in the second section with a brief survey of the concept of function, functional decompositions and briefly discusses the advantages of specific models of functional decomposition. The third section describes functional decomposition models advanced in the FB, FCO, and DS approaches The fourth section reviews the conversion methodology and its application to FB-FCO model conversions. Section five presents the alternative conversion strategy. The strategy is demonstrated in terms of FB-FCO model conversions as well as conversions of other models of functional decomposition. Section six gives a general discussion, positioning model conversions to related research and commenting on future research opportunities. Section seven concludes the paper.

6.2. Survey: technical function and functional decomposition

6.2.1. Different meanings of the concept of function

The concept of function is used with different meanings in the engineering domain (see Chittaro and Kumar: 1998; Chandrasekaran and Josephson: 2000; Far and Elamy: 2005; Erden et al.: 2008; Van Eck: 2009b; Vermaas: 2009). This paper considers three meanings (cf. Vermaas: 2009):[1]

- Behavior function: desired behavior of a device
- Effect function: desired effect of behavior of a device
- Purpose function: purpose for which a device is designed

Behavior functions are advanced in, for instance, the Systematic approach of Pahl and Beitz (1988), the FB approach of Stone and Wood (2000), the Func-

1 I do not consider user functions – intentional behaviors carried out by an agent using a device – in this paper. For an in depth analysis of such functions, see (Van Eck: 2009a).

tional Reasoning approach of Chakrabarti and Bligh (2001) and the Reverse Engineering and Redesign approach of Otto and Wood (2001). In these approaches, functions are modeled as operations-on-material, energy or signal flows. These descriptions refer to physical behaviors since they represent conversions of matter and/or energy in which the input quantity matches the output quantity, meeting physical conservation laws (cf. Otto and Wood: 2001). A description of a behavior function is, for instance, 'converting electricity into torque and heat' of an electric screwdriver (see Stone and Wood: 2000) in which the energy of the electricity equals the sum of the energies of heat and torque.

Effect functions are advanced in, for instance, the Multi Level Flow approach (see Lind: 1994), the FCO approach, the Dual Stage approach (see Deng: 2002) and the Functional Interpretation Language approach (see Price: 1998; Bell, Snooke and Price: 2007). A function example may be, say, 'producing torque'. This description also refers to (features of) behavior but does not meet conservation laws, referring only to the desired effects of behavior.[2]

Purpose functions are advanced in, for instance, the Function-Behavior-Structure approach of Gero (1990), the framework of Chakrabarti (1998) and the DS approach (in which both effect functions and purpose functions are advanced). A function example may be, say, 'having a rotational force down a shaft'. This description refers to a state of affairs in the world, intended by a designer.

6.2.2. Functional decomposition models

These three notions of function are also described in models of functional decomposition, i.e., graphical representations of organized sets of functions. The paper considers three different models of functional decomposition (*fd*):

- Functional decomposition as a model of an organized set of behavior functions (behavior *fd*)
- Functional decomposition as a model of an organized set of effect functions (effect *fd*)

[2] In these approaches the concept of behavior is introduced alongside the concept of function. The behavior concept takes care of physical laws in these approaches.

- Functional decomposition as a model of an organized set of purpose functions (purpose *fd*)

Such models are put to a variety of uses. They are, among others, used in the conceptual phase of engineering designing to specify the desired functions of some product-to-be and to facilitate the selection of design solutions for those functions (see Stone and Wood: 2000; Chakrabarti and Bligh: 2001); in the reverse engineering of existing products (see Otto and Wood: 2001); and they are laid down in engineering knowledge bases for archival and communication purposes (see Kitamura et al.: 2005/6). One can distinguish strategies in which models of functional decompositions are developed in a solution-neutral fashion from ones in which known technical solutions for sub functions are incorporated from the outset (see Van Eck et al.: 2007). It has been argued that depending on the specifics of such design knowledge employment, particular models are chosen and suited to achieve particular objectives (see Van Eck: 2010). Other work further expands this analysis, demonstrating that behavior, effect, and purpose *fd*s each have specific benefits (see Van Eck: 2011). For instance, if the constraint applies that an *fd* should define a configuration of design solutions (for the functions in the *fd*) in which all design solutions are compatible with one another, behavior *fd*s are more suited than effect and purpose *fd*s.[3] If, on the other hand, the constraint applies that the realization (by design solutions) of the functions in an *fd* should be conceivable as independent from the realization of other functions, purpose *fd*s are better suited than behavior and effect *fd*s (see Van Eck: 2011).

Given that particular *fd*s have specific benefits (rather than one *fd* being clearly superior to other ones) efforts to clarify their relationships and investigate the possibilities to translate them are grounded. The next section provides an illustration of behavior, effect and purpose *fd*s. Subsequent sections discuss the conversion of such *fd*s.

---

3 Of course, the constraint that all design solutions are compatible with one another (and, hence, all functions realized) is a crucial constraint in all functional modeling frameworks. Modeling frameworks differ, however, in which design phase this constraint must be satisfied. In some approaches, *fd*s should satisfy this constraint (see Chakrabarti and Bligh: 2001) whereas in others it should be satisfied in later design phases (see Deng: 2002). Thus only in some approaches is it a constraint that applies to *fd*s.

6.3. Approaches toward functional decomposition

6.3.1. The Functional Basis approach

The Functional Basis (FB) approach, formulated by Stone and Wood (2000) is an approach to functional modeling that is aimed at creating a common and consistent functional design language, dubbed a functional basis. This language allows designers to model overall product functions as sets of interconnected sub functions. The FB approach is focused on especially the electromechanical and mechanical domains. The approach is presented as supporting the archiving, comparison, and communication of functional descriptions of existing products, as well as the engineering designing of new products. Since the approach was proposed it has been developed further. It is used to build a web-based repository in which *fd*s of existing products are archived, as well as components counting as design solutions for the sub functions that are part of these *fd*s. The function and flow information of components archived in this repository has been employed by Bryant et al. (2007) in building a function-based component ontology. In this ontology product components are classified based on their most commonly ascribed sub functions as archived in the repository.

In the FB approach, an overall product function refers to a general input/output relationship defined by the overall task of the product. This overall product function is described in a verb-object form and represented by a black-boxed operation on flows of materials, energies and signals. A sub function, describing a part of the product's overall task, is also described in a verb-object form but represented by a well-defined basic operation on a well-defined basic flow of materials, energies or signals. The black-boxed operations on general flows representing product functions are derived from customer needs, and the basic operations and basic flows representing sub functions are laid down in common and limited libraries that span the functional design space. These libraries are called a *functional basis*.

In 2002, the FB approach was reconciled with an approach developed by Szykman et al. (1999) in collaboration with Julie Hirtz, Daniel McAdams, and Simon Szykman (2002), and coined Reconciled Functional Basis.

To support engineering designing, Stone and Wood (2000) present a three-step methodology to develop *fd*s. The methodology starts with describing a product function in a verb-object form, represented by a black-boxed operation on flows of materials, energies and signals. A chain of operations-on-flows is then specified, called a function chain, for each black box input flow, which transform that flow step-by-step into an output flow. These operations-on-flows are to be selected from the FB libraries. Finally, these temporally ordered function chains are aggregated into a single *fd* of a product.

An FB *fd* of a hand-held stapler is shown in Figure 1, adapted from Stone et al. (2004). This model consists of temporally ordered chains of sub functions that transform the material input flows of "hand", "staples" and "sheet", and the energy input flow of "human force", step by step into output flows. This model is a behavior *fd*, organized such that the output flows of preceding behavior functions constitute the input flows of succeeding behavior functions.

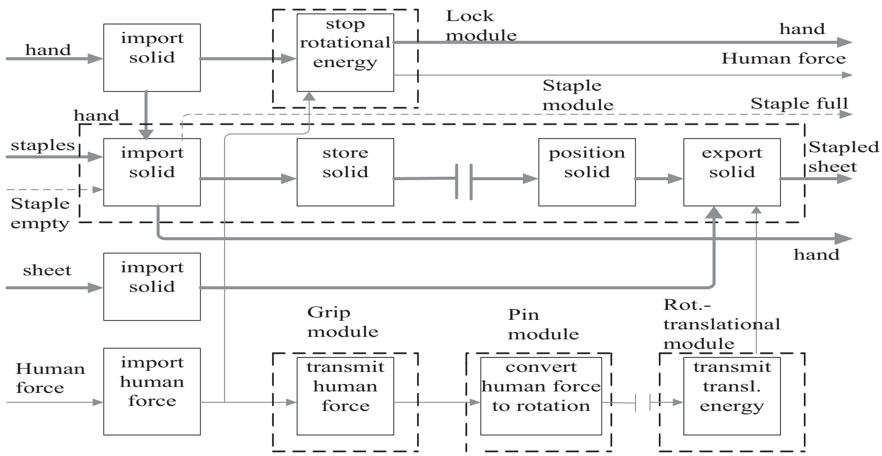

Figure 1: (Functional Basis) behavior *fd* of a stapler

6.3.2. The Functional Concept Ontology approach

The Functional Concept Ontology (FCO) approach developed by Kitamura and Mizoguchi (2003, 2004) and refined by Kitamura et al. (2005/6), is an approach to functional modeling that is aimed at facilitating the sharing of engineering functional knowledge. In this approach, in order to facilitate knowledge exchange, a set of modeling guidelines and a functional modeling language are developed to assist the systematic and reusable description of *fd*s of devices. The approach supports various tasks. It is employed in building an ontology for functions, in which their underlying behaviors and structures are archived as well, and in developing an automated design support system (see Kitamura and Mizoguchi: 2003).

In the FCO approach, both behavioral models and *fd*s of devices are developed. Behaviors of devices and their components are defined as input-output relations between operand states. Operands refer to energy, fluid, material, motion, force or information. Behaviors are represented as input-output state changes of properties of operands. Both overall functions and sub functions of devices are defined as roles played by behaviors, intended by designers or by users. Functions and sub functions are represented in terms of verb-operand pairs. The functional modeling language used in this approach consists of a generic set of verbs. These verbs are called *functional concepts* (see Kitamura et al.: 2005/6).

In an *fd* a set of sub functions is described that realize an overall function, and a specification is given of the technical principles by means of which the sub functions achieve the overall function. These specifications are referred to as "way of achievement", referring to knowledge on structures and the behaviors they exercise (see Kitamura et al.: 2005/6).

An FCO *fd* of a stapler is shown in Figure 2, adapted from Ookubo et al. (2007). This model consists of the overall function of the stapler, and sub functions of the modules and components of the stapler. Ways of achievement are shown in the model, specifying how the component functions realize the module functions, and how the module functions realize the overall function. The module function "combine sheets and staples", for instance, contributes to the realization of the overall function "combine sheets" by an "intermediate way" that represents the combining of paper sheets via staples acting as intermediates between the sheets. This model is an effect *fd*, in which the effect functions are grouped together (organized) in terms of their ways of achievement.

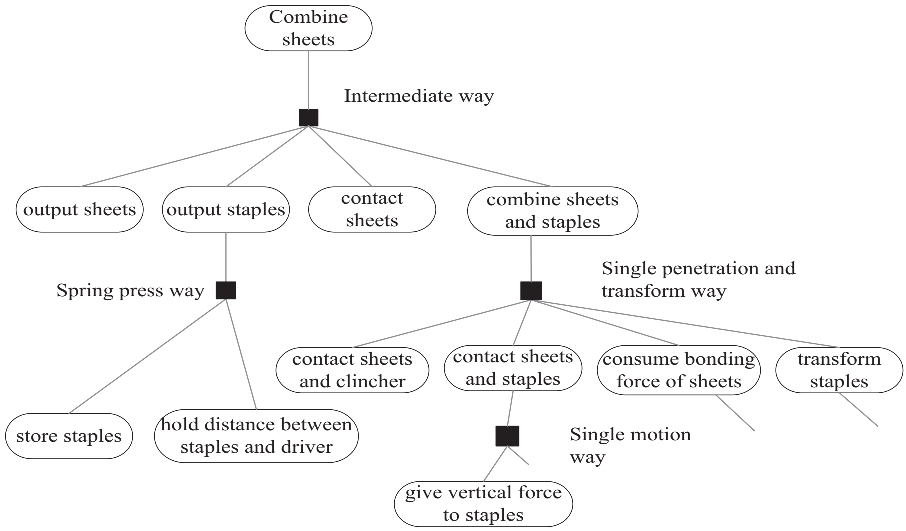

Figure 2: (Functional Concept Ontology) effect *fd* of a stapler

6.3.3. The Dual Stage approach

The Dual Stage (DS) approach, developed by Deng, Tor, and Britton (2000a, 2000b, 2002), is an approach to functional modeling that is aimed at supporting the engineering designing of products in the mechanical domain. The approach has also been used to build functional knowledge bases for automated design support systems (see Zhang et al.: 2001).

In this approach two types of functions are defined: purpose functions and action functions (see Deng: 2002). A purpose function refers to a designer's intention or purpose of a design. An action function is defined as an abstraction of intended behavior. Both types of function are represented by verb-noun descriptions. In addition to the concept of function, this approach also employs the concepts of behavior, structure, and working environment. The latter concept is used to refer to those interactions between a device and its environment that are required in order to let the device implement its desired functions.

To support engineering designing, Deng, Tor, and Britton (2000a, 2000b, 2002) present a knowledge base-assisted method to develop *fd*s of an overall purpose function. First, an overall purpose function is decomposed into purpose sub functions, using a function-library in which existing functional decomposition-design knowledge is stored. This library archives descriptions of purpose functions that have "pointers" added to them, linking them to sub functions and to functions of which they are a functional element. An overall purpose function is decomposed into those sub functions to which it is linked in the library. Then, these purpose sub functions are mapped onto structures using a physical structure-library, in which descriptions of commonly used structures are archived. The purpose sub functions stored in the function-library also have pointers to the structures housed in the physical structure-library that are suitable to implement them, thus supporting function-structure mapping. These steps constitute the first stage of the DS-modeling framework.

In this first stage, *fd*s are developed in which a set of purpose functions is described that realize an overall purpose function. In addition, "functional routes" are described which refer to technical knowledge on structures and the behaviors that they exercise by means of which sub functions achieve a function (see Deng: 2002). When functions from the function-library do not have pointers to physical structures, a physical phenomena-library is in a second stage searched to retrieve behavioral effects – action functions – that are considered suitable to achieve an unmapped purpose sub function. This physical phenomena library archives structures, behaviors and behavioral effects. These behavioral effects have pointers to physical structures, thus supporting function-structure mapping.

A DS *fd* of a terminal insertion device is shown in Figure 3, adapted from Deng (2002). This model consists of the overall function of the terminal insertion device and sub functions that achieve this overall function. How they achieve the overall function is specified by a "gang insertion"-functional route, which refers to the notion that an entire row (gang) of terminals is inserted with each insertion cycle. This model is a purpose *fd*, in which the purpose functions are grouped together (organized) in terms of their functional route (cf. Deng 2002).

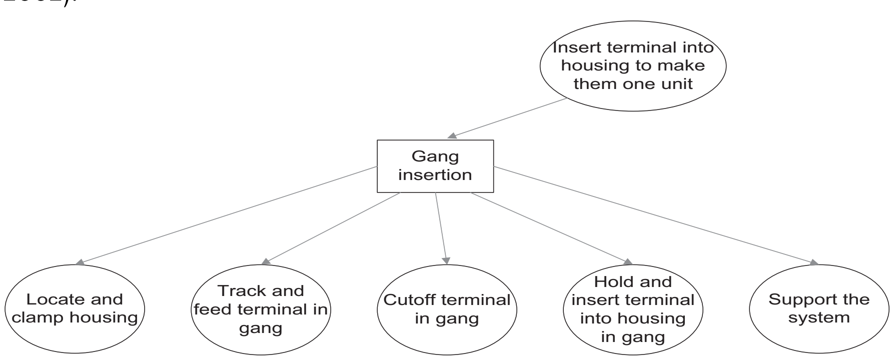

Figure 3: (Dual Stage) purpose *fd* of a terminal insertion device

Each of these *fd*s represents specific design knowledge of artifacts. For instance, whereas behavior *fd*s represent matter and energy conversions carried out by artifacts, purpose *fd*s represent the purposes for which artifacts are used. The conversion methodology of Kitamura et al. (2007) and Ookubo et al. (2007) aims to exchange such knowledge across functional modeling frameworks by converting *fd*s between them.

6.4. The conversion methodology

6.4.1. Function translations

In the conversion methodology, *fd*s are converted between modeling frameworks by carrying out two steps. In a first step, the function terms of a to-be-converted *fd* are translated into function terms that will be included in the converted *fd*. These function translations are developed by means of a “reference ontology” for functions (see Kitamura et al.: 2007; Ookubo et al.: 2007). In this reference ontology types or categories of functions are defined based on the functional conceptualizations of the FCO approach. The FCO concepts of device, behavior, function, and operand are further specified into subtypes called “descriptors of functions”, on the basis of which types/categories of functions are specified (see Kitamura et al.: 2007). These function categories are employed to identify the specific meaning(s) of functions in *fd*s and based on these identifications to translate functions from to-be-converted *fd*s into functions that will be included in converted *fd*s.

Different sorts of function translations can be carried out, depending on how function terms in to-be-converted *fd*s are classified. Function terms that can be translated into ones belonging to the same function category are presented as straightforward, since the same meaning is ascribed to these function terms. Such translations are called “within category” mappings. When the to-be-converted *fd* includes function terms that are classified in a certain function category and the modeling framework in terms of which the converted *fd* will be specified lacks function terms belonging to that same function category, the translation of these function terms of the to-be-converted *fd* involves more complex procedures. Such function terms (and their meaning) are namely part of one modeling framework, but not part of the other framework. These more complex translations are called “between category” mappings.

6.4.2. Modifying organizational features of functional decomposition models

In a second step, conceptual differences are explicated between the organizational features of to-be-converted *fd*s and *fd*s of the (other) modeling framework in terms of which the converted *fd* will be specified. That is, an *fd* of the same device as the to-be-converted *fd*, but which is constructed based on modeling criteria of the other framework, is used as a comparison *fd* to explicate differences between *fd*s of different modeling frameworks. Those organizational features of to-be-converted *fd*s that differ from (comparison) *fd*s of the other modeling framework in terms of which the converted *fd* will be specified, are modified such that the converted *fd* (increasingly) has the same organizational features as these comparison *fd*s.

After these two steps, an *fd* is converted from one functional framework to another.

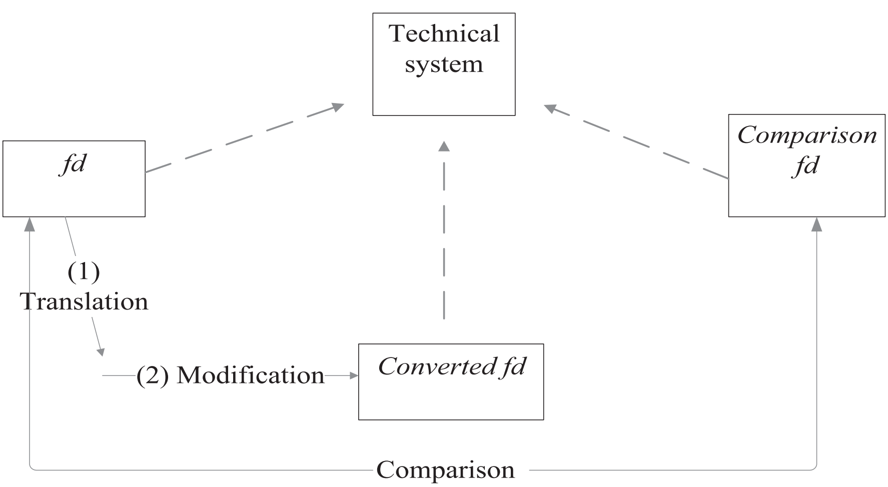

Figure 4: The Conversion Methodology

6.4.3. Converting functional decompositions between the Functional Basis and the Functional Concept Ontology

Ookubo et al. (2007) demonstrate their method by a conversion of an FB *fd* of a stapler represented in terms of the FB approach into a (converted) *fd* represented in terms of the FCO approach. They also use a comparison FCO *fd* of a stapler in this conversion to identify conceptual differences between the organizational features of FB *fd*s and FCO *fd*s. The FB *fd*, which Ookubo et al. (2007) adapt from Stone et al. (2004), is shown in Figure 1. The comparison FCO *fd* is shown in Figure 2 and the converted FB *fd* is shown in Figure 5.[4]

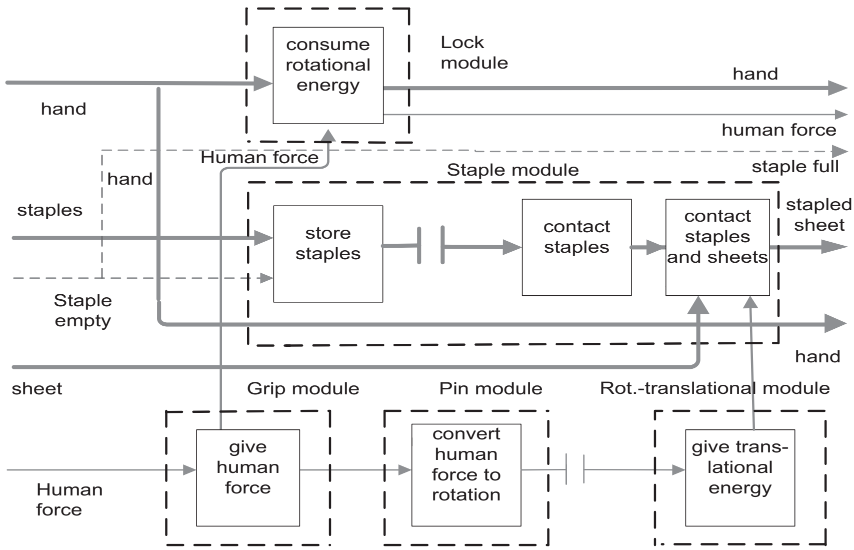

Figure 5: Converted Functional Basis *fd* of a stapler

In this conversion of the FB *fd* of the stapler, Ookubo et al. (2007) translate functions of this *fd* both by "within category" mappings and by "between category" mappings. Most FB function terms and all FCO function terms are classified in the reference ontology in the "flowing object" function category (see Kitamura et al.: 2007). Flowing object functions refer to a specific type of behavior, to wit: temporal changes in attributes of a physical entity, such as matter and energy flows or operands, within a device's system boundary. A role – captured by a flowing object-function description – is ascribed to these behaviors in a teleological context (see Kitamura et al.: 2007; Ookubo et al.: 2007). Since most

4 Figure 1 gives the same adaptation of the FB *fd* as Ookubo et al. (2007) give. This adaptation consists in excluding several operations-on-flows that are described in the original FB *fd*. The vertical lines intersecting the "human force" flow and the "staples" flow represent this exclusion.

function terms in the FB and FCO approaches are classified as flowing object functions, the same meaning is ascribed to most FB and FCO functions, i.e., roles of systemic behaviors. These FB functions are translated by "within category" mappings. An example of such a within category mapping in this *fd* conversion is the FB function "transmit human force" (Figure 1) that is translated into the FCO function "give human force" (Figure 5).

Some of the functions in the FB *fd* of the stapler are classified in the reference ontology as "system interface functions". System interface functions represent temporal changes in attributes of a physical entity on a system boundary. The FB "import" and "export" function terms are classified as system interface functions. Since the FCO solely consists of "flowing object functions", these FB function terms are translated by means of a between category mapping: the FB "import" and "export" operations-on-flows are translated into FCO input and output operands (see Ookubo et al.: 2007). Examples of such between category translations are the "import solid (sheet)" and "export solid (stapled sheet)" functions of the FB *fd* (Figure 1) that are translated into FCO input and output operands of "sheet" and "stapled sheet" (see Figure 5).

After these translations, the FB *fd* (Figure 1) and the comparison FCO *fd* (Figure 2) are in a second step compared to identify conceptual differences between the organizational features of these *fd*s. Those organizational features of the to-be-converted FB *fd* that differ from the comparison FCO *fd* are modified such that the converted *fd* (increasingly) has the same organizational features as this comparison *fd*.

Six such conceptual differences are identified between the FB *fd* and the comparison FCO *fd* (see Ookubo et al.: 2007):

(1) In FCO *fd*s, overall functions are related to sub functions of modules, which are related to sub functions of components. FB *fd*s do not represent relationships between sub functions of modules and components.
(2) In FCO *fd*s, functions are not connected by operands, whereas functions are connected by input-output flows in FB *fd*s.
(3) In FCO *fd*s, ways of achievements are described, whereas these are not described in FB *fd*s.

(4) In FCO *fd*s, changes in distance between physical objects – matter and energy flows/operands – are described, whereas these are not described in FB *fd*s.
(5) In FCO *fd*s, features of users are not described, whereas features of users are described by human material flows in FB *fd*s.
(6) In FCO *fd*s, material and energy operands may be grouped together in descriptions of functions, whereas material and energy flows are separated in FB *fds*.

Of these six differences, two have currently been modified in the conversion process: the difference in distance changes between physical objects (4) and the difference in user features (5). Addressing these other differences is announced as future work (see Ookubo et al.: 2007). The converted FB *fd* in Figure 5 thus currently is the endpoint of the conversion process; it is the result of the translation of functions in the first step, and from the modifications in the second step that address differences in representing distance changes between flows/operands, and differences in representing (parts of) users of devices.

Modifying by adding and removing functional information

The difference between the FB *fd* (Figure 1) and the comparison FCO *fd* (Figure 2) concerning distance changes between flows/operands is addressed by *adding* an FCO function from the comparison *fd* to the converted *fd* (Figure 5). In the FB *fd*, the "staple" and "sheet" flows enter the stapler as separate flows and exit as the combined flow "stapled sheet". The actual combining of these flows, referring to a change in distance between flows, is not represented in this FB *fd*. In contrast, this combining is explicitly represented in the comparison FCO *fd* by the function "contact staples and sheets". This difference is addressed by adding this FCO function to the converted *fd*.

The difference between FB and FCO *fd*s regarding the representation of (parts of) users of devices is addressed by *removing* FB functions in the model conversion. In the FB *fd*, parts of users are represented in terms of flows of human materials such as "hand". In contrast, parts of users are not represented in FCO *fd*s of devices. Ookubo et al. (2007) address this difference by removing FB functions that have input or output flows of human materials. The FB function "import solid (hand)", for instance, is removed and not represented in the converted *fd*.

6.4.4. Information loss

These modifications are problematic. Previous work argues that the removal of human material flows leads to considerable information loss, particularly in those cases in which whole function chains in to-be-converted *fd*s are represented in terms of such flows (see Van Eck: 2009a). As a result of their removal in the conversion process, converted *fd*s then do not represent this functional information. Other modification steps to-be-discussed below entail a different sort of information loss. This type of information loss leads to problems that more directly hamper design knowledge re-use in engineering designing.

Given that the second modification step aims to modify organizational features of to-be-converted *fd*s that differ from (comparison) FCO *fd*s such that converted *fd*s (increasingly) have the same organizational features as (comparison) FCO *fd*s, similar removal steps must be taken to address the other conceptual differences identified between FB and FCO *fd*s. For instance, the input-output connectivity of FB functions in terms of flows (2) and the separation of material and energy flows (6) in FB *fds*, are *incompatible* with FCO *fd*s. Modification of these differences thus will entail the removal of these organizational features of FB *fd*s. However, such removal steps result in significant information loss: by removing the input-output flow connections between FB functions and by removing the separation of material and energy flows, key organizational features of FB *fd*s are removed in a conversion.

This removal has a substantial consequence for the prospects of establishing design knowledge transfer between functional modeling approaches. Functional decompositions as archived in the FB design repository also define or are linked to spatial configurations, or what we may call *structural organizations*, of the components counting as design solutions for the functions in *fd*s. That is, *fd*s in the design repository provide information on how the components making up a device are configured. For instance, the input-output connectivity (or temporal ordering) of functions by means of flows in *fd*s provides information on the spatial connections between components that have these functions. Such linkages between *fd*s and spatial configurations of components can now also be automatically generated by means of a design tool coined the Concept Generator (see Bryant et al.: 2006). This tool generates, by means of several algorithms, *fd*s of overall product functions as well as configurations of components solving the functions in these *fd*s.

Yet, when information on the input-output connectivity of FB functions in terms of flows is removed in *fd* conversions, information on the spatial configuration of the components (solving these functions) of devices is lost as well. In effect, this information cannot be shared and re-used between functional modeling frameworks. Knowledge on function-component connections and their configurations cannot be accessed and used for design purposes. For instance, in the stapler case, information on spatial connections between spring, spring mount, spring holder and casing, which are design solutions linked to interconnected functions of a stapler *fd*, would be lost (cf. Stone et al.: 2004).

The next section presents a strategy to solve this problem.

6.5. Converting behavior functional decompositions into behaviorr models

6.5.1. Conceptual background

A solution to convert input-output flow connections between functions and the separation of material and energy flows in conversions of FB *fd*s, and thus preserve information on the spatial configuration of the components of devices, comes in view when one considers that FB *fd*s are behavior *fd*s and FCO *fd*s are effect *fd*s (see sections 2 and 3).

Whereas FB-to-FCO conversions in the conversion methodology are carried out under the assumption that the concept of function in the FB approach is (to a large extent) the same as the concept of function in the FCO approach (see section 4.3), the analysis given in sections 2 and 3 instead shows that the meaning of the concept of FB function differs from the meaning of the concept of FCO function: FB functions correspond to desired physical behaviors, whereas FCO functions – roles of behaviors – only refer to particular features of physical behaviors, namely their desired effects. Similarly, these approaches advance different models of functional decomposition. FB *fd*s are behavior *fd*s, in which the functions are organized such that the output flows of preceding behavior functions constitute the input flows of succeeding behavior functions. FCO *fd*s, instead, are effect *fd*s, in which effect functions are grouped together (organized) in terms of their ways of achievement. Thus, the conceptual differences (2) and (6) between FB *fd*s (e.g., Figure 1) and FCO *fd*s (e.g., Figure 2) emerge as differences between *FB behavior fd*s *and FCO effect fd*s.

Yet, whereas these approaches advance different models of functional decomposition, FB *fd*s are very similar to FCO models of organized sets of physical behaviors. Let us refer to these latter models as "behavior models". In the FCO approach, instead of functions, behaviors of components are connected by input-output operands of material, energy, and signal in models of (organized sets of) behaviors (see Ookubo et al.: 2007). Similarly, material and energy operands are separated in these behavioral models (see Sasajima et al.: 1996). Thirdly, behavioral descriptions in FCO behavior models meet physical conservation laws just as functional descriptions in FB *fd*s do.[5] Thus, FB *fd*s correspond to FCO behavior models rather than to FCO *fd*s.

This correspondence between FB *fd*s and FCO behavior models gives means to convert input-output flow connections between functions as well as the separation of material and energy flows.

6.5.2. An alternative proposal: conversions of behavior functional decompositions into behavior models

As mentioned, the converted FB *fd* in Figure 5 currently is the endpoint of the conversion process by the conversion methodology. The modification of the organizational features of the input-output connectivity of FB functions in terms of flows (2) and the separation of material and energy flows (6) still needs to be addressed. Moreover, as mentioned, modification of these differences will entail the removal of these organizational features of FB *fd*s. The resultant information loss hampers the exchange and re-use of design knowledge between functional modeling frameworks on function-component connections and the spatial configuration of components of devices.

However, by identifying FB *fd*s with FCO behavior models modifications that entail the removal of these organizational features can be avoided. Rather than converting FB *fd*s into FCO *fd*s one can convert FB *fd*s into FCO behavioral

---

5 FCO *fd*s do not take physical law into account. This makes perfect sense, since in the FCO approach the concept of behavior takes care of physical laws. An example given by Sasajima et al. (1996) illustrates the point clearly. They describe the behavior of a particular device as dividing an input saline solution into pure salt and a saline solution. The function ascribed to this behavior is specified as "producing salt".

models. Such conversions are able to preserve information on the input-output connectivity of FB functions in terms of flows (2) and the separation of material and energy flows (6), since FCO behavioral models also exhibit these organizational features. In effect, by preserving this information one can also transfer knowledge on FB function-component connections and component configurations. FCO behavioral models provide access to this information: information on FB components (and their configuration) counting as design solutions to FB functions can be re-used as FCO behavior-component connections.

Interestingly, the current endpoint of the FB-FCO conversion in the conversion methodology (the model in Figure 5) can be interpreted as exemplifying such a conversion of an FB *fd* into and FCO behavioral model. Since differences in the input-output connectivity of FB functions in terms of flows and the separation of material and energy flows are not modified at this point, functions in the converted model in figure 5 are still connected by input-output flows and material and energy flows are still separated. The converted FB model thus exhibits the features characteristic of FB behavior *fd*s and FCO behavior models. Interpreted in this fashion, this conversion can be seen to support the transfer of behavioral descriptions.

Such conversions of behavior *fd*s into behavioral models are a first crucial step in the transfer of design knowledge on functions and function-component connections between modeling frameworks that advance different models of functional decomposition. Yet, in order to make this knowledge also available for the purpose of re-using function-structure connections (and their configurations), rather than behavior-structure connections, a second step needs to be taken. The different models of functional decomposition – here FB *fd*s and FCO *fd*s – need to be related. By relating FB *fd*s and FCO *fd*s – in the case of FB-FCO conversions – FB components (and their configurations) counting as design solutions to the functions in FB *fd*s become available as possible design solutions to FCO functions.

6.5.3. A two step conversion strategy: converting behavior functional decompositions and abstracting effect functional decompositions

This paper proposes the following two steps to convert behavior *fd*s and relate them to effect *fd*s. In a first step, (FB) behavior functions are translated into (FCO) behaviors. Flow connections between behavior functions and the separation of material and energy flows are converted as well. This step establishes a conversion of an (FB) behavior *fd* into an (FCO) behavior model, in which model features of input-output flow connectivity and the separation of material and energy flows are preserved. This step supports the exchange of behavioral descriptions.

In a second step, the relevant features of the behaviors represented in the converted model are abstracted and incorporated into effect function descriptions. These effect function descriptions are used to develop an (FCO) effect *fd*. This abstraction step links converted (FB) behavior *fd*s to (FCO-based) effect *fd*s. These abstraction steps follow the guidelines for the abstraction of functions from behaviors of the specific modeling approach in terms of which the converted and abstracted models are constructed. Typically, archived design knowledge on function-behavior connections is employed to make these abstractions (e.g., see Deng: 2002; Kitamura and Mizoguchi: 2004). This second step supports the transfer of functional descriptions. By relating (FB) behavior functions to (FCO) effect functions design solutions to these behavior functions become available as possible design solutions for these effect functions. This two-step strategy is depicted in figure 6.

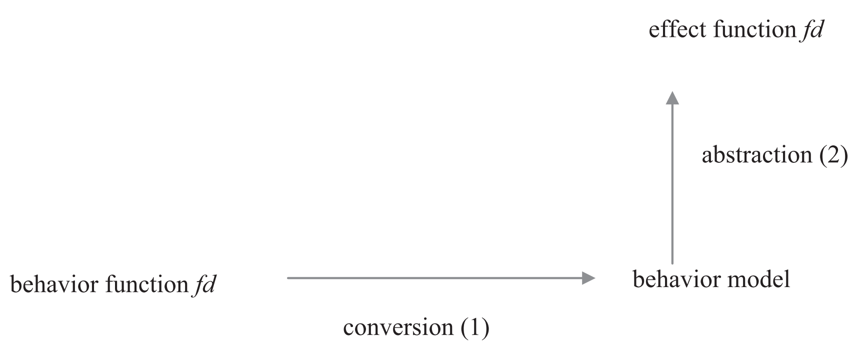

Figure 6: Relating a behavior function *fd* to an effect function *fd*

Applied to FB-FCO conversions, the results of this strategy are as follows. The models of the stapler in Figures 1, 5 and 7 illustrate this proposal. The FB model in Figure 1 represents the to-be-converted behavior *fd*, the model in Figure 5 represents its converted behavioral counterpart and the *fd* in Figure 7 represents an FCO-based effect *fd*, abstracted from the converted (behavioral) model. This abstraction step can be taken in terms of the modeling guidelines of the FCO, typically supported by employing archived design knowledge on function-behavior connections. The functions in the abstracted *fd* in figure 7 are represented according to their grouping in function chains and modules in the converted FB model (Figure 5). Using this module information of the converted FB model for the grouping of FCO functions in the *fd* in Figure 7 accords with the use of ways of achievements for the grouping of functions in FCO *fd*s. Ways of achievement refer to information about structures and the behaviors that they exercise. Likewise, FB modules refer to information about structures and FB functions to the behaviors of these structures. Module information is given at the nodes (cf. Figure 5). I use oval nodes in Figure 7 to distinguish module information from ways of achievement, which are represented in FCO *fd*s by squares (cf. Figure 2). This abstraction step retains FB module information, thus making design solutions to FB behavior functions available as possible design solutions to FCO effect functions.

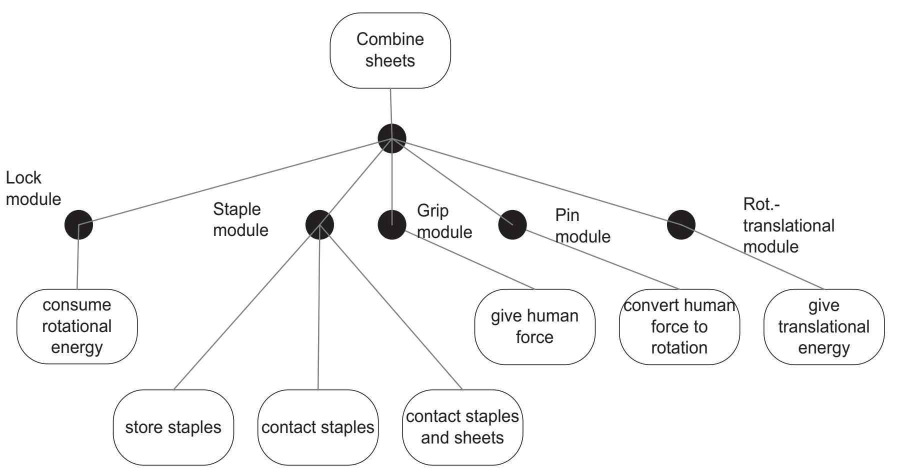

Fig 7: Abstracted FCO-based effect function *fd*

6.5.4. Converting behavior functional decompositions and abstracting purpose functional decompositions

One can also apply this strategy to the conversion of behavior *fd*s and subsequent abstraction of purpose *fd*s. This application is demonstrated in terms of an FB behavior *fd* (the FB model of the stapler of Figure 1 is the to-be-converted *fd)* and a DS purpose *fd*.

In the DS approach, operation-on-flow representations are taken to refer to physical behaviors (see Deng, Tor, and Britton: 2000a). In this approach, the physical behaviors considered most relevant in mechanical engineering design are the physical interactions between components of a device and between a device and its working environment. Material, energy, and signal flows are viewed as attributes of these interactions. One can, hence, identify the concept of FB behavior function with that of (attributes of) physical behavior in the DS approach. This identification gives means to relate FB behavior *fds* to DS purpose *fds* as follows. FB behavior *fd*s are in a first step converted into DS behavior models: FB functions are translated into DS behaviors, and the FB model features of flow connections between functions and the separation of flows are expressed in the converted (behavior) model as well. (The DS approach has the expressive power to do so. In physical phenomena libraries both behaviors and their attributes of material, energy, and signal are represented (see Deng: 2002). The verb-noun descriptions employed in the DS approach combined with these attribute representations thus give the possibility to represent operations-on-flows).

In a second step the relevant effects of the behaviors represented in the converted model are abstracted as effect functions. This abstraction step is done in term of the modeling guidelines of the DS, typically supported by employing archived design knowledge on function-behavior connections. In a third step, purpose functions are then abstracted from these effect functions. This abstraction step again can be taken by means of archived DS knowledge on connections between effect functions and purpose functions. These purpose function descriptions are used to develop a DS purpose *fd*. The purpose functions in this *fd* are grouped together (organized) based on the module information and grouping of functions in function chains in the (converted) FB model. Using this FB module information for grouping the purpose functions in the DS purpose *fd*, accords with the use of functional routes in the DS approach for grouping sub functions. As mentioned, functional routes in the DS approach refer to technical knowledge on structures and the behaviors that they exercise by means of which sub functions achieve a function (see Deng: 2002). Likewise, FB modules refer to information about structures and FB functions to the behaviors of these structures. This application is depicted in figure 8.

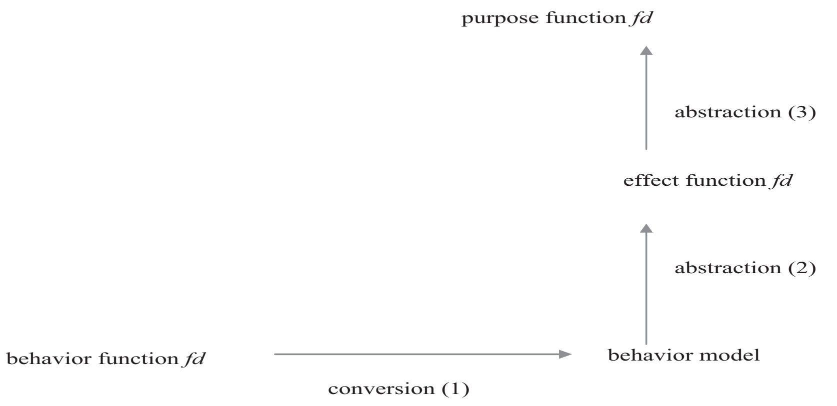

Figure 8: Relating a behavior *fd* to a purpose *fd*

The *fd* in Figure 9 represents the result of this application. Module information in this *fd* is given at the nodes. FB information on the grip, pin and rotational-translational modules is combined to specify the DS purpose function "transmit human energy and convert it into rotational energy". I use oval nodes (as in Figure 7) to distinguish module information from functional routes, which are represented in DS models by squares (for brevity I have skipped the step of constructing a converted (behavior) model, but see the converted FB model of the stapler (Fig 5) for the general idea)

The first conversion step supports the exchange of behavioral information and, by relating behavior *fd*s to purpose *fd*s in a second and third step, the transfer of functional descriptions is supported. By relating (FB) behavior functions to (DS) purpose functions, in which FB module information is retained, design solutions to these behavior functions become available as possible design solutions for these purpose functions.

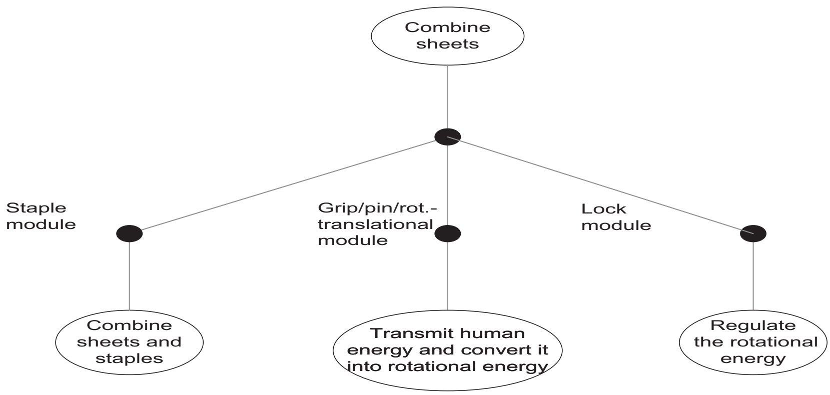

Figure 9: Abstracted DS-based purpose *fd*

6.5.5. Converting effect functional decompositions models and abstracting purpose functional decompositions

The steps discussed above also provide the means to relate effect *fd*s to purpose *fd*s. This application is demonstrated in terms of an FCO effect *fd* (the FCO *fd* of the stapler in Figure 2 is the to-be-converted model) and a DS purpose *fd*.

Both FCO functions and DS "action" functions are effect functions, and are represented by verb-noun descriptions. This identification gives means to relate FCO effect *fd*s to DS purpose *fd*s as follows. FCO effect *fd*s are in a first step converted into DS effect *fd*s: FCO effect functions are translated into DS effect functions, and the functions in the converted effect *fd* are grouped together (organized) in terms of the ways of achievement-information in the original FCO *fd*. Using this information for grouping the effect functions in the converted *fd*, accords with the use of functional routes in the DS approach for the grouping of functions. Both these concepts are employed to refer to technical knowledge on structures and the behaviors that they exercise by means of which sub functions achieve a function.

In a second step, purpose functions are then abstracted from the effect functions in the (converted) effect *fd*. This abstraction step can be taken by means of archived DS knowledge on connections between effect functions and purpose functions. These purpose function descriptions are used to develop a DS purpose *fd*. The purpose functions in this model are grouped together (organized) based on the ways of achievement-information in the converted effect *fd*. Information on the grouping of functions is thus transposed from the converted model to the purpose *fd*. This application is depicted in figure 10.

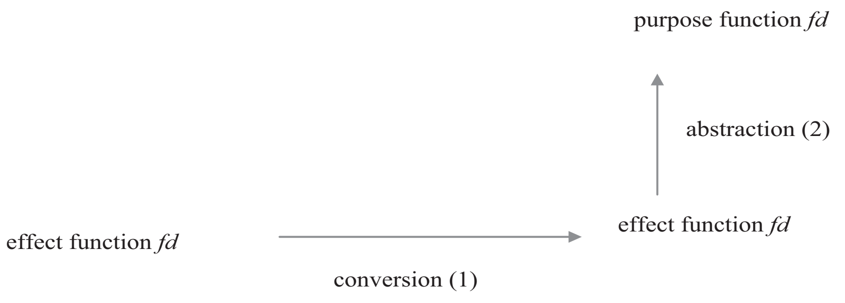

Figure 10: Relating an effect *fd* to a purpose *fd*

The result of this application is the DS-based purpose *fd* in Figure 11. (Ways of achievement-descriptions in FCO *fd*s are increasingly specified when moving from the level of module functions to the level of component functions. Functional routes in DS *fd*s are not elaborated in this fashion. In the abstracted DS purpose *fd* the ways of achievement-information is combined in specifying the purpose functions in this model). Ways of achievement-information is given at the nodes. I use oval nodes to distinguish ways of achievements from functional routes in the DS approach (although these concepts are similar), which are both represented by squares.

This strategy supports the exchange of (FCO) effect function information and links effect function descriptions to purpose function descriptions, facilitating cross- communication of these types of functional descriptions. The first conversion step supports the exchange of effect function information and, by relating effect *fd*s to purpose *fd*s in a second step, the transfer of functional descriptions is supported. By relating (FCO) effect functions to (DS) purpose functions, in which FCO ways of achievement-information is retained, design solutions to these effect functions become available as possible design solutions for these purpose functions. (That is, FCO effect *fd*s are linked to FCO models of behaviors of components. By relating FCO effect *fd*s to DS purpose *fd*s, those FCO components can be re-used as possible design solutions to DS purpose functions).

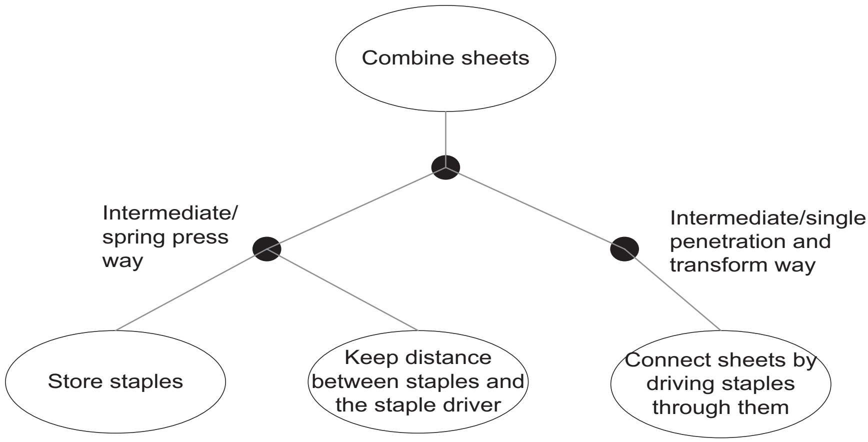

Figure 11: Abstracted DS-based purpose function *fd*

6.5.6. Two-way conversions

The proposed strategy can be applied both ways. This paper focused on one-way conversions. In opposite direction, we can also proceed by, for instance, taking the FCO *fd* of a stapler in figure 2 as starting point of a conversion process into an FB *fd*. Firstly, we identify the FCO behavior model related to the FCO effect *fd*. This step is easily established since FCO *fd*s are related to FCO behavior models (as archived in the FCO ontology). We can subsequently convert this FCO behavior model into an FB behavior *fd*. (Such an application is not demonstrated here due to space limitations). Thereby, design knowledge on FCO components (and their configurations) become available as possible design solutions to the FB functions in the (converted) FB behavior *fd*. The proposed key step of relating different models of functional decomposition via behavior models thus supports the exchange and re-use of knowledge on (configurations of) design solutions for the functions in models of functional decomposition back and forth.

This exchange of design knowledge back and forth is an advantage over the conversion methodology. When the conversion methodology would be applied to two-way conversions, critical information on organizational features of to-be-converted *fd*s, again, would be lost. By implication, transfer and re-use of knowledge on design solutions for functions is hampered. Consider applying the conversion methodology to the conversion of the FCO *fd* of the stapler in Figure

2 into an FB *fd*. Application of the conversion methodology entails that the FCO *fd* of the stapler in Figure 2 is compared with an FB *fd* of a stapler and that, based on this comparison, organizational features of the to-be-converted *fd* (the FCO *fd* of the stapler in Figure 2) that differ from the comparison FB *fd* are modified such that the converted *fd* (increasingly) has the same organizational features as this comparison FB *fd* (see section 4). In this case this implies, for instance, that ways of achievement-information is removed in the conversion process since ways of achievement are not represented in FB *fd*s. However, ways of achievement refer to knowledge on FCO behaviors and components that implement the functions in an FCO *fd* (which are more elaborately described in FCO behavior models). Yet, by removing ways of achievement-information in the conversion process, information on structures/components counting as design solutions for FCO functions is lost as well. Thereby, design knowledge on (configurations of) FCO function-component connections cannot be transferred to and re-used by the FB approach. The strategy proposed in this paper does support such knowledge exchange.

6.6. Discussion

Most recent research on the management of functional design knowledge is focused on the consistent description of specific notions of function and/or functional decomposition models (e.g., see Szykman et al.: 2001; Kitamura and Mizoguchi: 2004). Such descriptions, laid down in function ontologies, aim to support the consistent storage, retrieval and cross-communication of functional descriptions of artifacts. The use of such ontologies is reported successful for the above purposes (see Kitamura et al.: 2005/6). However, this success depends on the sharing of the *same* ontology by the relevant engineers and engineering teams involved. When functional conceptualizations are not shared, however, research challenges emerge for the communication of design knowledge across *different* functional frameworks. The conversion research presented in this paper is aimed to contribute to this kind of design knowledge management.

Research into design knowledge management across functional frameworks is still in a conceptual phase. The current state is about clarifying distinctions and relationships between different notions of function and models of functional decomposition (cf. Kitamura and Mizoguchi: 2009) and developing conceptual and manual tools for knowledge exchange (cf. Ookubo et al.: 2007; Kitamura et al.: 2008; Van Eck: 2009a). A next step in this research is the automation of conversion instructions across functional frameworks. Yet, taking this step first entails formalizing notions of function and models of functional decomposition in a logically precise way such that they can be implemented in automated tools. Such formalizations are only fairly recently being developed (cf. Borgo et al.: 2009). The research presented in this paper provides conceptual input for the formalization of functional decompositions and their relationships. This is a topic for future research.

A final comment concerns the generality of the proposed conversion strategy. The strategy is focussed on the exchange and re-use of known function-structure connections in engineering designing. Different modelling approaches propose different ways to connect functions to components and/or abstract functions from the behaviours of components. The proposed strategy explicitly develops such abstractions of functions from behaviours in terms of the modelling guidelines or available knowledge on function-behaviour connections of the specific approaches between which design knowledge is to be transferred. This feature of the strategy is motivated by the assumption that diverse techniques for the abstraction of functions from behaviours of components are likely to lead to diverse connections between functions and components. By keeping these different techniques aboard, the available engineering spectrum of function-component couplings can be accessed and re-used. Adopting a specific technique for the abstraction of functions from the behaviours of components may, instead, lead to a narrowing down of this spectrum.

By focusing on known function-structure connections the strategy is (currently) limited to engineering redesign. In some approaches innovative designing entails the development of *fd*s that are "form-independent", meaning that they are constructed without taking existing function-component couplings into account during the construction of such *fd*s (e,g., see Stone and Wood: 2000). This procedure is motivated by the idea that novel (configurations of) function-structure connections may be arrived at. It is an avenue for future research if and how "form independent" *fd*s can be converted into *fd*s that are expressed in terms of functional frameworks that take knowledge on existing function-structure connections into account from the outset in the construction of *fd*s.

6.7. Conclusions

This paper presented a strategy for the exchange and re-use of design knowledge across functional modelling frameworks. Specifically, this strategy is intended to support the exchange and re-use of functional decompositions and knowledge on (configurations of) design solutions for the functions in these decompositions. This strategy relates different functional decompositions by means of the conversion of behaviour functional decompositions into behaviour models. This conversion is novel when compared with other conversion methods. The paper argued that this novel conversion enables the exchange and re-use of knowledge on (configurations of) function-structure connections, which is not adequately supported by other conversion methods.

Acknowledgements

I would like to thank the guest editors, four anonymous reviewers, and Pieter Vermaas for their valuable comments. This research is supported by the Netherlands Organization of Scientific Research (NWO).

References

Bell, J., Snooke, N., and Price C. (2007) 'A Language for Functional Interpretation of Model Based Simulation', in: *Advanced Engineering Informatics* 21: 398-409.

Borgo, S., Carrara, M., Garbacz, P., and Vermaas, P.E. (2009) 'A Formal Ontological Perspective on the Behaviors and Functions of Technical Artifacts', in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 23: 3-21.

Bryant, C. R., McAdams, D. A., Stone, R. B., Kurtoglu, T., and Campbell, M. I. (2006) 'A Validation Study of an Automated Concept Generator Design Tool', in: *Proceedings of the 2006 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, DETC2006-99489: 1-12.

Bryant, C.R, Stone, R. B., Greer, J.L., McAdams, D.A., Kurtoglu, T., and Campbell, M.I. (2007) 'A Function-Based Component Ontology for Systems Design', in: *Proceedings of International Conference on Engineering Design (ICED 07), August 28-31, 2007, Paris, France*: 478.1-478.12.

Chakrabarti, A. (1998) 'Supporting Two Views of Function in Mechanical Designs', in: *Proceedings 15th National Conference on Artificial Intelligence, AAAI'98, July 26-30, 1998, Madison, Wisconsin, USA*.

Chakrabarti, A. and Bligh, T.P. (2001) 'A Scheme for Functional Reasoning in Conceptual Design', in: *Design Studies* 22: 493-517.

Chandrasekaran, B. and Josephson, J.R. (2000) 'Function in Device Representation', in: *Engineering with Computers* 16: 162-177.

Chittaro, L. and Kumar, A. N. (1998) 'Reasoning about Function and its Applications to Engineering', in: *Artificial Intelligence in Engineering* 12: 331-336.

Deng, Y.M. (2002) 'Function and Behavior Representation in Conceptual Mechanical Design', in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 16: 343-362.

Deng, Y.M., Tor, S.B., and Britton, G.A. (2000a) 'A Dual-Stage Functional Modelling Framework with Multi-Level Design Knowledge for Conceptual Mechanical Design', in: *Journal of Engineering Design* 11 (4): 347-375.

Deng, Y.M., Tor, S.B., and Britton G.A. (2000b) 'Abstracting and Exploring Functional Design Information for Conceptual Mechanical Product Design', in: *Engineering with Computers* 16: 36-52.

Erden, M.S., Komoto, H., Van Beek, T.J., D'Amelio, V., Echavarria, E, and Tomiyama, T. (2008), 'A Review of Function Modelling: Approaches and Applications', in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 22: 147-169.

Far, B.H. and Elamy, A.H. (2005) 'Functional Reasoning Theories: Problems and Perspectives', in: *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing* 19: 75-88.

Garbacz, P. (2006) 'A Formal Model of Functional Decomposition', in: *Proceedings of the 2006 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, September 10-13, 2006, Philadelphia, Pennsylvania, USA*. DETC2006-99097: 115-124.

Gero, J.S. (1990) 'Design Prototypes: A Knowledge Representation Schema for Design', in: *AI Magazin,* 11(4): 26-36.

Hirtz, J., Stone, R. B., McAdams, D. A., Szykman, S., and Wood, K. L. (2002) 'A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts', in: *Research in Engineering Design* 13: 65-82.

Kitamura, Y. and Mizoguchi, R. (2003) 'Ontology-Based Description of Functional Design Knowledge and its Use in a Functional Way Server', in: *Expert Systems with Applications* 24: 153-166.

Kitamura, Y. and Mizoguchi, R. (2004) 'Ontology-Based Systematization of Functional Knowledge', in: *Journal of Engineering Design* 15 (4): 327-351.

Kitamura, Y. and Mizoguchi, R. (2009) 'Some Ontological Distinctions of Function based on the Role Concept', in: Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE), August 30 – September 2, 2009, San Diego, California, USA. DETC2009-87168: 1-10.

Kitamura, Y., Koji, Y., and Mizoguchi, R. (2005/2006) 'An Ontological Model of Device Function: Industrial Deployment and Lessons Learned', in: *Applied Ontology* 1: 237-262.

Kitamura, Y., Takafuji, S., and Mizoguchi, R. (2007) 'Towards a Reference Ontology for Functional Knowledge Interoperability', in: *Proceedings of the ASME 2007 International Design Engineering Technical Conferences &*

Computers and Information in Engineering Conference, IDETC/CIE '07, September 4-7, 2007, Las Vegas, Nevada, USA. DETC2007-35373: 1-10.

Kitamura, Y., Sano, T., Namba, K., and Mizoguchi, R. (2002) 'A Functional Concept Ontology and its Application to Automatic Identification of Functional Structures', in: *Advanced Engineering Informatics* 16 (2): 145-163.

Kitamura, Y., Segawa, S., Sasajima, M., Tarumi, S., and Mizoguchi, R. (2008) 'Deep Semantic Mapping Between Functional Taxonomies for Interoperable Semantic Search', in: Dominique, J. and Anutariya, C (eds.), *The Semantic Web*, Springer, Berlin:137-151.

Lind, M. (1994) 'Modeling Goals and Functions of Complex Industrial Plants', in: *Applied Artificial Intelligence* 8: 259-283.

Ookubo, M., Koji, Y., Sasajima, M., Kitamura, Y., and Mizoguchi, R. (2007) 'Towards Interoperability Between Functional Taxonomies Using an Ontology-Based Mapping', in: *Proceedings of the International Conference on Engineering Design, ICED '07, August 28-31, 2007, Paris, France*: 154.1-154.12.

Otto, K.N. and Wood, K.L. (2001) *Product Design: Techniques in Reverse Engineering and New Product Development*, Prentice Hall, Upper Saddle River NJ.

Pahl G. and Beitz, W. (1988) *Engineering Design: A Systematic Approach*, Springer, Berlin.

Price, C.J. (1998) 'Function-Directed Electrical Design Analysis', in: *Artificial Intelligence in Engineering*, 12 (4): 445-456.

Rosenman, M.A. and Gero, J.S. (1999) 'Purpose and Function in a Collaborative CAD Environment', in: *Reliability Engineering and System Safety* 64: 167-179.

Sasajima, M., Kitamura, Y., Ikeda, M., and Mizoguchi, R. (1996) 'A Representation Language for Behavior and Function: FBRL', in: *Expert Systems with Applications* 10 (3/4): 471-479.

Stone, R. B. and Wood K.L. (2000) 'Development of a Functional Basis for Design', in: *Journal of Mechanical Design* 122: 359-370.

Stone, R.B., McAdams, D.A., and Kayyalethekkel, V. (2004) 'A Product Architecture-Based Conceptual DFA Technique', in: *Design Studies* 25: 301-325.

Szykman, S., Racz, J.W., and Sriram, R.D. (1999) 'The Representation of Function in Computer-Based Design', in: *Proceedings of the 1999 ASME Design Engineering Technical Conferences, 1999, Las Vegas, Nevada, USA*. DETC99/DTM-8742: 1-14.

Szykman, S., Fenves, S.J., Keirouz, W., and Shooter, S.B. (2001) 'A Foundation for Interoperability in Next-Generation Product Development Systems', in: *Computer-Aided Design* 33: 545-559.

Van Eck, D., McAdams, D.A., and Vermaas P.E. (2007) 'Functional Decomposition in Engineering: A Survey', in: *Proceedings of the International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE '07, September 4-7, 2007, Las Vegas, Nevada, USA*. DETC2007-34232: 1-10.

Van Eck, D. (2009a) 'On the Conversion of Functional Models: Bridging Differences Between Functional Taxonomies in the Modeling of User Actions', in: *Research in Engineering Design* 21 (2): 99-111. DOI 10.1007/s00163-009-0080-7. Published online 21 october 2009.

Van Eck, D. (2009b) 'On Relating Functional Modeling Approaches: Abstracting Functional Models from Behavioral Models', in: *Proceedings of The 17th International Conference on Engineering Design (ICED 09), August 24-27, 2009, Stanford, California, USA*: 2.89-2.100.

Van Eck, D. (2010) 'Explaining and Relating Different Engineering Models of Functional Decomposition', in: Proceedings Design Research Society (DRS) International Conference, 07-09 July 2010, Montréal, Canada: 122.1-122.11.

Van Eck, D. (2011) 'Incommensurability and Rationality in Engineering Design: The Case of Functional Decomposition', forthcoming in: *Techné: Research in Philosophy and Technology*.

Vermaas, P.E. (2009), 'The Flexible Meaning of Function in Engineering', in: *Proceedings of the 17th International Conference on Engineering Design (ICED 09), August 24-27, 2009, Stanford, California, USA*: 2.113-2.124.

Zhang W.Y., Tor S.B., Britton G.A. and Deng Y.M. (2001), 'EFDEX: A Knowledge-based Expert System for Functional Design of Engineering Systems', in: *Engineering with Computers* 17: 339-353.

7 On the Conversion of Functional Models: Bridging Differences Between Functional Taxonomies in the Modeling of User Actions

This chapter appeared as an article in Research in Engineering Design:

Van Eck, D. (2010) 'On the Conversion of Functional Models: Bridging Differences Between Functional Taxonomies in the Modeling of User Actions', in: *Research in Engineering Design* 21 (2): 99-111. DOI 10.1007/s00163-009-0080-7.

Abstract

In this paper I discuss a methodology for the conversion of functional models between functional taxonomies developed by Kitamura et al. (2007) and Ookubo et al. (2007). They apply their methodology to the conversion of functional models described in terms of the Functional Basis taxonomy into functional models described in terms of the Functional Concept Ontology taxonomy. I argue that this model conversion harbors two problems. One, a step in this model conversion that is aimed to handle differences in the modeling of user features consists of the removal of Functional Basis functions. It is shown that this removal can lead to considerable information loss. Two, some Functional Basis functions, that I argue correspond to user functions, get re-interpreted as device functions in the model conversion. I present an alternative strategy that prevents information loss and information change in model conversions between the Functional Basis and Functional Concept Ontology taxonomies.

Keywords: Engineering design, functional modeling, knowledge exchange, user actions

7.1. Introduction

As can be seen in a current review by Erden et al. (2008) engineering design research has produced an impressive wealth of functional modeling approaches. In these approaches also a variety of definitions of functions, representations for functions and strategies for decomposing functions into sub functions are proposed. For instance, Chakrabarti (1998) and Deng (2002) distinguish functions corresponding to intended behaviors from functions corresponding to purposes. With regard to the representation of functions, Chakrabarti and Blessing (1996) identify three frameworks that are in use in engineering: verb-noun representations, input-output flow transformations, and input-output state transformations. Exponents of these representational frameworks are, for instance, the function-behavior-state approach of Umeda et al. (1996) in which verb-noun representations are used, the systematic approach of Pahl and Beitz (1988) in which input-output flow transformations are employed, and the adaptive design approach of Goel and Stroelia (1996) in which functions are represented by input-output state transformations. More recently, Deng et al. (2000a, 2000b) and Deng (2002) have added to this representational diversity by proposing the concepts of action and input-output flow of action transformation to represent functions. Concerning the decomposition of functions into sub functions, Van Eck et al. (2007) distinguish strategies in which functional decompositions are developed in a solution-neutral fashion from strategies in which known technical solutions for sub functions are incorporated from the outset. In addition, Goel et al. (2009) distinguish the modeling of artifacts qua "teleological systems" in which functions are realized by causal processes that mediate between function and structure, from the modeling of artifacts in which functions directly emerge from the shape of an artifacts' structure. One may conclude from the works above that many flowers bloom in the functional modeling segment of engineering design research.

A current research theme within functional modeling research concerns the development of methods that support the exchange and sharing of functional knowledge, both between engineering design teams and between members of design teams. To support these tasks, different methods are proposed. For instance, Nagel et al. (2007) have proposed a functional grammar to formalize functional modeling, and Szykman et al. (2001), Zhang et al. (2005), Kitamura and Mizoguchi (2004) and Kitamura et al. (2005/6) have proposed function ontologies to archive and exchange functional knowledge. With regard to the exchange and sharing of functional knowledge one can identify a reservation in the engineering literature though. Some observe that the concept of function is ambiguous. Chandrasekaran and Josephson (2000), for instance, state that this ambiguity hampers the automation of function-based reasoning tasks, and Deng (2002) remarks that this ambiguity undermines the interchange of research ideas. A second reservation one can make is that the above methods for knowledge exchange are framed *within* the confines of a specific functional modeling approach or taxonomy, each with their own definitions of functions and schemes for representing functions. Although knowledge exchange is facilitated by shared representational schemes, if the observations of Chandrasekaran and Josephson (2000) and Deng (2002) carry weight, challenges may emerge when the aim is to establish knowledge exchange *between* different functional taxonomies or approaches. Kitamura et al. (2007) and Ookubo et al. (2007) have developed a methodology to support such knowledge exchange between different functional taxonomies. Their method to establish this is by converting functional models between functional taxonomies. Ookubo et al. (2007) apply this method to a conversion of functional models described in terms of the Functional Basis taxonomy of Stone and Wood (2000) into functional models described in terms of the Functional Concept Ontology taxonomy of Kitamura et al. (2005/6).

In this paper I review this model conversion and argue that ambiguities surrounding functional representations pose challenges for the conversion of functional models between the Functional Basis and Functional Concept Ontology taxonomies. More specifically, I argue that the model conversion leads to a number of problems. Firstly, conceptual differences in the modeling of (parts of) users of devices, which are modeled in Functional Basis models and not in Functional Concept Ontology models, are handled by Ookubo et al. (2007) by removing Functional Basis functions that have input or output flows of human materials. I show that this removal can lead to considerable information loss in specific model conversions. Secondly, Hirtz et al. (2002) present the Functional Basis taxonomy as a taxonomy in which solely functions carried out by devices are described, and Ookubo et al. (2007) interpret the Functional Basis in a similar device-oriented fashion. I argue however that functions in the Functional Basis taxonomy may refer to functions of devices and to functions of user actions. I argue for this claim by analyzing Functional Basis models of a stapler, presented by Stone et al. (2004), and a power screwdriver, presented by stone et al. (1998) and Stone et al. (2000). I show that the device-oriented perspective of Ookubo et al. (2007) on the Functional Basis taxonomy leads them to re-interpret Functional Basis functions corresponding to user actions as functions carried out by devices, and that this re-interpretation leads to information change in specific model conversions.

I then propose an alternative strategy to handle differences in the modeling of user aspects between the Functional Basis and the Functional Concept Ontology taxonomies, which addresses both these problems of information loss and information change. My main aim in this paper is to propose a strategy that supports improved knowledge exchange between the taxonomies. In a more speculative discussion, I suggest that this alternative strategy may also offer a solution for two additional research issues, currently investigated by Ookubo et al. (2007), in model conversions between the Functional Basis and Functional Concept Ontology taxonomies.

The paper has the following organization. The Functional Basis and the Functional Concept Ontology approaches are presented in section 1. The model conversion methodology is presented in section 2, where the methodology is illustrated with a discussion of a conversion of a functional model of a stapler. The removal solution is further analyzed in section 3, and the user action analysis is presented there. These issues are illustrated with a discussion of conversions of functional models of a stapler and a power screwdriver. I then present my alternative strategy in section 4. I suggest how the proposed strategy may solve other research issues in the model conversion in section 5. I conclude the paper in section 6.

7.2. Functional modeling approaches

7.2.1. The Functional Basis approach

The Functional Basis (FB) approach, formulated by Stone and Wood (2000) is an approach to functional modeling that is aimed at creating a common and consistent functional design language, dubbed a functional basis. This language allows designers to model overall product functions as sets of interconnected sub functions. The FB approach is focused on especially the electromechanical and mechanical domains. The approach is presented as supporting the archiving, comparison, and communication of functional descriptions of existing products, as well as the engineering designing of new products. Since the approach was proposed it has been developed further. It is for instance used to develop a method to identify modules from functional models (see Stone et al.: 2000). It is also used to build a web-based repository in which functional decompositions of existing products are archived, as well as components counting as design solutions for the sub functions that are part of these decompositions.[1] The function and flow information of components archived in this repository has recently been employed by Bryant et al. (2007) in building a function based component ontology. In this ontology product components are classified based on their most commonly ascribed sub functions as archived in the repository. The FB has also been extended by Nagel et al. (2007) to domains outside engineering design proper, using the FB language to model manual processes.

In the FB approach, an overall product function refers to a general input/output relationship defined by the overall task of the product. This overall product function is described in a verb-object form and represented by a black-boxed operation on flows of materials, energies, and signals. A sub function, describing a part of the product's overall task, is also described in a verb-object form but represented by a well-defined basic operation on a well-defined basic flow of materials, energies, or signals. The black-boxed operations on general flows representing product functions are derived from customer needs, and the basic operations and basic flows representing sub functions are laid down in common and limited libraries that span the functional design space. These libraries are called a *functional basis*. In 2002, the FB approach was reconciled with an approach developed by Szykman et al. (1999) in collaboration with Julie Hirtz, Daniel McAdams, and Simon Szykman (2002), and coined Reconciled Functional Basis.

Stone and Wood (2000) present a three-step methodology to develop functional models or functional decompositions of products. The methodology starts with describing a product function in a verb-object form, represented by a black-boxed operation on flows of materials, energies, and signals. A chain of operations-on-flows is then specified, called a function chain, for each black box input flow, which transform that flow step-by-step into an output flow. These operations-on-flows are to be selected from the FB libraries. Finally, these temporally ordered function chains are aggregated into a single functional model of a product.

A FB model of a hand-held stapler is shown in figure 1, adapted from Stone et al. (2004). This model consists of temporally ordered chains of sub functions

1 http://function.basiceng.umr.edu/delabsite/repository.htmlref

that transform the material input flows of "hand", "staples" and "sheet", and the energy input flow of "human force", step by step into output flows.

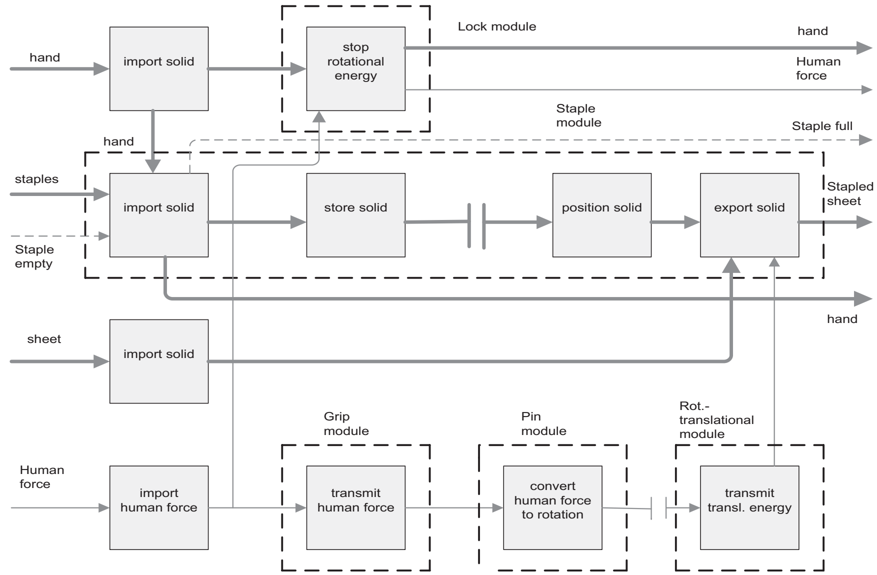

Figure 1: FB model of stapler, adapted from Stone et al. (2004)

7.2.2. The Functional Concept Ontology approach

The Functional Concept Ontology (FCO) approach, developed by Kitamura and Mizoguchi (2003, 2004) and Kitamura et al. (2005/6), is an approach to functional modeling that is aimed at facilitating the sharing of engineering functional knowledge. In this approach, in order to facilitate knowledge exchange, a set of modeling guidelines and a functional modeling language are developed to assist the systematic and reusable description of functional models of devices. The approach supports various tasks. It is for instance employed in building an ontology for functions and in developing an automated design support system (see Kitamura and Mizoguchi: 2003). The approach is currently deployed in an engineering division of a Japanese industrial firm for sharing functional device knowledge amongst its team members (see Kitamura et al.: 2005/6).

In the FCO approach, both behavioral models and functional models of devices are developed concurrently. Behaviors of devices and their components are defined as input-output relations between operand states. Operands refer to energy, fluid, material, motion, force, or information. Behaviors are represented as input-output state changes of properties of operands. Both overall functions and sub functions of devices are defined as roles played by behaviors, intended by designers or by users. Functions and sub functions are represented in terms of verb-operand pairs. The functional modeling language used in this approach consists of a generic set of verbs. These verbs are called functional concepts (see Kitamura and Mizoguchi: 2003; Kitamura et al.: 2005/6).

In a functional model or functional decomposition a set of sub functions is specified that realize the overall function. Sub functions and overall functions are represented in terms of functional concepts. In a functional decomposition it is furthermore specified by means of which technical principles the sub functions achieve the overall function. These specifications are referred to as "way of achievement" (see Kitamura and Mizoguchi: 2003, 157).

A FCO model of a stapler is shown in figure 2, adopted from Ookubo et al. (2007). This model consists of the goal function of the stapler, and sub functions of the modules and components of the stapler. Ways of achievement are shown in the model, specifying how the component functions realize the module functions, and how the module functions realize the goal function. The module function "combine sheets and staples", for instance, contributes to the realization of the goal function "combine sheets" by an "intermediate way" that represents the combining of paper sheets via staples acting as intermediates between the sheets.

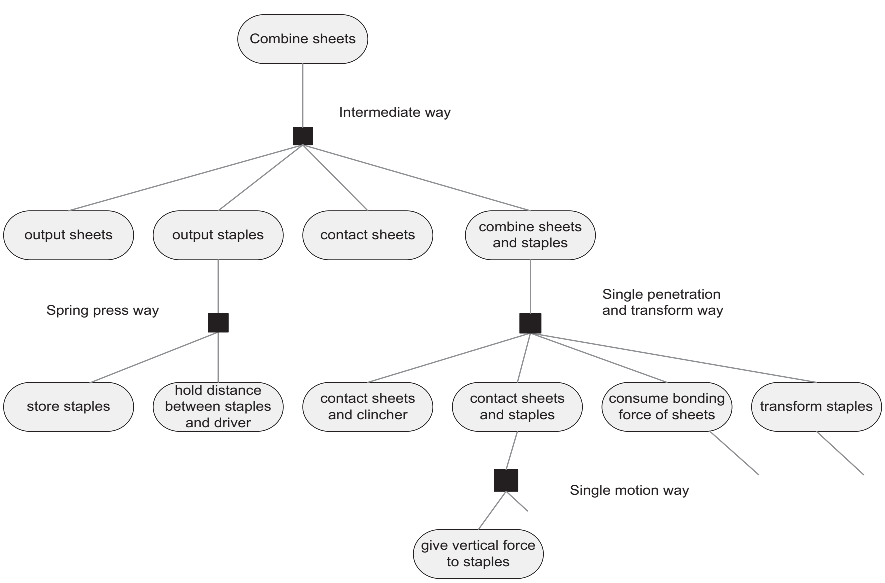

Figure 2: FCO model of stapler, adopted from Ookubo et al. (2007)

7.3. Functional model conversions

7.3.1. The conversion methodology

Kitamura et al. (2007) and Ookubo et al. (2007) aim with their methodology to support the conversion of a functional model *fm1*, which is based on one functional taxonomy *fx1*, into a (converted) functional model *fm2*, which is based on another functional taxonomy *fx2*. Functional models are converted by carrying out two steps. In the first step, the function terms of *fx1* are translated into function terms of the other taxonomy *fx2*. By this *fx1*-to-*fx2* function term translation, function terms in *fm1* are translated into function terms that will be included in *fm2*. In the second step, conceptual differences between models based on *fx1* and models based on *fx2* are explicated, and measures are developed and carried out to minimize these in the model conversion. By minimizing these differences, Ookubo et al. (2007) and Kitamura et al. (2007) aim to improve knowledge exchange between *fx1* and *fx2*. After these translation and modification steps a functional model *fm1* based on *fx1* is converted into a functional model *fm2* based on *fx2*.

In the first step, function terms are translated by using a "reference ontology" for functions (see Kitamura et al.: 2007, 2; Ookubo et al.: 2007). This reference ontology is used to identify the meaning of functions that are part of functional taxonomies and, based on this identification, to translate functions between taxonomies. In this reference ontology, function categories are defined which are stated to correspond to existing engineering meanings of functions. Definitions of these function categories are based upon the conceptual structure of the FCO approach (see Kitamura et al.: 2007; Ookubo et al.: 2007). The FCO concepts of device, behavior, function, and operand are further specified into subtypes called "descriptors of functions" (see Kitamura et al.: 2007, 3). With these descriptors of functions, different function categories are defined in the reference ontology. With these function categories they aim to identify different meanings of the concept of function in the engineering domain.

According to Kitamura et al. (2007) and Ookubo et al. (2007), by first classifying the function terms from *fx1* and *fx2* into function categories their meaning can be established. This classification is done by matching the definitions of function terms of *fx1* and *fx2*, as laid down in *fx1* and *fx2*, with the function categories in the reference ontology. The function terms in *fm1* are then translated into function terms that will be part of *fm2*. Depending on how these function terms are classified, different sorts of translations are carried out. Translations between function terms that are classified in the same function category are presented as straightforward, since the same meaning is attached to these function terms. These translations are called "within category" mappings (see Ookubo et al.: 2007, 154.6). When *fx1* includes function terms that are classified in a certain function category and *fx2* lacks function terms that can be classified in that same function category, translating these function terms from *fx1* to *fx2* involve more complex procedures. Such function terms (and their meaning) are namely part of one taxonomy, but not part of the other taxonomy (see Kitamura et al.: 2007, Ookubo et al.: 2007). These more complex translations are called "between category" mappings (see Ookubo et al.: 2007, 154.8).

After this first translation step an interim functional model *fm** results consisting of translated function terms that are represented in terms of *fx2*. In this phase, *fm** still has the same model structure as *fm1*, i.e., all the model features of *fm1* are also represented in *fm**. In the second step, conceptual differences between models based on *fx1* and models based on *fx2* are further explicated. This is done by comparing *fm1* with a functional model of the same device that is described in terms of *fx2* functions and according to *fx2* modeling criteria. Let us abbreviate this comparison model as *fmC*. The conceptual differences identified between *fm1* and the comparison model *fmC*, are then used to modify *fm**, resulting in *fm2*. After these translation and modification steps, a functional model *fm1* based on *fx1* is converted into a functional model *fm2* based on *fx2*.

7.3.2. The methodology at work: an FB-to-FCO model conversion

Ookubo et al. (2007) demonstrate their method by a conversion of an FB model (*fm1*) of a stapler represented in terms of the FB taxonomy (*fx1*) into a model (*fm2*) represented in terms of the FCO taxonomy (*fx2*). They also use a comparison FCO model of a stapler (*fmC*) in this conversion. This comparison FCO model (*fmC*) is used to identify conceptual differences between models based on the FB taxonomy and models based on the FCO taxonomy. The FB model (*fm1*), which Ookubo et al. (2007) adapt from Stone et al. (2004), is shown in figure 1, the comparison FC model (*fmC*) is shown in figure 2 and the converted FCO model (*fm2*) is shown in figure 3.[2]

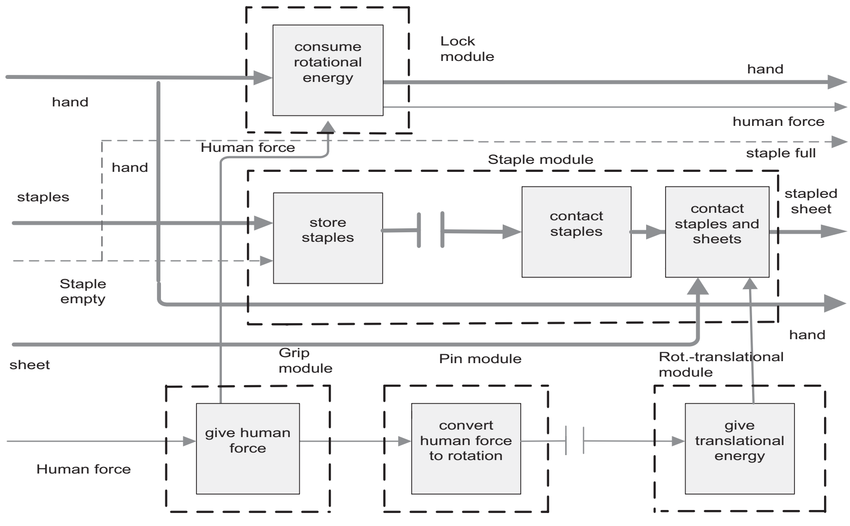

Figure 3: Converted FCO model of stapler, adopted from Ookubo et al. (2007)

Step 1: Translating FB function terms into FCO function terms

In the first step of the model conversion, Ookubo et al. (2007) translate functions both by "within category" mappings and by "between category" mappings. Most FB function terms and all FCO function terms are classified in the "flowing object" function category (see Kitamura et al.: 2007, 8). Flowing object functions correspond to a specific type of behavior, to wit: temporal changes in attributes of a physical entity, such as matter and energy flows or operands, within a device's system boundary. A role is attached to these behaviors in a

[2] I present the same adaptation of the FB model as Ookubo et al. (2007) give. This adaptation consists in excluding several operations-on flows-which are described in the original FB model. The vertical lines intersecting the "human force" flow and the "staples" flow represent this exclusion. In addition, to be precise, the converted FCO model is a converted FB model, expressed in terms of the FCO taxonomy. Please note that I use the term "converted FCO model" for brevity, but that this term has the meaning expressed above. Finally, Ookubo et al. (2007) use the concept of an 'interim functional model' at a conceptual level, but do not give an example of such a model. I follow their usage of this concept here.

teleological context (see Kitamura et al.: 2007; Ookubo et al.: 2007). Since most function terms in the FB and FCO taxonomies are classified as flowing object functions, the same meaning is attached to them. These function terms are translated by "within category" mappings. An example of a within category mapping of flowing object functions in the model conversion is the FB function "transmit human force" (figure 1) that is translated into the FCO function "give human force" (figure 3).

Some of the FB function terms in the FB stapler model are classified in the reference ontology as "system interface functions" (see Kitamura et al.: 2007, 8). System interface functions represent temporal changes in attributes of a physical entity on a system boundary. The FB "import" and "export" function terms are classified as system-interface functions. Since the FCO solely consists of "flowing object functions" (see Ookubo et al.: 2007), these FB function terms are translated by a between category mapping: the FB "import" and "export" operations-on-flows are translated in the model conversion into FCO input and output operands. Examples of between category translations are the "import solid (sheet)" and "export solid (stapled sheet)" functions of the FB model (figure 1) that Ookubo et al. (2007) represent in the converted FCO model as input and output operands of "sheet" and "stapled sheet" (see figure 3). This first translation step establishes a model (*fm**) in which translated functions are described in terms of the FCO taxonomy, but the model still has the structure of the FB model.

Other function categories into which FB function terms are classified are the "function with way of achievement" function category and the "composite device" function category (see Kitamura et al.: 2007, 8). Function terms of the FB model of the stapler are not classified in these categories, but I describe them here for sake of completeness. FB function terms classified as functions "with way of achievement" correspond to a flowing object function but in addition also refer to a way of achievement. An example given by Ookubo et al. (2007) is the FB term "link", which has both the (flowing object function) meaning of "coupling flows together" and also refers to how this coupling is achieved, namely by an "intermediary flow". FB function terms classified as "composite device" functions correspond to a flowing object function and the meaning of the term, as defined in the FB taxonomy, can be interpreted in two different ways viewed from the FCO taxonomy. An example given by Ookubo et al. (2007) is the FB term "guide" which they interpret as either referring to "supply motion" or to "change direction of motion".

Step 2: Modifying the interim model

After these translations, the FB model (*fm1*, figure 1) and the comparison FCO model (*fmC*, figure 2) are compared in the second step to identify conceptual differences between these models. Based on these differences, procedures are then developed to modify the interim model (*fm**). The end result of these translations and modifications is the converted FCO model (*fm2*, figure 3). Six conceptual differences are identified between the FB model and the comparison FCO model (see Ookubo et al.: 2007):

(1) In FCO models, goal functions are related to sub functions of modules, which are related to sub functions of components. FB models do not represent relationships between sub functions of modules and components.

(2) In FCO models, functions are not connected by operands, whereas functions are connected by flows they have as input or output in FB models.

(3) In FCO models, ways of achievements are described, whereas these are not described in FB models.

(4) In FCO models, changes in distance between physical objects – matter and energy flows/operands– are described, whereas these are not described in FB models.

(5) In FCO models, features of users are not described, whereas features of users are described by human material flows in FB models.

(6) In FCO models, material and energy operands may be grouped together in descriptions of functions, whereas material and energy flows are separated in FB models.

In the conversion of the FB stapler model, Ookubo et al. (2007) develop and carry out modifications to handle the difference in distance changes between physical objects (4) and to handle the difference in user features (5). They are currently investigating modifications to handle the difference in connections between functions (2) and to handle the difference in separating vs. grouping material and energy (6). The converted FCO model in figure 3 thus is the result from the translation of functions in the first step, and from the modifications in the second step that address differences in representing distance changes between flows/operands, and differences in representing (parts of) users of devices. This model is currently the endpoint of the conversion process (see Ookubo et al.: 2007).

The difference between the FB model and the comparison FCO model concerning distance changes between flows/operands is handled by adding an FCO function from the comparison model to the interim model. In the FB model, the "staple" and "sheet" flows enter the stapler as separate flows and exit as the combined flow "stapled sheet". The combining of these flows, referring to a change in distance between flows, is not represented in the FB model. In contrast, this combining is explicitly represented in the comparison FCO model by the function "contact staples and sheets". This difference is handled by adding this FCO function of the comparison model to the interim model.

The difference between the models regarding the representation of (parts of) users of devices is handled by *removing* FB functions in the model conversion. In the FB model, parts of users are represented in terms of flows of human materials such as "hand". In contrast, parts of users are not represented in the comparison FCO model, nor are they in FCO models of devices in general. The FCO treats (parts of) users as external to devices and therefore does not represent these in functional models of devices. Ookubo et al. (2007) handle this difference by removing FB functions that have input or output flows of human materials. In the interim model, for instance, the FB function "import solid (hand)" is removed.[3]

The end result of these translations and modifications is the converted FCO model (*fm2*) in figure 3. As can be seen, the function FCO "contact staples and sheets" is added to this model, and the FB function "import solid (hand)" is removed from this model.

7.4. Problems concerning user functions

In this section I argue that the FB-to-FCO model conversion, interesting though it is, leads to a number of problems. One, the removal of FB functions that have input or output flows of human materials may lead to considerable information loss. I argue that the converted FCO model of the stapler is a case in which the loss of information is limited. I then present an example of a FB model of a power screwdriver that gives a more extreme illustration of this information loss. Two, not all FB functions involving human material flows are actually removed

3 Ookubo et al. (2007, 10) state that "the function whose input or output is part of the user as flow is removed as a result of the transformation".

in the stapler model conversion. I argue that such a partial application of the removal solution may lead to function-to-function translations in which the meaning of some FB functions is altered. Ookubo et al. (2007) interpret the FB as modeling only functions performed by devices. However, I will argue that in the FB model of the power screwdriver some of the operations on human material flows represent user actions. If the device-oriented perspective of Ookubo et al. (2007) on the FB is maintained in the screwdriver case, reinterpretations of FB functions that correspond to user actions as functions carried out by devices will occur. This results in information change.

These problems lead to either information loss or information change, limiting the establishment of knowledge exchange and interoperability between taxonomies. I discuss these problems further in the sections below. I then present my alternative strategy in section 4 and show that it prevents these problems.

7.4.1. Removing FB functions

In the stapler model conversion, the removal solution is only partly applied. Firstly, a “hand” flow/operand is represented in the converted FCO model (*fm2*, figure 3). Secondly, the function “consume rotational energy” that transforms a “hand” flow/operand is still represented in the converted FCO model. When the removal solution would have been strictly applied, the functionality of the lock module, represented by the “consume rotational energy” function, would be lost in the conversion as well, in addition to the loss of the “import solid (hand)” function. This would have resulted in (limited) information loss. The FB model of the screwdriver, shown in figure 4, gives a more extreme illustration of this information loss.

The FB functional model of the power screwdriver, described by Stone et al. (1998) and Stone et al. (2000), is shown in figure 4. Stone et al. (1998) state that the first function chain represents the insertion and removal of the screw bit, that the second represents the fastening of the screw bit, that the third represents the positioning of the screwdriver and that the fourth represents the actuation of the device. The first and second function chains consist solely of functions that transform a (branching) human material “hand” flow from input to output. The third function chain consists of two FB functions that transform a “hand” flow into output.

If this screwdriver model would be selected for a model conversion, strict application of the removal solution will lead to the complete removal of the first three function chains. Consequently, a converted FCO model of the screwdriver will not represent the functionality of interchangeable screw bits, nor the functionality of the fastening/locking mechanism of the screw bit and neither the functionality of the positioning of the screwdriver.

Besides this information loss by removal of FB functions, a second problem may emerge in FB-to-FCO model conversions. In case of the screwdriver model, the misclassification of FB functions leads to information change.

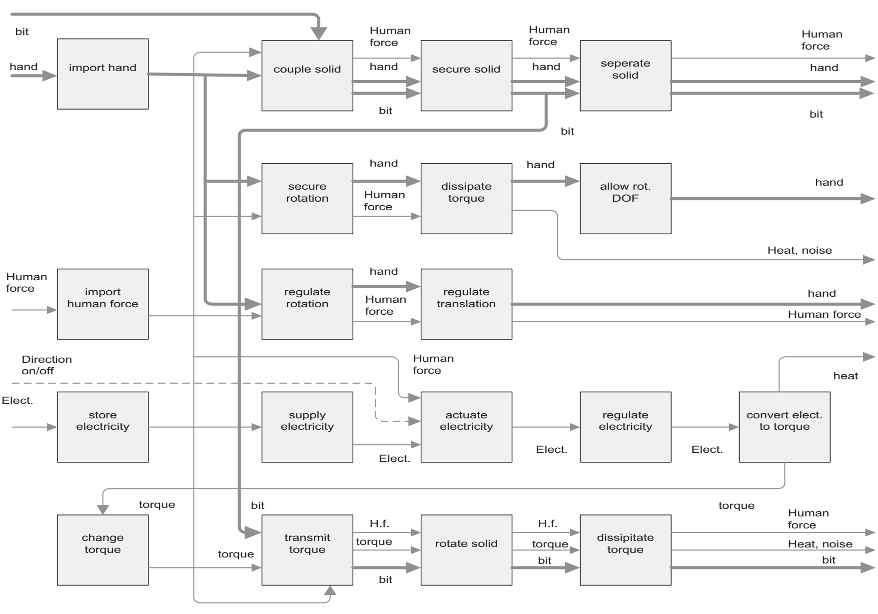

Figure 4: FB model of screwdriver, adopted from Stone et al. (1998) and Stone et al. (2000)

7.4.2. Misidentifications of FB functions

Both the FB approach and the FCO approach are presented as device-oriented taxonomies. Ookubo et al. (2007, 4) write that the FCO adopts "a device-oriented viewpoint" towards the modeling of functions of devices and components. Hirtz et al. also present the FB as device-oriented, by remarking that:

> We judge a function term's suitability based on whether or not it describes an operation that a product or device carries out on a flow. This ensures that the reconciled functional basis will consist of only *device functions*, as opposed to *user functions* (Hirtz et al.: 2002, 72, italics in original)

Hirtz et al. (2002) illustrate the difference between device functions and user functions with an example of a coffee machine: a coffee machine imports a flow of water, whilst a user pours water into the device. In this example they characterize the notion of a user function as an operation (pouring) carried out by a user on a flow (water). In other words, their characterization of a user function corresponds to a user action. The position taken by Hirtz et al. (2002) on the FB as modeling only device functions, and not user functions, is also adopted by Ookubo et al. (2007). Ookubo et al. (2007, 5) are explicit in their device-oriented perspective on the FB approach: "our functional concept ontology shares a device-oriented viewpoint with FB". This perspective informs their classification of FB functions into function categories, which are all categories of functions implemented by devices (see Kitamura et al.: 2007; Ookubo et al.: 2007).

In my view, examples can be given that contradict the device-oriented perspectives of Hirtz et al. (2002) and Ookubo et al. (2007). For instance, the FB function "import solid (hand)", which is removed in the stapler model conversion, is a function performed by a user. Whereas Ookubo et al. (2007) classify this function as a "system interface function", the importing of the hand represents an operation carried out by a user on a flow. Returning to the FB model of the power screwdriver, however, much more functions that have input or output flows of human materials correspond to user functions.

I argue that all the FB functions of the first function chain and the leftmost function of the second function chain of the power screwdriver exemplify the characterization of user functions given by Hirtz et al. (2002). As can be seen in figure 4, the first function chain is represented in terms of four FB functions that transform the flows "hand", "bit", and "human force" from input to output. By representing the insertion and removal of the screw bit in terms of a sequence of FB functions that transform a material "bit" flow, a "human force" flow, and a "hand" flow, the (de)coupling of the screw bit is represented as *a sequence of user functions*. More specifically, the (de)coupling of the screw bit is realized through human force applied through the hand, i.e., operations on flows carried out by a user. This analysis applies as well to the leftmost function "secure rotation" of the second function chain, which represents the manual fastening of the screw bit. In this function chain the FB function "secure rotation" transforms a "human force" flow and a "hand" flow, describing that the securing operation is realized by human force applied through the hand.

In this example, Ookubo et al.'s (2007) device oriented perspective on the FB, given that they do not or partially apply their removal solution, results in information change. The above FB functions, identified as user functions, will be misclassified as functions carried out by devices. The device-oriented perspective put forward by Hirtz et al. (2002) and adopted by Ookubo et al. (2007) unfortunately leads to function-to-function translations in which the meaning of functions is altered.

I do not want to end on these critical notes however. Both the model conversion methodology and the FB and FCO approaches are too valuable and interesting to end with these critical observations. In the remainder of this paper I present a possible solution for the problems outlined above, and apply it to both the converted FCO model of the stapler (figure 3) and to the FB model of the screwdriver (figure 4).

7.5. A Strategy for translating user functions

In order to avoid information loss and information change one can imagine the following alternative. I propose that after a model conversion, in which the removal solution is not applied, one can derive functions of devices and user functions from their corresponding operations-on-flows of the converted FCO model (*fm2*). These derived functions can then be represented in another FCO functional model. Whereas the converted FCO model (figure 3) is currently the endpoint of the conversion process in the proposal of Ookubo et al. (2007), I use it in my derivation strategy as an interim step for developing a derived FCO functional model. In this derived FCO model, both functions of devices and user functions are represented. The derived functions corresponding to functions of devices can be represented in terms of the FCO language. The derived functions corresponding to user actions can be represented in terms of an application of the function behavior representation language (FBRL), on which the FCO taxonomy is based, developed by Van der Vegte et al. (2004). Van der Vegte et al. (2004) apply the FBRL towards the modeling of user actions. If one accepts that FB functions may correspond to user actions and, hence, that translations of functions in FB-to-FCO model conversions may concern translations of user functions, this application becomes available as a means to represent user functions in a derived FCO model.

In the application of Van der Vegte et al. (2004) FBRL function verbs and operands are used to describe both functions carried out by devices and user actions with devices. Whereas in the former case a device is (somewhat confusingly) considered the "agent" of the function, in the latter the user is considered the "agent" of the function. For instance, in the case of a coffee machine, they describe functions of devices such as "conduct hot water" of a tube, and user actions such as "move basket" and "deform filter" (see Van der Vegte et al.: 2004, 6). In this extension of FBRL, models of user actions are represented separately from models of device functions. In contrast, my analysis of the FB screwdriver model shows that user functions are represented within this FB functional model.

Applied to the converted FCO model of the stapler, the result of my strategy is shown in figure 5. The derived FCO model in figure 5 has a similar format as the FCO comparison model (cf. figure 2), except that ways of achievement are not represented. As mentioned in section 3.1, if the removal solution had been applied consistently by Ookubo et al. (2007) the functionality of the lock module would be lost in the conversion. With my strategy this functional information is preserved straightforwardly.

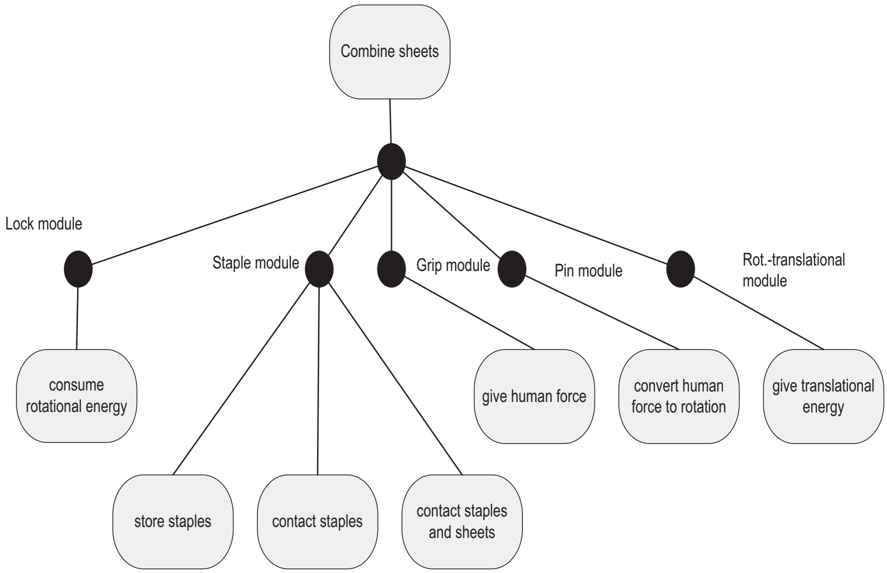

Figure 5: Derived FCO model of stapler. The goal function "combine sheets" is the same as the FCO model of figure 2. The functions are represented according to their grouping in function chains in the converted FCO model (cf. figure 3). Module information is given at the nodes (cf. figure 3).

Applied to the FB screwdriver model, the result of my strategy is shown in figure 6. For brevity, I have omitted the step of presenting a converted FCO model. In the derived FCO model in figure 6 both the functions carried out by the screwdriver and the user functions with the screwdriver are described. The derived FCO model in my proposal has a similar format as the FCO comparison model (cf. figure 2), except that ways of achievement are not represented. In line with the FBRL user action application, the user functions are grouped together and separated from the functions of the device. This model thereby accords with the device-oriented perspective of FCO. In accordance with the FCO taxonomy and FB-to-FCO model conversions, functions corresponding to the "import" operations-on- flows "import hand" and "import human force" are not derived. In my derivation strategy the flows that are present in a converted FCO model, including the human material flows/operands, are not represented in a derived FCO model. In my view, differences in the modeling of user aspects are with this strategy addressed, in a way that is congruent with the modeling guidelines of FCO, that does not entail information loss by the removal of FB functions (see figures 5 and 6) and that does not change the meaning of FB user functions (see figure 6).

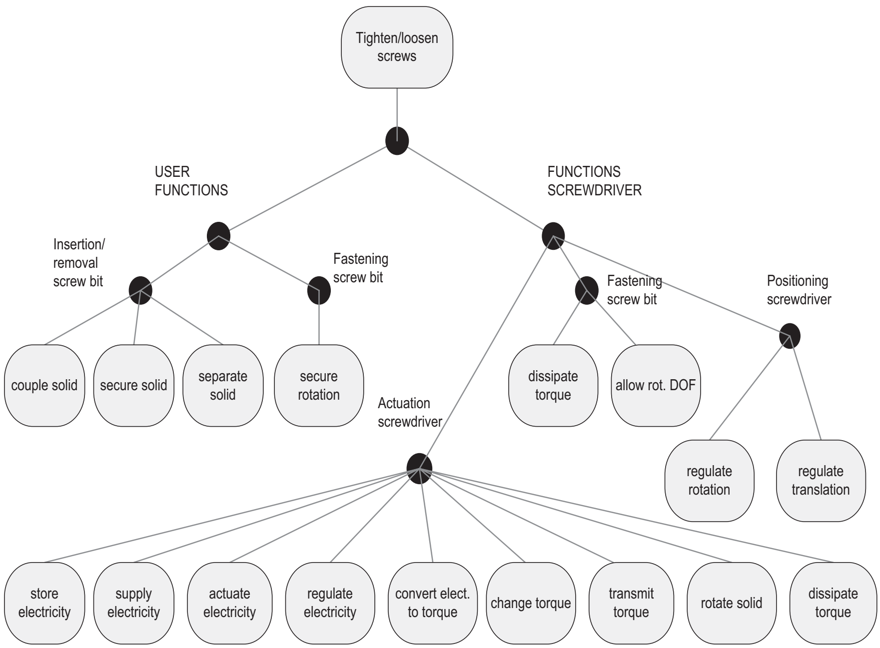

Fig 6: Derived FCO model of screwdriver. The goal function "tighten/loosen screws" is the same as the product function of the screwdriver given by Stone et al. (1998) and Stone et al. (2000). User functions are separated from functions carried out by the screwdriver. The derived functions are represented according to their grouping in function chains in the FB model of the power screwdriver (cf. Figure 4). The functionality of the function chains, adopted from Stone et al. (1998), is described at the nodes.

In line with the aim underlying the conversion methodology to establish functional knowledge exchange knowledge between taxonomies, I present my alternative as a conceptual tool to address the loss of and changes in functional information in FB-to-FCO model conversions. A general suggestion that may be drawn from the presented analysis is that the inclusion of a function category of user functions in the reference ontology would enhance the translation possibilities with the conversion methodology. Given that other functional modeling approaches are developed in which user functions are described, such as Otto and Wood's Reverse Engineering and Redesign approach (1998, 2001), this seems an extension worth considering. Inclusion of a user function category would enable the identification and translation of user functions between taxonomies. How to proceed in specific cases will depend on the specific taxonomies paired in a model conversion. The strategy described above gives a handle on this issue in the case of FB-to-FCO model conversions.

The solution that I have presented is a conceptual one and not empirically clear-cut. The method that I adopt in this paper is analytic, by which I mean the analysis of concepts and assumptions that are used in the functional modeling approaches and the conversion methodology. The advantage of this method, in my view, is that it is well suited for elucidating concepts. It is however less suited for empirical testing. I acknowledge this limitation, and therefore leave validation of my proposal by the empirical experts.

The position that I developed above may have additional practical utility. It may offer a solution for two research issues in FB-to-FCO model conversions, currently investigated by Ookubo et al. (2007). And, in addition, my solution for these research issues may be extended to model conversions between other functional modeling approaches. I outline this solution in the next section.

7.6. Discussion: generalizing the derivation strategy

In this paper I have focused my derivation strategy on the translation of user functions between the FB and FCO approaches, addressing functional information loss and information change. My strategy is not limited to these two approaches. It generalizes, for instance, straightforwardly to conversions between Otto and Wood's Reverse Engineering and Redesign approach (1998, 2001), in which both device functions and user functions are represented by operations-on-flows, and the FCO approach. Since in Otto and Wood's approach user functions are represented exactly the same as in the FB approach, these would be removed in the conversion methodology of Kitamura et al. (2007) and Ookubo et al. (2007). In contrast, my strategy enables their representation in a derived FCO model, thus preventing information loss.

The derivation strategy that I have described and demonstrated seems, in addition, an adequate tool for solving two research issues currently investigated by Ookubo et al. (2007), mentioned in section 2.2. These research issues concern, first, the modeling of connections between functions in terms of flows that are modeled in FB models, but not in FCO models, and, second, the separation of material and energy in FB models, which, instead, may be combined in FCO models. For instance, the FCO description of the stapler's sub function "give vertical force to staples", in which energy and material are combined (see Ookubo et al.: 2007; cf. figure 2). With my strategy both these differences can be handled. Regarding the first issue, my proposed step of deriving device functions and user functions from their corresponding operations-on-flows in the converted FCO model leaves flow connections between functions in a converted FCO model. Thereby, flow connectivity information is preserved in an (interim) FB-FCO model conversion. Yet, in accordance with FCO modeling rules, this connectivity is not represented in a derived FCO model. My strategy thus handles this difference, without information loss (cf. figures 5 and 6). Regarding the second issue, a solution can be developed along similar lines. The separation of material and energy in FB descriptions of functions follows from the fact that they are connected by separate material and energy flows, which they have as input and output. Since in a derived FCO model in my strategy, functions are not connected by material and energy flows, it is also no longer required that functions are described in terms of the separation of material and energy. In my strategy, one can thus take the separation of material and energy as a feature of converted FCO models, but not of derived FCO models. Thereby, information on material and energy separation is preserved in an FB-FCO model conversion. Yet, in accordance with FCO modeling rules, this separation is not represented in a derived FCO model. My strategy thus handles this difference, without information loss.

Taking these next steps in the above fashion in FB-FCO model conversions seems a promising way tackle differences between other approaches in model conversions as well. The connectivity between functions and the separation of material and energy are two features that are highly discriminative between functional modeling approaches (see for instance the review by Erden et al. (2008)). The specific details of such model conversion-cases will, of course, depend on the approaches paired in a model conversion. The strategy proposed here provides a general conceptual framework for developing them.

The examples of the stapler and screwdriver models discussed in this paper also highlight a general research challenge that must be addressed in model conversions between specific functional modeling approaches: when certain types of functions are modeled in one approach, but not in the other, measures need to be developed that prevent information loss and/or information change problems. For instance, in the Multi Level Flow approach of Lind (1994) functions may correspond to operator actions, whereas in the FB and FCO approaches they do not. This difference needs to be addressed in order to avoid information loss.

Future work is aimed at investigating in detail the issues raised in this discussion section. The main contribution presented in this paper concerns an alternative way of handling differences in the modeling of user aspects in FB-to-FCO model conversions that prevents the loss of and change in functional information.

7.7. Conclusions

In this paper I have reviewed a methodology for the conversion of functional models between functional taxonomies developed by Kitamura et al. (2007) and Ookubo et al. (2007). They apply their methodology to the conversion of functional models described in terms of the Functional Basis taxonomy into functional models described in terms of the Functional Concept Ontology taxonomy. I have argued that these model conversions harbor two problems. One, a step in these model conversions that is aimed to handle differences in the modeling of user features is shown to lead to considerable information loss. Two, it is shown that Functional Basis functions that correspond to user actions get reinterpreted as functions carried out by devices, leading to information change. After this analysis I have presented and demonstrated an alternative strategy for solving this information loss and information change. I ended the paper by outlining how my alternative strategy may also solve other research issues, both in model conversions between the Functional Basis and Functional Concept Ontology taxonomies and between other functional modeling approaches. Future work is aimed at testing the strategy in detail with respect to these research issues. At a more general level, the research presented here is submitted as a contribution towards the clarification of the meaning and representation of functions in engineering, and towards the support of functional knowledge exchange between functional modeling approaches.

Acknowledgements

I would like to thank the reviewers for their insightful comments. I would also like to thank Maarten Franssen, Yoshinobu Kitamura, Peter Kroes, Riichiro Mizoguchi, and in particular Pieter Vermaas for their incisive comments on earlier versions of this paper. This research is supported by the Netherlands Organization of Scientific Research (NWO).

References

Bryant, C. R., McAdams, D. A., Stone, R. B., Kurtoglu, T., and Campbell, M. I. (2006) 'A Validation Study of an Automated Concept Generator Design Tool', in: *Proceedings of the 2006 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, DETC2006-99489: 1-12.

Bryant, C.R, Stone, R. B., Greer, J.L., McAdams, D.A., Kurtoglu, T., and Campbell, M.I. (2007) 'A Function-Based Component Ontology for Systems Design', in: *Proceedings of International Conference on Engineering Design (ICED 07), August 28-31, 2007, Paris, France*: 478.1-478.12.

Chakrabarti, A. (1998) 'Supporting Two Views of Function in Mechanical Designs', in: *Proceedings 15th National Conference on Artificial Intelligence, AAAI'98, July 26-30, 1998, Madison, Wisconsin, USA*.

Chakrabarti, A. and Blessing, L. (1996) 'Special Issue: Representing Functionality in Design', in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 10 (5): 251-253.

Chandrasekaran, B. and Josephson, J.R. (2000) 'Function in Device Representation', in: *Engineering with Computers* 16: 162-177.

Deng, Y.M. (2002) 'Function and Behavior Representation in Conceptual Mechanical Design', in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 16: 343-362.

Deng, Y.M., Tor, S.B., and Britton, G.A. (2000a) 'A Dual-Stage Functional Modelling Framework with Multi-Level Design Knowledge for Conceptual Mechanical Design', in: *Journal of Engineering Design* 11 (4): 347-375.

Deng, Y.M., Tor, S.B., and Britton G.A. (2000b) 'Abstracting and Exploring Functional Design Information for Conceptual Mechanical Product Design', in: *Engineering with Computers* 16: 36-52.

Erden, M.S., Komoto, H., Van Beek, T.J., D'Amelio, V., Echavarria, E, and Tomiyama, T. (2008), 'A Review of Function Modelling: Approaches and Applications', in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 22: 147-169.

Goel, A.K. and Stroelia, E. (1996) 'Functional Device Models and Model-Based Diagnosis in Adaptive Design, in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 10: 355-370.

Goel, A.K., Rugaber, S., and Vattam, S. (2009) 'Structure, Behavior, and Function of Complex Systems: The Structure, Behavior, and Function Modeling Language', in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 23: 23-35.

Hirtz, J., Stone, R. B., McAdams, D. A., Szykman, S., and Wood, K. L. (2002) 'A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts', in: *Research in Engineering Design* 13: 65-82.

Kitamura, Y. and Mizoguchi, R. (2003) 'Ontology-Based Description of Functional Design Knowledge and its Use in a Functional Way Server', in: *Expert Systems with Applications* 24: 153-166.

Kitamura, Y. and Mizoguchi, R. (2004) 'Ontology-Based Systematization of Functional Knowledge', in: *Journal of Engineering Design* 15 (4): 327-351.

Kitamura, Y., Koji, Y., and Mizoguchi, R. (2005/2006) 'An Ontological Model of Device Function: Industrial Deployment and Lessons Learned', in: *Applied Ontology* 1: 237-262.

Kitamura, Y., Takafuji, S., and Mizoguchi, R. (2007) 'Towards a Reference Ontology for Functional Knowledge Interoperability', in: *Proceedings of the ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE '07, September 4-7, 2007, Las Vegas, Nevada, USA*. DETC2007-35373: 1-10.

Lind, M. (1994) 'Modeling Goals and Functions of Complex Industrial Plants', in: *Applied Artificial Intelligence* 8: 259-283.

Nagel, R.L., Stone, R.B., and McAdams, D.A. (2007) 'A Theory for the Development of Conceptual Functional Models for Automation of Manual Processes', in: *Proceedings of the ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE), September 4-7, 2007, Las Vegas, Nevada, USA*. DETC2007-35620: 1-13.

Nagel, R.L., Vucovich, J.P., Stone, R.B., and McAdams, D.A. (2007) 'Signal Flow Grammar from the Functional Basis', in: *Proceedings of International Conference on Engineering Design (ICED 07), August 28-31, 2007, Paris, France*: 401.1-401.10.

Ookubo, M., Koji, Y., Sasajima, M., Kitamura, Y., and Mizoguchi, R. (2007) 'Towards Interoperability Between Functional Taxonomies Using an Ontology-Based Mapping', in: *Proceedings of the International Conference on Engineering Design, ICED '07, August 28-31, 2007, Paris, France*: 154.1-154.12.

Otto, K.N. and Wood, K L. (1998) 'Product Evolution: A Reverse Engineering and Redesign Methodology', in: *Research in Engineering Design* 10 (4): 226-243.

Otto, K.N. and Wood, K.L. (2001) *Product Design: Techniques in Reverse Engineering and New Product Development*, Prentice Hall, Upper Saddle River NJ.

Pahl G. and Beitz, W. (1988) *Engineering Design: A Systematic Approach*, Springer, Berlin.

Stone, R. B. and Wood K.L. (2000) 'Development of a Functional Basis for Design', in: *Journal of Mechanical Design* 122: 359-370.

Stone, R.B., Wood, K.L., and Crawford, R.H. (1998) 'A Heuristic Method to Identify Modules from a Functional Description of a Product', in: *Proceedings of 1998 ASME Design Engineering Technical Conferences, September 13-16, 1998, Atlanta, Georgia, USA*. DETC98/DTM-5642: 1-11.

Stone, R.B., Wood, K.L., and Crawford, R.H. (2000) 'Using Quantitative Functional Models to Develop Product Architectures', in: *Design Studies* 21: 239-260.

Stone, R.B., McAdams, D.A., and Kayyalethekkel, V. (2004) 'A Product Architecture-Based Conceptual DFA Technique', in: *Design Studies* 25: 301-325.

Szykman, S., Racz, J.W., and Sriram, R.D. (1999) 'The Representation of Function in Computer-Based Design', in: *Proceedings of the 1999 ASME Design Engineering Technical Conferences, 1999, Las Vegas, Nevada, USA*. DETC99/DTM-8742: 1-14.

Szykman, S., Fenves, S.J., Keirouz, W., and Shooter, S.B. (2001) 'A Foundation for Interoperability in Next-Generation Product Development Systems', in: *Computer-Aided Design* 33: 545-559.

Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y., and Tomiyama, T. (1996) 'Supporting Conceptual Design based on the Function-Behavior-State-Modeler', in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 10: 275-288.

Van der Vegte, W., Kitamura, Y., Koji, Y., and Mizoguchi, R. (2004) 'Coping with Unintended Behavior of Users and Products: Ontological Modeling of Product Functionality and Use', in: *Proceedings of the ASME 2004 Design Engineering Technical Conferences & Computers in Engineering Conference, September 28-october 2, 2004, Salt Lake City, Utah, USA*. DETC2004-57720: 1-10.

Van Eck, D., McAdams, D.A., and Vermaas P.E. (2007) 'Functional Decomposition in Engineering: A Survey', in: *Proceedings of the International Design*

Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE '07, September 4-7, 2007, Las Vegas, Nevada, USA. DETC2007-34232: 1-10.

Zhang, W.Y., Lin, L.F., Tong, R.F., Li X., and Dong, J.X. (2005) 'XB-FES/o: An Ontology-Based Scheme for Functional Modelling in Design for Semantic Web Applications', in: *The 9th International Conference on Computer Supported Cooperative Work in Design, Vol. 2*: pp. 1146-1151.

8 Mechanisms, Functional Hierarchies, and Levels of Explanation

This chapter is submitted as an article.

Abstract

In this paper I analyse and extend the concept of role function in Craver's account of mechanistic explanations. I argue that Craver's earlier account conflates descriptions of activities that fill roles with descriptions of role functions. This conflation leads to two problems. Firstly, mechanistic schemes fail to make insightful how activities of entities contribute to an explanandum activity. Secondly, inter-level integration of mechanistic explanations is not supported. I further argue that Craver's later account displaces descriptions of activities that fill roles altogether. This displacement leads to the problem that mechanistic explanations are left incomplete. I outline a solution to avoid these problems. I detail my analysis in terms of Craver's showcase example of the functions of the heart in the circulatory system.

Keywords: Mechanistic explanation, role function description, isolated description

8.1. Introduction

Roughly, the last fifteen years or so the concepts of mechanism and mechanistic explanation have been the subject of detailed philosophical analysis and dispute (see. Bechtel and Richardson: 1993; Glennan: 1996, 2002; Machamer et al.: 2000; Woodward: 2002). And the diverse set of topics that are scrutinized can be taken as evidence for the vibrancy of the field. Analyses are, for instance, devoted to the nature of mechanisms (see Glennan: 1996, 2002; Machamer et al.: 2000; Tabery: 2004; Torres: 2009), the relationship between mechanistic explanation and reduction (see Craver: 2005, 2007; Wright and Bechtel: 2006; Wright: 2007), the distinction between covering-law explanation and mechanistic explanation (see Bechtel and Abrahamsen: 2005; Craver: 2008), the connection between lineage explanation and mechanistic explanation (see Calcott: 2009), levels of mechanistic explanation (see Craver: 2002, 2005, 2007) and the articulation of normative constraints for mechanistic explanations (see Craver: 2006, 2007). A topic that has also received considerable attention is the concept of function in mechanistic explanations and/or its analysis from a mechanistic perspective (e.g., see Mundale and Bechtel: 1996; Craver: 2001; Wimsatt: 2002; Piccinini: 2007). Craver's (2001) treatment of this concept in the context of mechanistic explanations is among the most elaborate ones.

Craver advances his account of functions as a refinement of Cummins' (1975, 1983) account of functions.[1] In Cummins' account, function ascriptions are conceptually dependent on an "analytical explanation" (see Cummins: 1975, 762) of a capacity of a containing system. In an analytical explanation, a systemic capacity is explained in terms of a number of other capacities of the system's component parts and/or processes that jointly realize the manifestation of the systemic capacity (1975: 760). Functions are ascribed to those capacities of the component parts/processes that figure in an analytic explanation of a system's capacity. Craver (2001), however, argues that Cummins' notion of an analytic explanation is imprecise and, hence, the description and ascription of functions as lacking in precision as well. To improve upon Cummins, Craver proposes to restrict the notion of system to the notion of mechanism, and to make the description and ascription of functions conceptually dependent on the details of a mechanism's organization. He submits that his account delivers a "richer and more precise image" of functions than Cummins' (see Craver: 2001, 57). In addition to refining Cummins' account of functions, Craver submits that his treatment of functions supports the inter-level integration of mechanistic explanations. Craver showcases both the feats of increased precision and inter-level integration in terms of the functions of the heart in the circulatory system.

In this paper I argue that Craver's treatment of functions harbors problems. Specifically, I will argue that (i) Craver's (2001) earlier account conflates descriptions of activities that fill roles with descriptions of role functions, and that (ii) his later account (see Craver: 2002, 2005, 2007) displaces descriptions of activities that fill roles altogether. I argue that the conflation in his earlier work leads to the problem that mechanistic schemes purported to explain an ex-

1 Other authors advance revisions of Cummins' account of functions, and relate their revised conception to mechanistic explanations (see Davies: 2001; Sustar: 2007).

planandum activity – an overall activity of a mechanism – fail to make insightful *how* activities of entities contribute to an explanandum activity. I argue that only role function descriptions, and not descriptions of activities that fill roles, are equipped for this job. By the conflation, however, filler descriptions are assumed (and presupposed) to make these contributions insightful as well. Yet, since these descriptions do not specify *how* activities of entities contribute to an explanandum activity, one fails to understand how this explanandum activity is brought about. This conflation also leads to a second problem that the integration of mechanistic explanations between levels is not supported. I further argue that the displacement in Craver's later work leads to the problem that a mechanistic explanation for an explanandum activity is left incomplete: without descriptions of activities that fill roles one does not explicate which activities of entities sub serve these role functions, and one thus fails to explicate what enables an entities' activity to play a given role function. I submit that (in the context of the restriction to mechanisms) Craver's account of functions is, hence, not precise enough. These problems are not beyond repair. After this analysis, I outline a solution to remedy the above-mentioned conflation and displacement. My analysis furthermore suggests that a mechanistic explanation for an explanandum activity gains explanatory leverage when contributions between different functions of an item are explicated, a feat that is not discussed in Craver's account. I outline a proposal to capture such contributions, in which I employ concepts from an approach towards the modeling of technical functions that is developed in the engineering sciences-domain.

This paper is organized as follows. I discuss Craver's concept of function in section 2. I specify the problems related to this concept in section 3, and discuss contributions between an item's functions in section 4. I suggest my solution and outline my proposal in section 5. Conclusions are given in section 6.

8.2. Craver's concept of role function

I focus the discussion of Craver's (2001) concept of role function around two major goals that he aims to satisfy with this concept. One, improving Cummins' concept of function. Two, integrating mechanistic explanations between lower and higher levels.[2]

8.2.1. Background: functions and mechanistic organization

Craver's (2001) analysis of the concept of function is, among others, aimed at refining Cummins' (1975, 1983) concept of function. In Cummins' account, function ascriptions are conceptually dependent on an "analytical explanation" (1975, 762) of a capacity of a containing system. A system capacity, coined analyzed capacity, is explained in terms of a number of other capacities, coined analyzing capacities, of the system's component parts and/or processes that jointly realize the manifestation of the system capacity (1975, 760). Functions are ascribed to those capacities of the component parts/processes that figure in an analytic explanation of a system capacity. In Cummins' account, more formally, the ascription of a function to an item X is specified as follows:

> X functions as a ϕ in S (or the function of X in S is to ϕ) relative to an analytic account A of S's capacity to ψ just in case X is capable of ϕ-ing in S and A appropriately and adequately accounts for S's capacity to ψ by, in part, appealing to the capacity of X to ϕ in S (Craver: 2001, 55; Cummins: 1975, 762) [3]

In case of the heart, Cummins asserts that the heart (X) functions as a pump (ϕ) in the circulatory system (S) relative to an analytical account (A) of the circulatory system's (S's) capacity to transport food, oxygen, and wastes (ψ) just in case the heart (X) is capable of pumping (ϕ-ing) in the circulatory system (S) and the analytical account (A) appropriately and adequately accounts for the circulatory system's (S's) capacity to transport food, oxygen, and wastes (ψ) by, in part, appealing to the capacity of the heart (X) to pump (ϕ) in the circulatory system (S). According to Cummins, the explanatory interest of an analytic account is roughly proportional to (1975, 764):

---

[2] Craver (2001) pursues a third major goal with his function concept: formulating a contextual variety of causal/mechanical explanation. I do not consider this work in this paper.

[3] I use here Craver's (2001) notation: the representation of the symbols slightly differs from Cummins (1975).

1. The extent to which the analyzing capacities are less sophisticated than the analyzed capacities
2. The extent to which the analyzing capacities are different in type from the analyzed capacities
3. The relative complexity of the organization of component parts/processes that is attributed to the system.

Craver's main criticism is directed at the concept of organization in Cummins' third criterion. In Cummins' account, organization refers to something that can be specified in a program or a flow chart (1975). Yet, Craver argues, since almost anything can be described in such a fashion, Cummins' concept of organization is too abstract for specifying the content of a function and for ascribing a function to an item in a precise manner (see Craver: 2001). In effect, Craver concludes that in Cummins' account one is unable to distinguish interesting from uninteresting analytic explanations and to distinguish warranted from unwarranted function ascriptions.

Craver aims to specify Cummins' concept of function further by restricting the notion of a system to that of a mechanism, and by detailing the description and ascription of an item's function in terms of organizational features characteristic of mechanisms (see Craver: 2001, 59-62). His preferred definition of mechanism(s) reads as follows:

> Mechanisms are collections of entities and activities organized in the production of regular changes from start or set up conditions to finish or termination conditions (Craver: 2001, 58; cf. Machamer et al.: 2000)

Entities refer to the physical parts of a mechanism and are taken by Craver to correspond to the X's in Cummins' definition, and are also represented by X's as in Cummins' definition. Activities are the things that entities do, by themselves or in interplay with other entities, and are represented by the ϕ's in Cummins' definition.[4] Entities and activities are organized in a specific fashion, constituting a mechanism, and this organization allows them to do something. What a mechanism does – the activity of the mechanism as a whole – is represented by the ψ in Cummins' definition. The S in Cummins' definition is used to repre-

[4] Craver uses different terminology than Cummins, switching from capacity talk to activity talk. For an explication and defense of this shift, see Machamer et al. (2000) and Craver (2001). This distinction has no bearing on the discussion in this paper.

sent a mechanism. Mechanisms exhibit an active organization: entities in a mechanism act and interact with each other. This active organization is sustained by the spatial organization of the entities and the temporal organization of the activities in the mechanism. Specific spatial features of the entities in a mechanism, such as their shape, size, orientation and location, and specific temporal features of the activities in a mechanism, such as their order, rate and duration, allow S's ψ-ing. An analytical account for S's ψ-ing, that is, a mechanistic explanation, is a description of a mechanism that specifies and makes insightful how activities and entities are organized actively, spatially and temporally such that together they instantiate S's ψ-ing.

Craver employs these organizational features of mechanisms in the ascription and description of functions. In Craver's account, the ascription of a function to an X's ϕ-ing involves specifying the contribution of an X's ϕ-ing to an S's ψ-ing in terms of/by referring to how X's ϕ-ing is organized with other X's (and their ϕ-ings), which allows X's ϕ-ing to make that contribution. The content of a function description is fixed by a function ascription: a description of a function specifies the contribution of an X's ϕ-ing to S's ψ-ing in terms of/by referring to how X's ϕ-ing is organized with other X's (and their ϕ-ings) such that X's ϕ-ing contributes to S's ψ-ing. Craver labels his concept of function "mechanistic role function" (see Craver: 2001, 61).[5]

8.2.2. Contextual and isolated descriptions of an X's ϕ-ing

Craver argues that by basing the ascription and description of role functions on the details of organization characteristics of mechanisms, he delivers a "richer and more precise image" than Cummins is able to provide on the ascription and description of functions. Craver asserts that the number of organizational features of a mechanism that one can employ in the ascription of a role function, and hence that is referred to in its description, is flexible. Regarding this flexibility, Craver (2001, 63-64) considers three ways to describe the role function of the heart's (X's) ϕ-ing in the mechanism (S) for the circulatory system's ψ-ing (delivering goods to tissues):

5 For the sake of simplicity, I will use the term role function instead of mechanistic role function when discussing Craver. From here on, I will use the term function when referring to Cummins' notion of function, and the term role function when referring to Craver's notion of function.

i. distributing oxygen and calories to the body
ii. pumping blood through the circulatory system
iii. expelling blood

The role function descriptions i, ii, and iii of the heart's ϕ-ing ("pumping blood") vary in the number of organizational features of the mechanism that they refer to, and that is considered in their ascription and description. The description 'expelling blood' (iii) refers to the mechanistic context of (the availability of) blood (see Craver: 2001, 64). In the description 'pumping blood through the circulatory system' more mechanistic context is taken into account. It refers to the mechanistic context of (the availability of) blood, and the relevant organization of veins and arteries (see Craver: 2001, 64).[6] Craver coins a role function description of an X's ϕ-ing that refers to entities and/or activities other than and outside X and with which X is organized, a "contextual" description of an X's ϕ-ing. Alternatively, he coins a description of that X's ϕ-ing that does not refer to entities and/or activities other than and outside X, an isolated description of an X's ϕ-ing (see Craver: 2001, 65). In such descriptions, one makes no commitments concerning the mechanistic context in which X's ϕ-ing is embedded by drawing an idealized dividing line at the spatial boundary of X. Craver considers one isolated description of the heart's ϕ-ing:

iv. contracting

In Craver's view, this conceptual distinction exposes an ambiguity in Cummins' account of functions. Craver asserts that in Cummins' account it is ambiguous whether a function is a capacity described contextually or, instead, whether a function is a capacity described in isolation (see Craver: 2001, 64-65).[7] That is, in activity talk, it is ambiguous whether Cummins' concept of function refers to a role function or to an activity that fills it. Craver submits his distinction between contextual and isolated descriptions of an X's ϕ-ing as a refinement on Cummins's concept of function. In addition, as will be outlined in the next section, the conceptual distinction between contextual and isolated descriptions of an X's

---

6 The term 'circulatory system' needs not necessarily refer to such mechanistic context. Craver (2001) does take the term to refer to such context in his example.

7 I do not consider the question here whether or not function ascriptions in Cummins' account are ambiguous in this sense.

ϕ-ing plays a key role in Craver's (2001) framework on establishing inter-level integration of mechanistic explanations.

8.2.3. Inter-level integration of mechanistic explanations

In addition to contextual and isolated descriptions of an X's ϕ-ing, Craver formulates a third descriptive perspective on an X's ϕ-ing (2001, 65). Isolated descriptions frame "constitutive" descriptions of an X's ϕ-ing. That is, isolated descriptions – 'contracting' (iv) in the heart example – specify an explanandum activity for which a mechanistic explanation at a lower level is developed (more on levels below). Craver coins such a lower-level explanation, a "constitutive" perspective on an X's ϕ-ing. Such a lower-level explanation or constitutive perspective specifies how activities (σ-ings) and entities (Ps) are organized such that together they instantiate X's ϕ-ing. Craver exemplifies this, in the heart example, by (organized) contractions (σ-ings) of heart muscles (Ps). In his three-tiered perspective, Craver recognizes three levels: one for S's ψ-ing, one for X's ϕ-ing, and one for P's σ-ing (2001, 67). And this three-tiered perspective amounts to a relation between mechanistic explanations that are specified at two levels: a higher-level explanation for S's ψ-ing specified in terms of Xs and ϕ-ings, and a lower-level explanation for X's ϕ-ing specified in terms of Ps and σ-ings.[8] Craver submits that by these three descriptive perspectives on an X's ϕ-ing, one can integrate mechanistic explanations between lower and higher levels.[9] This integration goes as follows. A contextual description specifies the contribution – the role function – of X's ϕ-ing to S's ψ-ing. For instance, say, the role function ii of 'pumping blood through the circulatory system'. Disconnected from the mechanistic context in which X's ϕ-ing is embedded, one describes X's ϕ-ing in isolation. In this case, 'contracting' (iv). And this isolated description of X's ϕ-ing constitutes the explanandum for a lower-level explanation in terms of σ-ings and Ps – (organized) contractions of heart muscles. Thereby, the lower-

---

8 Craver (2001, 62-65) switches between an analysis of levels in terms of a mereological part-whole relationship between S, X, and P, resulting in three levels, and an analysis in terms of mechanistic explanations, resulting in two levels at which explanations are given. I consider the latter analysis in this paper. For an in-depth (and novel) treatment of mereological part-whole relationships, see (Craver: 2007).

9 Craver (2001) considers determination of the precise number of levels and their individuation an empirical affair. See also Machamer et al. (2000).

level explanation is integrated into the higher-level one. The distinction between contextual and isolated descriptions of an X's ϕ-ing is thus pivotal in Craver's framework for inter-level integration of mechanistic explanations. This integration scheme in terms of a three-tiered perspective on X's ϕ-ing is depicted graphically (based on Craver: 2001) in Figure 1.

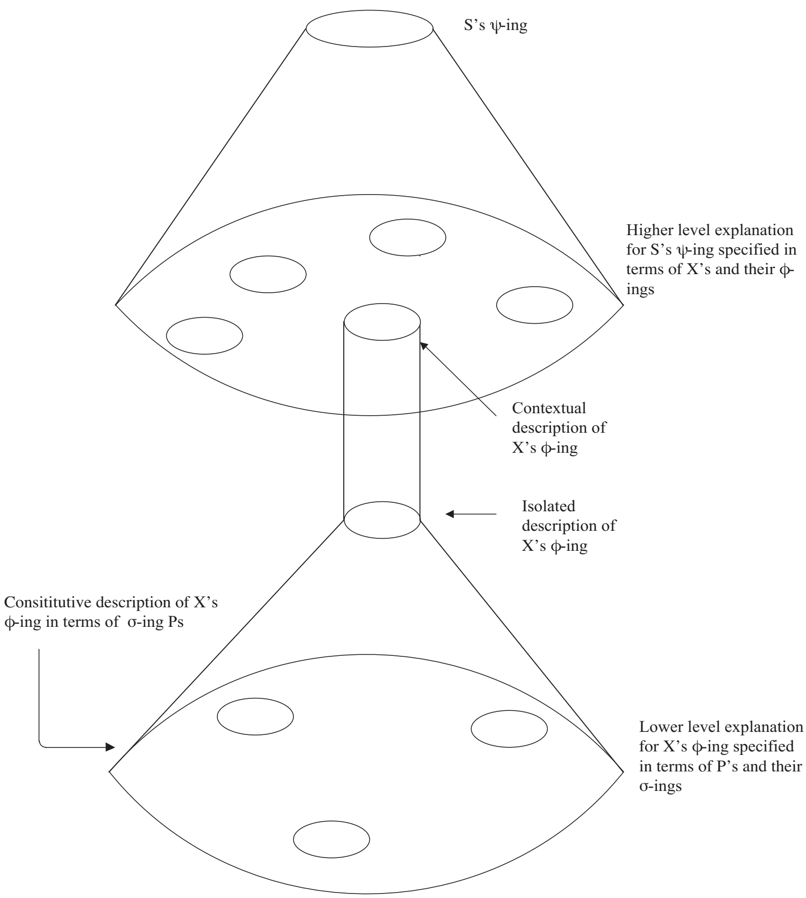

Figure 1: Contextual, isolated, and constitutive perspectives on an X's ϕ-ing.

8.2.4. Summary

To summarize, key to Craver's treatment of role function is restricting this concept to the concept of mechanism, and taking the details of mechanistic organization into account in the ascription and description of role functions. The pay-offs advanced by Craver of this strategy are the following. One, the ascription and description of role functions delivers a more precise image than Cummins' account of functions gives: whereas Craver asserts that in Cummins' picture one cannot distinguish interesting from uninteresting analytic explanations, his treatment of role functions is presented as supporting analytical, i.e., mechanistic, explanations that are explanatorily adequate. That is, his concept of role function explicates how Xs and their ϕ-ings precisely contribute to an S's ψ-ing. Two, his treatment of role functions supports the inter-level integration of mechanistic explanations. In the remainder of this paper I will consider these two claims in more detail, viz.:

CL1. In the context of the restriction to mechanism (and mechanistic explanation), Craver's image on role functions is precise enough to support explanatorily adequate mechanistic explanations.

CL2. Craver's treatment of role functions supports the inter-level integration of mechanistic explanations.

8.3. Evaluating Craver's Concept of Role Function

In this section I will argue that Craver's treatment of role functions, interesting though it is, lacks precision to the effect that:

A. It leads to mechanistic schemes that contain non-explanatory elements.
B. It leaves mechanistic explanations explanatorily incomplete.
C. It fails to integrate mechanistic explanations between levels.

I develop claim A in section 3.1, claim B in section 3.2 and claim C in section 3.3. I develop these claims in terms of a detailed analysis of Craver's showcase of the functions of the heart in the circulatory system. These claims are based upon a critical assessment of Craver's view on the use of organization-characteristics of mechanisms in the ascription and description of role functions. Based on this analysis, I spell out desiderata that Craver's role function concept must meet in order to address these challenges. I also sketch a solution to meet these desiderata in the remainder of the paper.

8.3.1. Challenge 1: non-explanatory schemes

In his 2001 account, Craver conflates isolated descriptions of an X's ϕ-ing with (contextual) role function descriptions of an X's ϕ-ing, which results in conceptual problems. Consider that, in explicating the distinction between isolated descriptions and contextual or role function descriptions of an X's ϕ-ing, Craver asserts the following:

> The relevant interfaces between the heart and the other components of the circulatory system lie at the spatial boundaries of the heart with the incoming veins, the outgoing arteries, and the blood coursing through them. *Described in isolation from its context at those interfaces, the contribution of the heart to the circulatory system is to contract; this is what it does (alone) that (in the right context) contributes to the expulsion of blood, the circulation of blood, and the distribution of oxygen and calories* (Craver: 2001, 64, italics added)

In light of Craver's distinction between role function descriptions – which capture contributions of an X's ϕ-ing to an S's ψ-ing – and (isolated) descriptions of activities that fill these role functions, the passage italicized above is confusing. It suggests that an isolated description of an X's ϕ-ing ('contracting') in itself expresses a contribution of X's ϕ-ing to S's ψ-ing, as well as to the (contextual) role functions of X's ϕ-ing. That is, rather than describing the activity that fills these role functions, Craver here seems to take 'contracting' as a description that in itself expresses a contribution to S's ψ-ing and to the role functions of 'expelling blood' (iii), 'pumping blood through the circulatory system' (ii) and to 'distributing oxygen and calories to the body' (i). Now, to be sure, the idea that an isolated description of X's ϕ-ing – 'contracting' – expresses a contribution to S's ψ-ing and/or to role functions of X's ϕ-ing is very different from the idea that the role functions (iii), (ii) and (i) of X's ϕ-ing express contributions to S's ψ-ing. In the former case, a description in which no reference is made to organizational features of a mechanism is taken to express a contribution to S's ψ-ing and/or to role functions of X's ϕ-ing (cf. section 6.2). Whereas in the latter case descriptions that do refer to such organizational features are taken to express contributions to S's ψ-ing (cf. section 6.2).

At first glance, one might perhaps be inclined to interpret Craver's conflation of isolated and (contextual) role function descriptions of an X's ϕ-ing as merely a lapse of expression on his part; what he really intended to say was that 'contracting' contributes to S's ψ-ing (delivering goods to tissues) in the sense of being the activity that fills the role functions (iii), (ii) and (i). However, contrary to this interpretation, in his discussion of the role functions of the heart's ϕ-ing (in which the above-quoted passage is situated) he explicitly and repeatedly (see Craver: 2001, 63-65) includes 'contracting' (iv) among the role function descriptions of the heart's ϕ-ing. The conflation of isolated and (contextual) role function descriptions of an X's ϕ-ing is, hence, not a momentary lapse of expression but, rather, a repetitive feat in Craver's account. So, by this conflation, in Craver's 2001 account the concept of an isolated description of an X's ϕ-ing – 'contracting' – is intended to do more work than specifying an explanandum activity for a lower-level explanation, and facilitating the transition to a contextual description of X's ϕ-ing that specifies the contribution – the role function – of X's ϕ-ing to S's ψ-ing. In addition to these two purposes, an isolated description of X's ϕ-ing is (1) *also taken to express a contribution of X's ϕ-ing to S's ψ-ing*; and an isolated description of X's ϕ-ing is (2) *also taken to express a contribution to the (contextual) role functions of X's ϕ-ing*, i.e., to 'expelling blood' (iii), to 'pumping blood through the circulatory system' (ii) and to 'distributing oxygen and calories to the body' (i). Craver thus attaches two additional explanatory goals to isolated descriptions of an X's ϕ-ing:

G1. Isolated descriptions of an X's ϕ-ing express contributions to an S's ψ-ing

G2. Isolated descriptions of an X's ϕ-ing express contributions to (contextual) role functions of that X's ϕ-ing

I will argue that these two additional explanatory goals cannot be achieved with Craver's concept of an isolated description of an X's ϕ-ing. I submit that, in effect, this leaves mechanistic schemes that purport to explain an S's ψ-ing (partially) non-explanatory.[10]

Let me start with Craver's first claim that the isolated description of 'contracting' (iv) specifies a contribution of X's ϕ-ing to S's ψ-ing. As said, in Craver's account, the ascription of a function to an X's ϕ-ing involves specifying the contribution of an X's ϕ-ing to an S's ψ-ing in terms of/by referring to how X's ϕ-ing is organized with other X's (and their ϕ-ings), which allows X's ϕ-ing to make

10 I speak of scheme rather than explanation, since the former contains in my usage of this term elements that are non-explanatory. Explanations are not supposed to contain such elements.

that contribution. And the content of a function description is fixed by a function ascription: a description of a function specifies the contribution of an X's ϕ-ing to S's ψ-ing by referring to how X's ϕ-ing is organized with other X's (and their ϕ-ings) such that X's ϕ-ing contributes to S's ψ-ing. So, *reference to relevant mechanistic context is a vital constraint in both the ascription and description of role functions.*[11] And as indicated by the role function descriptions (iii), (ii) and (i), the number of organizational features of the mechanism taken into account in the ascription and description of functions may vary. However, the isolated description 'contracting' (iv) of X's ϕ-ing does not refer, as explicitly stated by Craver himself, to any mechanistic context. As said, using this concept one draws an idealized dividing line at the spatial boundary of an X and one makes no commitments concerning the mechanistic context in which X's ϕ-ing is embedded. Yet, without reference to such context, an isolated description does not express a contribution to S's ψ-ing, i.e., does not qualify as a description of a role function. In order to capture such a contribution, the description needs to refer to mechanistic context. Yet, the necessary mechanistic context for capturing such a contribution is (explicitly) not taken into account in an isolated description of an X's ϕ-ing. An isolated description of an X's ϕ-ing – 'contracting' (iv) – therefore does not express a contribution in Craver's framework.[12] Hence, it does not deliver the explanatory goods that Craver wants from it.

Now let us consider Craver's second claim that the description 'contracting' (iv) specifies a contribution to 'expelling blood' (iii), to 'pumping blood through the circulatory system' (ii) and to 'distributing oxygen and calories to the body' (i). For the same reason as explicated in the previous paragraph we can establish that the isolated description of an X's ϕ-ing does not specify a contribution to

---

11 Craver (2001) is explicit on this constraint. For instance, describing a role function involves "describing how X is organized with the other entities in S such that it contributes to S's ψ-ing"
(61); such a description expresses "the mechanistic organization by which it [X] makes that contribution' (61); "The description includes reference not just to X (and its ϕ-ing) but also to X's place in the organization of S's ψ-ing" (63).

12 Note that the term 'contracting' can figure, of course, in both isolated and (contextual) role function descriptions of an X's ϕ-ing. However, this term is detached from any mechanistic context in the former type of description, whereas in the latter type it would be an element in a (different) description that does refer to mechanistic context.

role functions of that X's ϕ-ing.[13] Again, the isolated description 'contracting' (iv) of X's ϕ-ing does not refer to any mechanistic context, and therefore does not express a contribution to either the role functions (iii), (ii) or (i). The necessary mechanistic context for capturing such a contribution is not included nor referred to in the description. So, an isolated description of an X's ϕ-ing does not specify a contribution to other role functions of that X's ϕ-ing. Again, an isolated description of an X's ϕ-ing does not deliver the explanatory goods that Craver wants from it. We thus reach the following two conclusions with regard to Craver's account:

CO1. Isolated descriptions of an X's ϕ-ing do not express contributions to an S's ψ- ing
CO2. Isolated descriptions of an X's ϕ-ing do not express contributions to (contextual) role functions of that X's ϕ-ing

The failure to meet the aims that isolated descriptions of an X's ϕ-ing express such contributions leaves mechanistic schemes that purport to explain an S's ψ-ing in terms of contributions of ϕ-ings of Xs that are (erroneously) captured by isolated descriptions (partially) non-explanatory. Recall that a mechanistic explanation is a description of a mechanism that specifies and makes insightful how activities and entities are organized actively, spatially and temporally such that together they instantiate S's ψ-ing. And that this insight is gained by (contextual) role function ascriptions and descriptions: a role function description specifies the contribution of an X's ϕ-ing to S's ψ-ing in terms of/by referring to how X's ϕ-ing is organized with other X's (and their ϕ-ings) such that X's ϕ-ing contributes to S's ψ-ing. Yet, by conflating isolated descriptions with (contextual) role function descriptions one presupposes that isolated descriptions express such contributions of an X's ϕ-ing to S's ψ-ing, i.e., that one can understand the contributions of an X's ϕ-ing to S's ψ-ing in terms of isolated descriptions. In effect, isolated descriptions get included in mechanistic schemes *as descriptions that are taken to express contributions of an X's ϕ-ing to S's ψ-ing.* Recall Craver's assertion that an isolated description of an X's ϕ-ing ('contracting') in itself expresses a contribution of X's ϕ-ing to S's ψ-ing, as well as to the role functions

[13] For the sake of argument, I ignore here the fact that role function descriptions in Craver's framework are intended to specify contributions of an X's ϕ-ing to an S's ψ-ing.

(iii), (ii) and (i) of X's ϕ-ing. However, isolated descriptions do not express such contributions. Hence, this leads to the problem that mechanistic schemes fail to make insightful *how* Xs and their ϕ-ings contribute to an S's ψ-ing. That is, one fails to understand how an S's ψ-ing is brought about by the contributions of ϕ-ings of Xs. Returning to the heart example, the isolated description of 'contracting' does not make insightful how X's ϕ-ing contributes to S's ψ-ing, and neither does this description express a contribution to the role functions of that X's ϕ-ing.[14]

8.3.2. Challenge 2: incomplete mechanistic explanations

Perhaps sensitive to the above problem that mechanistic schemes may include non-explanatory elements when isolated and (contextual) role function descriptions are conflated, Craver displaces the notion of an isolated description of an X's ϕ-ing in his later work altogether (see Craver: 2002, 2005, 2007). In this later work he explicitly introduces the constraint that a description of an explanandum activity must be a (contextual) role function description. Let us call this constraint a "contextual description constraint".[15] For instance, in the context of discussing research into an electrophysiological phenomenon known as Long-Term Potentiation (LTP), and its relation to learning and memory processes,

---

14 Note that the inclusion of isolated descriptions in mechanistic explanations as descriptions of activities that fill roles is, of course, explanatorily relevant: they specify how role functions are implemented, i.e., they specify what *enables* an X's ϕ-ing to exercise a role function. Conflating them with role function descriptions, however, leads to the problem that one erroneously assumes that isolated descriptions express contribution(s) of an X's ϕ-ing to an S's ψ-ing, and that one can understand the contribution(s) of an X's ϕ-ing to an S's ψ-ing in terms of them. In section 3.2 I spell out in more detail what sense isolated descriptions have explanatory relevance.

15 Since Craver (2005, 2007) in his later work does not distinguish between isolated and (contextual) role function descriptions and introduces this constraint I assume that Craver does not conflate these descriptions in his later work. If he in fact does, the problems would redouble. Due to this conflation, his later work then would both face the problem of non-explanatory schemes and, due to the "contextual description constraint", also the problem of incomplete explanations (see this section for the details of this problem). I take the difference between these problems to be as follows: a non-explanatory scheme may include all the relevant isolated and contextual descriptions of Xs ϕ-ings, but the explanatory status of these isolated descriptions is gotten wrong. In contrast, an incomplete explanation is one in which isolated descriptions are left out period (as due to the "contextual description constraint").

Craver asserts that this research defined two "integrative goals" (see Craver: 2005, 390; 2007, 257) for establishing the inter-level integration of mechanistic explanations. One, an upward-looking goal of showing that an X's ϕ-ing is a relevant element in an explanation for S's ψ-ing by specifying the role function of that X's ϕ-ing in S's ψ-ing. In the LTP case, the role function(s) of inducing LTP by ϕ-ings of neurons (Xs) in the mechanism for the generating of spatial maps (ψ-ing) by the hippocampus (S). Two, a downward-looking goal of building lower-level explanations for such ϕ-ing Xs.[16] In this case, an explanation that describes and makes insightful how, among others, activating (σ-ing) NMDA receptors (Ps) contribute to X's ϕ-ing. Craver argues that these upward and downward-looking goals place constraints on explanandum activity descriptions of an X's ϕ-ing (see Craver: 2005, 390; 2007, 259). Among these constraints, Craver includes the constraint that the description of an explanandum activity must be such that it *also* describes *the role of an explanandum activity* (2005, 391) in a (still) higher-level mechanism (targeted by a higher-level mechanistic explanation).[17] Given this constraint, an explanandum activity description of an X's ϕ-ing (targeted by a lower-level explanation specified in terms of Ps and σ-ings) thus must also express a (contextual) role function of that X's ϕ-ing in S's ψ-ing (targeted by a higher-level explanation specified in terms of Xs and ϕ-ings). Craver is explicit on this "contextual description constraint" on several occasions. For instance, in discussing the merits of (contextual) role function conceptualizations over the adaptive view on functions regarding explanandum activity descriptions of an X's ϕ-ing, he (again) asserts that "*one needs the more limited sense of role-functions*" (see Craver: 2007, footnote 10, 124-125, italics added; cf. Wright: 1973; cf. Mundale and Bechtel: 1996).[18]

16 Craver (2005, 2007) discusses these integrative goals in the context of advancing an alternative model for interfield integration to reduction models.

17 Such an upward-looking constraint seems sensible, given his aspirations to advance a competitor to
reduction models of interfield integration.

18 Given this "contextual description constraint" we can (with hindsight) understand why Craver conflates (contextual) role function descriptions with isolated descriptions of an X's ϕ-ing in his 2001 account. If Craver in this work is already concerned with the constraint that a description of an explanandum activity should be such that that it *also* expresses a role function of that activity in a higher-level mechanism, one can understand why Craver (2001) interprets isolated descriptions as descriptions of role functions, i.e., takes an isolated description of an X's ϕ-ing as expressing a contribution to an S's ψ-ing and to (contextual) role functions of that X's ϕ-ing.

However, due to this "contextual description constraint" a different problem now emerges for Craver's later account (2002, 2005, 2007). This constraint namely *precludes* the inclusion of isolated descriptions of an X's ϕ-ing in mechanistic explanations, since these do not express contributions of an X's ϕ-ing to an S's ψ-ing. By this displacement of isolated descriptions, however, one loses one of the "distinct perspectives" (see Craver: 2001, 67) on an X's ϕ-ing in a mechanism. Loss of this distinctive perspective leads to the problem that one does not explicate how the ϕ-ings of Xs can have, play or implement their role functions. That is, without descriptions of such details – ϕ-ings of Xs described in isolation – one does not explicate what allows ϕ-ings of Xs (how ϕ-ings of Xs are able) to exercise their role functions in S's ψ-ing. Returning to the heart example, the isolated description 'contracting' of X's ϕ-ing cannot (in Craver's later account) be included in the mechanistic explanation for S's ψ-ing. Hence, one cannot explicate that 'contracting' is the activity underlying/enabling the role functions of 'expelling blood' (iii), 'pumping blood through the circulatory system' (ii) and 'distributing oxygen and calories to the body' (i). However, such displacements of isolated descriptions leave mechanistic explanations incomplete, since it cannot be explicated what *enables* ϕ-ings of Xs to exercise their role function in/contribute to S's ψ-ing. One thus fails to fully understand how S's ψ-ing is brought about. By this loss of information, Craver hence fails to meet what he coins the "core normative requirement on mechanistic explanations" (see Craver: 2006, 367; 2007, 122): that they must account fully for an S's ψ-ing.[19] [20]

19 By losing the distinction between a role function and an activity that fills it, Craver also discards one of the distinctive selling points by which he aims to refine Cummins's concept of function, to wit: solving the ambiguous character that he takes to surround Cummins's concept of function (see section 6.2).

20 Craver's (2007) goal in advancing this and other requirements is to provide an alternative mechanistic "regulative ideal" on explanation to classical reduction (e.g. Nagel: 1961). Well-known problems with the classical model of reduction, such as the impossibility to formulate cross-theoretical identity statements between terms of a reduced and reducing theory when the reduced theory is (partially) false, led more recent reductive offshoots (e.g. see Bickle: 1998, 2003) to abandon this cross-theoretical identity condition, replacing it with weaker cross-theory relations such as "ontological reductive links" (see Bickle: 1998). According to Craver (2007) such moves have a price: by abandoning the classical model, one also abandons the covering-law model of explanation that constitutes its core and, in Craver's view, there is no alternative "regulative ideal" on explanation available to take its place. Craver's advancement of requirements or norms of mechanistic explanations is aimed to fill that gap. I do not elaborate on the relationship (and subtleties involved) between mechanistic explanation and reduction, classical

8.3.3. Challenge 3: inter-level separation of mechanistic explanations

Both the conflation of isolated and (contextual) role function descriptions of an X's ϕ-ing in Craver's earlier 2001 account and the displacement of isolated descriptions in his later work (2002, 2005, 2007) have further repercussions, landing his 2001 strategy for the inter-level integration of mechanistic explanations in trouble.

Recall, this strategy aims to integrate a higher-level explanation for an S's ψ-ing, specified in terms of Xs and ϕ-ings, with a lower-level explanation for an X's ϕ-ing, specified in terms of Ps and σ-ings. As we saw, in the heart example, Craver (2001) specifies this integration as follows. A contextual description specifies the contribution – in terms of one of the role functions (iii), (ii) or (i) – of X's ϕ-ing to S's ψ-ing. Disconnected from the mechanistic context in which X's ϕ-ing is embedded, X's ϕ-ing is described in isolation as 'contracting' (iv). And this isolated description of X's ϕ-ing characterizes the explanandum for a lower-level explanation, specified in terms of (organized) σ-ings and Ps – (organized) contractions of heart muscles. In this scheme, the link between lower-level and higher-level explanations is thus established by taking contextual (role function) descriptions as expressing the *contribution* of X's ϕ-ing described in isolation – 'contracting' – to S's ψ-ing. This link between lower-level and higher-level explanations is, however, severed when isolated descriptions of an X's ϕ-ing are conflated with (contextual) role function descriptions of an X's ϕ-ing. The integration scheme then changes into one in which an isolated description of X's ϕ-ing – 'contracting' – is presupposed to express a contribution to S's ψ-ing (and/or to the (contextual) role functions (iii), (ii) and (i) of X's ϕ-ing). This is not the case, however, for as I have argued isolated descriptions do not specify such contributions. By implication, the isolated description of X's ϕ-ing is then not an element of the explanans of the higher-level explanation for S's ψ-ing, and thus not related to the higher-level explanation. And since the isolated description of X's ϕ-ing characterizes the explanandum for a lower-level explanation, the explanandum of the lower-level explanation is unrelated to (the explanans of) the higher-level explanation. This separates lower-level and higher-level explanations.

---

or otherwise, in this paper. For a defense of a non-reductive position – in line with Craver's position – see (Van Eck et al.: 2006).

The displacement of isolated descriptions due the "contextual description constraint" in Craver's more recent work (2002, 2005, 2007) is an unsatisfactorily solution to this problem, despite the fact that it conceptually restores the inter-level integration of mechanistic explanations: if a description of an explanandum activity of X's ϕ-ing should be a role function description, also expressing a contribution of X's ϕ-ing to S's ψ-ing, the explanandum of the lower-level explanation – X's ϕ-ing described contextually – simultaneously is an element of the explanans of the higher-level explanation. However, such a scheme establishes integration between incomplete explanations in which isolated descriptions of ϕ-ings of Xs (descriptions of activities that fill roles) are discarded. Relevant details in the link between higher-level and lower-level explanations are omitted. In the heart example, one then moves from a lower-level explanation in terms of (organized) contractions (σ-ings) of heart muscles (Ps) directly to an X's ϕ-ing described contextually – say, to the role function (ii) of 'pumping blood through the circulatory system – that is an element in the higher-level explanation for S's ψ-ing. Yet, how this role function of X's ϕ-ing is implemented by 'contracting' – X's ϕ-ing described in isolation – is not explicated, and neither how 'contracting' is instantiated by, among others, (organized) contractions (σ-ings) of heart muscles (Ps). Such integration thus loses track of relevant mechanistic details.

8.3.4. Desiderata

To sum up, a solution for addressing the problems of non-explanatory schemes, incomplete explanations and inter-level separation should be able to do the following:

D1. It should distinguish isolated descriptions from (contextual) role function descriptions of an X's ϕ-ing
D2. It should include isolated descriptions of an X's ϕ-ing in mechanistic explanations

I address these desiderata in section 5. In the next section, I first consider in more detail how the role functions (iii), (ii) and (i) of X's ϕ-ing relate to one another (and to S's ψ-ing). This feat is not discussed in Craver's account (2001). This analysis suggests that a mechanistic explanation gains explanatory leverage when contributions between different role functions of an X's ϕ-ing are explicated. I suggest a strategy to explicate such contributions in section 5.

8.4. Functional Hierarchies of Role Functions

Given Craver's description of different (contextual) role functions of the same X's ϕ-ing – the role functions (iii), (ii) and (i) – the question manifests itself how these role functions themselves relate to one another (and to S's ψ-ing). I address this question below.

8.4.1. Functional hierarchies

The notion of mechanistic organization – mechanistic context – is crucial for understanding S's ψ-ing. Different (contextual) role functions of X's ϕ-ing are ascribed and defined in reference to a varying number of organizational features (contexts) of the mechanism. Hence, understanding the relationships between these contexts and on the basis thereof between these (contextual) role functions is crucial for understanding the mechanistic organization by which S engages in ψ-ing. The explanatory leverage of role function ascriptions and descriptions is that they explicate the relationships between an X's ϕ-ing and an S's ψ-ing in terms of how X's ϕ-ing, given its place in a mechanistic organization, contributes to S's ψ-ing (see Craver: 2001). Therefore, in the context of different role functions of the same X's ϕ-ing, understanding their relationships amounts to explicating how they contribute to one another (and to S's ψ-ing). In the example under scrutiny, this entails explicating how 'expelling blood' (iii) contributes to 'pumping blood through the circulatory system' (ii), how 'pumping blood...' (ii) contributes to 'distributing oxygen and calories to the body' (i) and how in turn 'distributing...' (i) contributes to 'delivering goods to tissues' (ψ-ing).[21] Let us call these contributory relationships between 'iii-ii-i-ψ', a *functional hierarchy*.[22]

---

[21] This desideratum finds support in, for instance, cardiac physiology. Specifications of contributions between different functions of the same item can be found in cardiac research on Coronary Circulation, a system consisting of blood vessels that has the function to continually supply blood to cardiac tissue of the heart. Kassab (2005), for instance, states that in the proximal portion of the intramyocardial (IMCA)- coronary arteries, the function 'conducting blood' contributes to the function 'delivering blood', and that in the distal portion of the IMCA-coronary arteries the function 'transporting blood' contributes to the function 'transporting

Explicating such a functional hierarchy can be done in Craver's framework, but only to a certain level of detail. Specifically, in the role function descriptions that Craver considers the mechanistic context to capture the contributions of role functions to one another in full detail is underspecified. Consider the relationship between the role function descriptions of 'expelling blood' (iii) and 'pumping blood through the circulatory system' (ii). The description 'expelling blood' (iii) refers to the mechanistic context of (the availability of) blood (see Craver: 2001, 64). This mechanistic context referred to in the ascription and included in the description 'expelling blood' (iii), however, is lacking in detail to explicate by which mechanistic organization blood expulsion precisely contributes to the pumping of blood through the circulatory system. In order to capture that contribution more precisely, additional mechanistic context needs to be referred to and included in the description. It needs reference, for instance, to the relevant organization of arteries, which allow the expulsion of blood to contribute to the circulation of blood through the circulatory system. The description of 'expelling blood' (iii) however lacks reference to – makes no commitments to – that context. The contribution of the role function 'pumping blood through the circulatory system' (ii) to the role function 'distributing oxygen and calories to the body (i) cannot be fully explicated for the same reason. The description 'pumping blood through the circulatory system' refers to the mechanistic context of (the availability of) blood and the relevant organization of veins and arteries (see Craver: 2001, 64). This mechanistic context referred to in the ascription and included in the description (ii), is, again, lacking in detail to explicate by which mechanistic organization the pumping of blood specifically contributes to the distribution of oxygen and calories. In order to capture that contribution more precisely, the description, again, needs to refer to additional mechanistic context. For instance, to oxygen and calories and (organized) capillaries that allows the pumping of blood to contribute to oxygen and nutri-

---

oxygen and nutrients' (blood transport is demarcated from blood conduction in terms of flow conductivity: a slower flow conductivity is needed in smaller IMCA vessels – described by the function 'transporting blood' – in order to ensure sufficient transmission of oxygen and nutrients to the environment (see Kassab: 2005).

22 The sense of hierarchy I have in mind is not a hierarchy of mereological levels or of levels of explanation (cf. footnote 8), but refers to the taking into account of an increasing number of organizational features of the (same) mechanism in explicating the contributory relationships from iii-to-ii, etc. Spelling out the contributory relationships between 'iii-ii-i-ψ' is set within specifying a mechanistic explanation for an S's ψ-ing.

ents distribution. The description of 'pumping blood through the circulatory system' (ii) however lacks reference to – makes no commitments to – that context. This leads to a third conclusion:

CO3. Contributions between different role functions of an X's ϕ-ing – a functional hierarchy – are not explicated in full detail.[23]

8.4.2. Mechanistic organization to the rescue?

Is there a route available within Craver's framework that makes it possible to avoid conclusion 3? One can imagine the maneuver to explicate the contributory relationships between 'iii-ii-i-ψ' by changing the entities to which these descriptions apply. Taking, say, the role function of the 'heart' (P) as 'expelling blood', one may explicate 'expelling blood' as a contribution of P to yet another entity's role function, say, 'pumping blood' of the 'heart plus its incoming veins and outgoing arteries' (X). And the role function of 'pumping blood' by X can be explicated further still as a contribution to, again, another entity's role function, say, 'oxygen and calories distribution' of the circulatory system (S). This maneuver, however, does not deliver the goods. By taking this maneuver it becomes indeterminate how role functions are to be ascribed to entities: mechanistically relevant entities *are no longer available as elements of relevant mechanistic context* to specify and ascribe role functions. Consider, say, the role function specified as 'pumping blood', ascribed to the 'heart plus its incoming veins and outgoing arteries' (X). Since the entity to which this role function is ascribed is specified as the 'heart plus its incoming veins and outgoing arteries', those veins and arteries are not separate entities themselves outside the heart that can be taken into account in ascribing the role function of 'pumping blood' to X. These entities are no longer part of the organizational context that can be employed for this role function ascription. But since these incoming veins and outgoing arteries cannot be taken into account in specifying the role function of 'pumping blood' to the heart plus its incoming veins and outgoing arteries (X), it becomes indeterminate how 'pumping blood' can be ascribed as a role function to X in the first place. This deflates the whole idea of function ascription in terms of

[23] Notice that the same problems occur when role functions (iii), (ii) and (i) would relate to different ϕ-ings of the same X (in the context of the same S's ψ-ing).

relevant (details of) mechanistic organization, for there would be no relevant context to refer to. Based on the foregoing we can specify a third desideratum:

D3. To capture in detail the contributions between different role functions of an X's ϕ-ing

8.5. Capturing functional hierarchies of role functions

One (obvious) solution to solve the problems of non-explanatory schemes, incomplete explanations and inter-level separation emerges straightforwardly: reject the "contextual description constraint", thereby making it possible to include isolated descriptions of Xs ϕ-ings, alongside (contextual) role function descriptions, in mechanistic explanations, and clearly distinguish between these descriptions (and their explanatory status). A solution to the above problems without rejecting the "contextual description constraint" seems not in view, for how then to include isolated descriptions of Xs ϕ-ings in mechanistic explanations without violating this constraint? A solution to capture functional hierarchies of role functions with increased precision requires more spelling out. In the remainder of the paper I outline a strategy for doing so. Recall, key to the problem of spelling out functional hierarchies of role functions in more detail is that the mechanistic context referred to in the ascription and included in the description of role functions lacks the detail to explicate by which mechanistic context different role functions of an X's ϕ-ing precisely contribute to one another. Therefore, in order to explicate these contributions in more detail, one needs to somehow supplement the relevant mechanistic context to these role function descriptions. In the following I outline a possible way to do so, borrowing concepts from an approach towards the modeling of functions of technical systems, developed in the engineering sciences-domain. The functional modeling branch of engineering science-research consists of a plethora of methods for the modeling of technical functions that are put to use in a variety of engineering settings, such as engineering designing, reverse engineering and failure analysis of technical systems.[24] One such approach models technical functions in terms of concepts that are suited to address our challenge at hand.

---

[24] For the interested reader, an overview of engineering approaches toward functional modeling can be found in Erden et al. (2008).

8.5.1. Trigger-Effect functions

The Functional Interpretation Language (FIL) is an approach to functional modeling that is primarily aimed at supporting failure analysis tasks, mainly in electro-mechanical systems (see Bell et al.: 2007). In this approach, a function is represented in terms of four elements: a "description" of the function, the "purpose" of the function, the "trigger" of the function and the "effect" of the function. For instance, the function of a torch is represented in FIL as follows: the function of the torch is described as "torch lit". This function, if the torch is functioning as expected, achieves the purpose of "cast light". Achievement of this purpose by the function is represented in terms of a trigger-effect pair associated with the torches' function: the trigger "switch on" achieves the effect of "lamp on". Triggers and effects in FIL describe the boundaries of a technical system or technical component that is functionally represented – in case of the torch its switch and its lamp – and are intended as labels that allow linking the cause of a failure (in the context of a failure analysis) to relevant properties of the behaviors of a technical system. For instance, if the switch position is "on" and the effect of a lit lamp is absent, the function of the torch has failed. This trigger-effect relation allows tracing the cause of the failure to relevant behavioral properties, say, an electrical short circuit.[25] More complex technical systems are modeled in FIL by several functions (which are connected by their triggers and/or effects).

8.5.2. A Functional hierarchy between role functions of the heart

The trigger and effect elements of functional representations in FIL provide concepts to capture functional hierarchies in more detail, i.e., by employing triggers and effects one can add precision to the explication of contributions between different role functions of an X's ϕ-ing. The conceptual reasoning to establish these feats is as follows. The relevant mechanistic context needed to add detail to these contributions can be expressed in terms of triggers and effects. These trigger and effect descriptions can then be supplemented to (contextual) role function descriptions of an X's ϕ-ing. Reference to relevant

[25] Alternatively, if the switch position is 'off' and nevertheless the lamp remains lit, this trigger-effect relation allows linking to other relevant behavioral properties, say, the mechanical-electrical connection between the switch and the electrical circuit of the torch. For all the conceptual machinery on fault analysis with FIL, see (Bell et al., 2007).

mechanistic context is accommodated by such supplementing, which allows one to explicate contributions between different role functions of an X's ϕ-ing with increased precision. This "supplementing strategy" goes as follows: by letting the effect supplemented to one role function correspond to the trigger of the role function to which it contributes, one can specify the contribution of the former to the latter.

Consider different role functions of an X's ϕ-ing. We supplement the trigger "blood available" and the effect "blood in arteries" to the role function 'expelling blood' (iii). The effect "blood in arteries" that is supplemented to the role function 'expelling blood' (iii) provides the triggering condition for the role function 'pumping blood through the circulatory system' (ii). The contribution of the role function 'expelling blood' to "pumping blood..." thus consists in providing the arteries with blood, allowing it to be pumped around. The effect "blood in arteries and capillaries" supplemented to the role function 'pumping blood through the circulatory system' (ii), in turn, provides the triggering condition for the role function 'distributing oxygen and calories to the body' (i), which has the effect "oxygen and calories distributed" supplemented to its description. This effect, in turn, contributes to S's ψ-ing of 'delivering goods to tissues'. In this fashion, the relevant mechanistic context needed to specify a contribution more fully can be expressed in terms of triggers and effects, which are supplemented to descriptions of role functions. Applied to the heart example, this proposal is depicted graphically in Figure 2.

I submit that this is a conceptually feasible way to specify contributions between different role functions of an X's ϕ-ing with additional detail, i.e., in the case considered the functional hierarchy between 'iii-ii-i-ψ'. This proposal extends the notion of mechanistic organization by introducing an organization between role functions in term of the functional input (trigger) and output (effect) by which they are connected.[26]

[26] This extension does not replace the notion of mechanistic organization. Au contraire, it explicitly takes organizational features of a mechanism into account in specifying an organization between different role functions of an X's ϕ-ing (cf. note 23). To my knowledge, capturing an organization between different role functions of an X's ϕ-ing has not been detailed in the literature on mechanistic explanations.

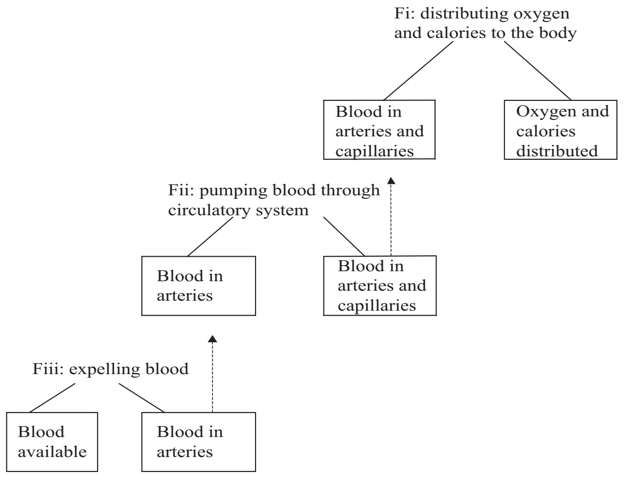

Figure 2: Functional hierarchy of role functions of the heart.

8.6. Concluding Remarks

In this paper I have analyzed the concept of role function in Craver's account of mechanistic explanations. Based on a detailed analysis of Craver's showcase example of the functions of the heart in the circulatory system, I have argued that Craver's earlier account conflates descriptions of activities that fill roles with descriptions of role functions. It is shown that this conflation leads to two problems. Firstly, mechanistic schemes purported to explain an explanandum activity contain non-explanatory elements. Secondly, it separates mechanistic explanations between levels. I have further argued that Craver's later account displaces descriptions of activities that fill roles altogether, leading to the problem that mechanistic explanations are left incomplete. I suggested a solution to avoid these problems. I ended the paper with a proposal to capture contributions between different role functions of the same item, a feat that is not discussed in Craver's account. I developed this proposal by employing concepts from an approach towards the modeling of technical functions, developed in the engineering sciences-domain. My proposal extends the notion of mechanistic organization by introducing an organization between role functions and does so in a way that is firmly grounded in the details of a mechanism's organization.

Acknowledgements

Many thanks to the following people for their very helpful comments and suggestions: Jelle de Boer, Maarten Franssen, Peter Kroes, Huib Looren de Jong, Phyllis McKay-Illari, Maurice Schouten, and Pieter Vermaas. I also kindly appreciate the comments from the audience at the Mechanisms and Causality in the Sciences Conference, 9-11 September 2009, Canterbury UK, in particular from Carl Craver.

References

Bechtel, W. and Richardson, R.C. (1993) *Discovering Complexity: Decomposition and Localization as Strategies in Scientific Research*, Princeton University Press, Princeton NJ.

Bechtel, W. and Abrahamson, A. (2005) 'Explanation: A Mechanist Alternative', in: *Studies in History and Philosophy of Biological and Biomedical Sciences* 36: 421-441.

Bell, J., Snooke, N., and Price, C. (2007) 'A Language for Functional Interpretation of Model Based Simulation', in: *Advanced Engineering Informatics* 21: 398-409.

Bickle, J. (1998) *Psychoneural Reduction: The New Wave*, MIT Press, Cambridge MA.

Bickle, J. (2003) *Philosophy of Neuroscience: A Ruthlessly Reductive Approach*, Kluwer Academic Publishers, Dordrecht.

Calcott, B. (2009) 'Lineage Explanations: How Biological Mechanisms Change', in: *British Journal for the Philosophy of Science* 60: 51-78.

Craver, C.F. (2001) 'Role Functions, Mechanisms, and Hierarchy', in: *Philosophy of Science* 68: 53-74.

Craver, C.F. (2002) 'Interlevel Experiments and Multilevel Mechanisms in the Neuroscience of Memory', in: *Philosophy of Science* 69: S83-S97.

Craver, C.F. (2005) 'Beyond reduction: Mechanisms, Multifield Integration and the Unity of Neuroscience', in: *Studies in History and Philosophy of Biological and Biomedical Sciences* 36: 373-395.

Craver, C.F. (2006) 'When Mechanistic Models Explain', in: *Synthese* 153: 355-376.

Craver, C.F. (2007) *Explaining the Brain: Mechanisms and the Mosaic Unity of Neuroscience*, Oxford University Press, New York.

Craver, C. F. (2008) 'Physical Law and Mechanistic Explanation in the Hodgkin and Huxley Model of the Action Potential', in: *Philosophy of Science* 75: 1022-1033.

Cummins, R. (1975) 'Functional Analysis', in: *Journal of Philosophy* 72: 741-765.

Cummins, R. (1983) *The Nature of Psychological Explanation*, MIT Press, Cambridge MA.

Davies, P. S. (2001) *Norms of Nature: Naturalism and the Nature of Functions*, MIT Press, Cambridge MA.

Erden, M.S., Komoto, H., Van Beek, T.J., D'Amelio, V., Echavarria, E, and Tomiyama, T. (2008), 'A Review of Function Modelling: Approaches and Applications', in: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 22: 147-169.

Glennan, S. (1996) 'Mechanisms and the Nature of Causation', in: *Erkenntnis* 44: 49-71.

Glennan, S. (2002) 'Rethinking Mechanistic Explanation', in: *Philosophy of Science* 69: S342- S353.

Kassab, G.S. (2005) 'Functional Hierarchy of Coronary Circulation: Direct Evidence of a Structure-Function Relation', in: *The American Journal of Physiology - Heart and Circulatory Physiology* 289: 2559-2565.

Machamer, P.K., Darden, L., and Craver, C.F. (2000) 'Thinking about Mechanisms', in: *Philosophy of Science* 57: 1-25.

Mundale, J. and Bechtel, W. (1996 'Integrating Neuroscience, Psychology, and Evolutionary Biology through a Teleological Conception of Function', in: *Minds and Machines* 6: 481-505.

Nagel, E. (1961) *The Structure of Science: Problems in the Logic of Scientific Explanation*, Harcourt, Brace and World, Inc., New York.

Piccinini, G. (2007) 'Computing Mechanisms', in: *Philosophy of Science* 74: 501-526.

Sustar, P. (2007) 'Neo-Functional Analysis: Phylogenetical Restrictions on Causal Role Functions', in: *Philosophy of Science* 74: 601-615.

Tabery, J.G. (2004) 'Synthesizing Activities and Interactions in the Concept of a Mechanism', in: *Philosophy of Science* 71: 1-15.

Torres, P.J. (2009) 'A Modified Conception of Mechanisms', in: *Erkentniss* 71: 233-251.

Van Eck, D., Schouten, M.K.D., and Looren de Jong, H. (2006) 'Evaluating New Wave Reductionism: The Case of Vision', in: *British Journal for the Philosophy of Science* 57: 167-196.

Wimsatt, W.C. (2002) 'Functional Organization, Analogy, and Inference', in: Ariew, A., Cummins, R., and Perlman, M. (eds.*), Functions: New Essays in the Philosophy of Psychology and Biology*, Oxford University Press, New York: 173-221.

Woodward, J. (2002) 'What is a Mechanism? A Counterfactual Account', in: *Philosophy of Science* 69: S366-S377.

Wright, L. (1973) 'Functions', in: *Philosophical Review* 82: 139-168.

Wright, C. D. and Bechtel, W. (2006) 'Mechanisms and Psychological Explanation', in: Gabbay, D.M., Thagard, P., and Woods, J (eds.), *Handbook Philosophy of Psychology and Cognitive Science*, Elsevier Science, New York: 31-79.

Wright, C. D. (2007) 'Is Psychological Explanation Becoming Extinct?', in: Schouten, M.K.D., and Looren de Jong, H. (eds.), *The Matter of the Mind: Philosophical Essays on Psychology, Neuroscience, and Reduction*, Blackwell, Oxford: 249-274.

Summary

My principal goal in this thesis is to explain why different engineers use different models of functional decomposition side-by-side in engineering. Engineers often break down or decompose functions into a number of other (sub) functions, a strategy coined functional decomposition. The relationships between functions and the sets of sub functions into which they are decomposed are then often graphically represented in *functional decomposition models*. Such models come in a variety of flavors. They can, among others, represent sets of sub functions that refer to *physical behaviors*, sets of sub functions that refer to the *desired effects of physical behaviors*, and sets of sub functions that refer to the *purposes for which technical artifacts are designed* (see chapters 2, 4-6). Engineers use such models, among others, in the conceptual phase of engineering designing to specify the desired functions of some artifact-to-be, to archive and exchange design knowledge, and to identify malfunctions of artifacts.

This conceptual diversity of functional decomposition models provides a striking contrast with research in the sciences. Whereas in science debates on the adequacy of key conceptualizations seem commonplace, such debate is by and large not waged in engineering. And whereas in science such debates often lead to commonly agreed-upon key conceptualizations, such conceptual convergence seems not in view in engineering. The to engineering key concept of technical function lacks a uniform meaning but, instead, is specific to particular modeling frameworks. Whereas some engineers use this concept to refer to physical behaviors, others use it to refer to the desired effects of physical behaviors or to purposes. The majority of engineers simply accept the status quo that different engineers use different meanings of function and different models of functional decomposition side-by-side. This side-by-side usage is all the more surprising given that both function and functional decomposition play a pivotal role in engineering and that engineers readily acknowledge that this conceptual diversity leads to communication problems between collaborating engineers. The usage of different models of functional decomposition hampers the exchange and re-use of design knowledge between engineers (see chapters 6-7). Yet, despite these problems, engineers do not engage in a focused debate to arrive at a commonly shared functional conceptualization(s). The usage side-by-side of

different models of functional decomposition may thus be a phenomenon that will persist in engineering. Why do engineers use different models side-by-side? Is this usage instrumentally rational in light of the above-mentioned problems it poses to the exchange of knowledge?

In this thesis I address these questions and develop an explanation why different engineers use different models of functional decomposition side-by-side. I focus, in particular, on the (related) questions of what leads engineers to choose specific functional decomposition models and whether or not each of these models has specific advantages (see chapters 4-5). In chapters 2 and 3 I present preliminary research for this explanation. These chapters provide comparative analyses of what sorts of functional decomposition strategies and functional decomposition models are advanced in engineering.

Co-existence of different conceptualizations of scientific key concepts, which is often a temporary phenomenon, can be observed in cases where different scientists choose competing theories. This phenomenon that different scientists may choose competing theories has in the philosophy of science been extensively analyzed under the heading of *methodological incommensurability*. In this thesis I apply the analysis of scientific theory choice in terms of methodological incommensurability to engineering, rather than to science (see chapters 4-5). I show that one can then explain the co-existence of engineering models of functional decomposition.

The thesis of methodological incommensurability asserts that epistemic standards that scientists' use to evaluate and choose theories vary between rival theoretical frameworks. Such standards do not function as algorithmic rules by which one is able to determine theory choice but, rather, as *values* guiding such choices. Epistemic values refer to characteristics or properties of scientific theories that are considered desirable by scientists relative to their objectives. For instance, a scientist who considers it important that a theory is able to explain a broad range of phenomena (scope) will choose the available theory that satisfies that value best. Yet another scientist may find it important that all the processes postulated by a theory are empirically verifiable and, hence, will choose the available theory that satisfies that value best. Based on such divergent assessments of the merits of scientific theories different scientists may choose different theories: scope may favor one theory, yet empirical verifiability another.

In chapter 5 I employ and expand on the notion that one can explain divergence of theory choice in terms of variation in values. Applying this notion to engineering, I argue that engineers' choices for competing models of functional decomposition are influenced by the values that they employ in model choice. I coin these values *engineering values*. I define an engineering value as a characteristic or property of a functional decomposition model that is considered desirable by an engineer relative to an objective. I argue that depending on the engineering value(s) that an engineer considers important, he/she chooses a model that satisfies that value best. Different models satisfy different engineering values best. Hence, since engineers differ in the engineering values that they employ in model choice, different engineers choose different models. I thus offer an explanation why different models of functional decomposition are used side-by-side in engineering in terms of variation in engineering values.

I demonstrate this analysis in terms of two case studies (chapters 4-5). For instance, in the case study in chapter 5 different engineers advance different models of functional decomposition to achieve a *common* objective of routine designing. In this case engineering values vary, leading to different model choices. To give two examples, some engineers deem it important that the realization of a function by a structure does not depend on the (prior) realization of another function by another structure. Given this engineering value, I explain why these engineers choose models of sets of purpose functions, rather than models of sets of behavior functions or effect functions. Other engineers, however, consider it important that a functional decomposition defines a configuration of design solutions or structures in which all structures are compatible with one another, so that all the functions in the model are realized. Given this engineering value, I explain why models of sets of behavior functions are chosen, rather than models of effect or purpose functions.[1]

Explanations of divergence of theory choice in terms of variation in (epistemic) values spawned extensive debate in philosophy of science. One of the main challenges is to show that, despite the fact that competing theories are chosen, the rationality of scientists' choice of theories is still ensured. Advocates of value

[1] The constraint that all structures are compatible with one another (and, hence, all functions realized) is of course a crucial constraint that is valued in all functional modeling frameworks. Modeling frameworks differ, however, in which design phase this value is to be satisfied. In some accounts functional decomposition models should satisfy this value, whereas in others it should be satisfied in later design phases. Thus, only in some accounts is it a value that applies to functional decomposition models.

variation argue that the rationality of theory choice is ensured if chosen theories contribute to the attainment of scientific objectives, precisely because they satisfy the value(s) on the basis of which these theories are chosen. That is, that a chosen theory contributes to attainment of an objective that one aims to achieve with the theory precisely because it satisfies certain (desired) characteristics or properties (values). For instance, a theory that satisfies the value that it does not contain ad hoc hypotheses conduces to the scientific objective of maximizing the falsifiability of theories. Insofar as values are such appropriate means to achieve scientific objectives, maintaining variation of these values in theory choice is considered rational. Such means-end interpretations of values allow for the possibility to rationally compare the merits of competing theories (or scientific models): this theory/model is better with respect to this value, that theory/model is better with respect to that value.

In chapter 5 I address this rationality issue in the engineering functional decomposition case. I argue that particular models contribute to attainment of the objectives for which they are used because they satisfy particular engineering values, i.e., that the (desired) characteristics or properties of the chosen models are (among) the features by means of which the models contribute to attainment of the objectives for which they are used. This means-end analysis of engineering values makes it explicit that engineers' usage of competing functional decomposition models is rational from a practical point of view. I distinguish between main objectives and sub objectives of engineers to demonstrate that engineering values are appropriate means to achieve objectives: I argue that models that satisfy particular engineering values directly contribute to the achievement of particular sub objectives and that the achievement of these sub objectives, in turn, contributes to the attainment of main objectives. For instance, returning tot the routine designing case, consider the engineering value that the realization of a function by a structure does not depend on the (prior) realization of another function by another structure. I argue that if this engineering value is satisfied by a functional decomposition model (of purpose functions) the model is conducive to the sub objective of being able to consider a broad range of function-structure connections. Having a broad range of function-structure connections, in turn, is a sub objective that, when achieved, contributes to the main objective of routine design. On the other hand, I argue that a model (of behavior functions) that satisfies the engineering value that the design solutions or structures defined by the model are compatible with one another contributes to the sub objective of having all functions in the model realized.

Achievement of this sub objective, in turn, supports the main objective of routine design.

The distinction between main and sub objectives thus makes it insightful why there occurs variation in engineering values in cases where different models are in competition, i.e., where different models are used to achieve a common main objective. This analysis, in addition, makes it possible to rationally compare models with respect to a particular value. For instance, if one values compatibility of structures, then one better opts for a model of behavior functions; if one values independence of function-structure connections, one better picks a model of purpose functions. There is not one model that satisfies all such engineering values best. Hence, I submit that the usage of different functional decomposition models by different engineers is rational from a practical point of view.

In addition to explaining the co-existence of functional decomposition models, I also analyze the communication problems engendered by the co-existence of functional decomposition models (see chapters 6-7). An issue closely related to methodological incommensurability is the question whether terms from competing conceptual frameworks can be cross-translated. This issue is extensively analyzed in the philosophy of science under the heading of semantic incommensurability. In chapters 1 and 5 I argue (based on research of chapters 2-4 and 6-7) that semantic incommensurability does not apply in the functional decomposition case. I argue that there is significant overlap in the content or conceptualizations behind the functional descriptions used in different functional decomposition models. Given this overlap in content, my second goal in this thesis is to improve the communication of functional decomposition models across frameworks.

In chapters 6 and 7 I review an engineering methodology that is aimed to translate models of functional decomposition across functional modeling frameworks, and I develop conceptual improvements to the translation steps proposed in it. In this methodology models are translated in terms of one specific (overarching) conceptualization of function and functional decomposition. This overarching conceptualization is used to identify notions of function and functional decomposition models. Based on these identifications, translations of models across frameworks are developed. I argue, however, that this overarching functional conceptualization collapses distinctions between notions of function and functional decomposition models. Resultantly, in the translation process, functional information is lost. The exchange and re-use of design knowledge across modeling frameworks is thereby hampered. I argue that my regimenting of functional decomposition models allows me to highlight and bypass this translation problem.

The basic idea of my alternative translation strategy is to first translate functional decomposition models of sets of behavior functions into models of sets of physical behaviors. This step preserves behavior function information. In a second step effects of the behaviors in translated models are abstracted. These effect descriptions are used to construct functional decomposition models of sets of effect functions. In similar fashion, in a third step, purpose functions can be abstracted from effect functions and models of sets of purpose functions constructed. These second and third step relate different functional decomposition models. I argue that my translation strategy enables the exchange and re-use of design knowledge across modeling frameworks. This result achieves my second goal to improve the communication of functional decomposition models across frameworks.

Samenvatting

Het doel van dit proefschrift is te verklaren waarom verschillende ingenieurs verschillende functionele decompositie modellen naast elkaar gebruiken. Ingenieurs splitsen functies vaak op in andere (sub) functies, een strategie die functionele decompositie wordt genoemd. De relaties tussen functies en de sets van sub functies waarin ze zijn opgesplitst worden vaak grafisch weergegeven in *functionele decompositie modellen*. Er zijn verschillende soorten modellen. Zo zijn er, onder andere, modellen waarin de sub functies verwijzen naar fysische gedragingen, modellen waarin de sub functies verwijzen naar de gewenste effecten van fysische gedragingen en modellen waarin de sub functies verwijzen naar de doelen waarvoor technische artefacten zijn of worden ontworpen (zie hoofdstukken 2, 4-6). Ingenieurs gebruiken zulke modellen, onder andere, in de conceptuele ontwerpfase voor het specificeren van de gewenste functies van het te ontwerpen technische artefact, voor het archiveren en uitwisselen van ontwerpkennis en voor het identificeren van malfuncties van artefacten.

Deze conceptuele verscheidenheid van functionele decompositie modellen contrasteert met onderzoek in de wetenschappen. Terwijl in de wetenschappen frequent debatten worden gevoerd over de adequaatheid van kern begrippen, worden zulke debatten in engineering over het algemeen niet gevoerd. Bovendien leiden zulke debatten in de wetenschappen vaak tot gemeenschappelijk gedeelde kernbegrippen, terwijl in engineering zo een conceptuele convergentie ver te zoeken is. Het kernbegrip technische functie kent in engineering geen uniforme betekenis, maar heeft in verschillende modelleer benaderingen een verschillende betekenis. Terwijl sommige ingenieurs het functie begrip gebruiken om naar fysische gedragingen te verwijzen, gebruiken anderen het om naar de gewenste effecten van gedragingen of naar doelen te verwijzen. Het merendeel der ingenieurs simpelweg accepteert deze status-quo dat verschillende ingenieurs verschillende betekenissen toekennen aan het functie begrip en verschillende functionele decompositie modellen naast elkaar gebruiken. Dit naast elkaar gebruiken van verschillende modellen is des te opmerkelijk, gegeven dat zowel functie en functionele decompositie een zeer belangrijke rol vervult in engineering en dat ingenieurs erkennen dat deze conceptuele verscheidenheid communicatie problemen tussen (samenwerkende) ingenieurs in de hand werkt. Het gebruik van verschillende functionele decompositie modellen bemoeilijkt het uitwisselen en hergebruiken van ontwerpkennis tussen ingenieurs (zie hoofdstukken 6-7). Ondanks deze problemen voeren ingenieurs, echter, geen debatten die zijn toegespitst op het komen tot een gemeenschappelijk gedeelde functionele conceptualisatie. Het naast elkaar bestaan van verschillende functionele decompositie modellen is derhalve een fenomeen dat lijkt te blijven bestaan in engineering. Waarom gebruiken ingenieurs verschillende modellen naast elkaar? Is dit gebruik rationeel, bezien in het licht van de hierboven genoemde problemen met betrekking tot het uitwisselen van kennis?

In dit boek adresseer ik deze vragen en ontwikkel een verklaring voor het fenomeen dat verschillende ingenieurs verschillende functionele decompositie modellen naast elkaar gebruiken. Ik concentreer me in het bijzonder op de (gerelateerde) vragen (i) op basis waarvan ingenieurs specifieke functionele decompositie modellen kiezen en (ii) of elk van deze modellen specifieke voordelen hebben (zie hoofdstukken 4-5). In hoofdstuk 2 en 3 presenteer ik voorbereidend onderzoek voor deze verklaring. Deze hoofdstukken bevatten vergelijkende analyses naar functionele decompositie strategieën en functionele decompositie modellen die beschreven worden in engineering.

Het naast elkaar bestaan van verschillende conceptualiseringen van wetenschappelijke kernbegrippen, wat vaak een fenomeen van tijdelijke aard is, kan men observeren in gevallen waar verschillende wetenschappers concurrerende theorieën kiezen. Dit fenomeen dat verschillende wetenschappers concurrerende theorieën kiezen is in de wetenschapsfilosofie uitvoering bestudeerd onder de noemer *methodologische incommensurabiliteit*. In dit proefschrift pas ik de analyse van wetenschappelijke theoriekeuze in termen van methodologische incommensurabiliteit toe op engineering, in plaats van op wetenschap (zie hoofdstukken 4-5). Ik laat zien dat op basis hiervan men het naast elkaar bestaan van verschillende modellen van functionele decompositie in engineering kan verklaren.

Methodologische incommensurabiliteit houdt in dat epistemische standaarden die wetenschappers hanteren om theorieën te evalueren en kiezen variëren tussen rivaliserende theoretische raamwerken. Zulke standaarden fungeren niet als algoritmische regels die theoriekeuzen vastleggen maar, in plaats hiervan, als waarden die zulke keuzen beïnvloeden. Epistemische waarden verwijzen naar kenmerken of eigenschappen van wetenschappelijke theorieën die wetenschappers wenselijk achtten gegeven hun doelen. Een wetenschapper die het, bijvoorbeeld, belangrijk acht dat een theorie in staat stelt tot het verklaren van een breed scala aan fenomenen (reikwijdte) zal die theorie kiezen die het beste voldoet aan deze waarde. Een andere wetenschapper kan het daarentegen belangrijk(er) vinden dat alle processen gepostuleerd door een theorie empirisch verifieerbaar zijn en derhalve kiezen voor de beschikbare theorie die het beste voldoet aan deze waarde. Gebaseerd op zulke uiteenlopende beoordelingen van de merites van wetenschappelijke theorieën kunnen verschillende wetenschappers kiezen voor concurrerende theorieën: het kan zo zijn dat de ene theorie het beste voldoet aan reikwijdte, terwijl een andere theorie het beste voldoet aan empirische verifieerbaarheid.

In hoofdstuk 5 gebruik ik de notie dat men uiteenlopende theoriekeuzen kan verklaren in termen van variatie in waarden, en werk deze notie verder uit. Ik pas deze notie toe op engineering en beargumenteer dat de keuzen van ingenieurs voor verschillende en concurrerende functionele decompositie modellen beïnvloed zijn door de waarden die ze hanteren in het kiezen van modellen. Ik noem deze waarden *engineering waarden*. Ik definieer een engineering waarde als een kenmerk of eigenschap van een functioneel decompositie model die ingenieurs wenselijk achtten gegeven hun doelen. Ik beargumenteer dat, afhankelijk van de engineering waarde(n) die een ingenieur belangrijk acht, hij/zij een model kiest dat het beste voldoet aan die waarde. Verschillende modellen voldoen het beste aan verschillende engineering waarden. Derhalve, gegeven dat ingenieurs verschillen in de engineering waarden de ze gebruiken in het kiezen van modellen, kiezen verschillende ingenieurs verschillende modellen. Op basis van variatie in engineering waarden verklaar ik aldus waarom verschillende functionele decompositie modellen naast elkaar worden gebruikt in engineering.

Ik demonstreer deze analyse middels twee case studies (hoofdstukken 4-5). In de case study in hoofdstuk 5, bijvoorbeeld, gebruiken verschillende ingenieurs verschillende functionele decompositie modellen voor een gemeenschappelijk doel van routinematig ontwerpen. In deze case is er variatie in engineering waarden, hetgeen leidt tot verschillende model keuzen. Om twee voorbeelden te geven, sommige ingenieurs achten het belangrijk dat de realisatie van een functie door een structuur niet afhankelijk is van de (eerdere) realisatie van een andere functie door een andere structuur. Gegeven deze engineering waarde verklaar ik waarom deze ingenieurs modellen van sets van doel functies kiezen (in plaats van modellen van sets van gedrag functies of effect functies). Andere ingenieurs, daarentegen, achten het belangrijk dat een functioneel decompositie model een configuratie van ontwerp oplossingen of structuren definieert waarin alle structuren compatibel zijn met elkaar zodat alle functies in het model te realiseren zijn. Gegeven deze engineering waarde verklaar ik waarom modellen van sets van gedrag functies gekozen worden (in plaats van modellen van effect functies of doel functies).[1]

Verklaringen van uiteenlopende theoriekeuzen op basis van variatie in (epistemische) waarden leidden tot uitvoerig debat in de wetenschapsfilosofie. Een van de voornaamste uitdagingen is te laten zien dat ondanks het feit dat concurrerende theorieën worden gekozen, theoriekeuzen van wetenschappers desalniettemin rationeel zijn. Voorstanders van variatie in waarden beargumenteren dat rationele theorie keuze is gewaarborgd als gekozen theorieën bijdragen aan het behalen van wetenschappelijke doelen, precies omdat ze voldoen aan de waarden op basis waarvan ze gekozen zijn. Dat wil zeggen, dat een gekozen theorie bijdraagt aan het behalen van een wetenschappelijk doel dat men wil bereiken met de theorie precies omdat deze voldoet aan bepaalde (gewenste) kenmerken of eigenschappen (waarden). Bijvoorbeeld, een theorie die voldoet aan de waarde dat deze geen ad hoc hypothesen bevat draagt bij aan het wetenschappelijk doel de falsifieerbaarheid van wetenschappelijke theorieën te maximaliseren. In zoverre waarden zulke geschikte middelen zijn voor het bereiken van wetenschappelijke doelen, wordt (het in stand houden van) variatie van deze waarden in theorie keuze rationeel geacht. Zulke middel-doel interpretaties van waarden stellen in staat tot het rationeel vergelijken van de merites van concurrerende theorieën (of wetenschappelijke modellen): deze theorie/model is beter met betrekking tot deze waarde, die theorie/model is beter met betrekking tot die waarde.

In hoofdstuk 5 adresseer ik deze rationaliteit kwestie in het geval van functionele decompositie in engineering. Ik beargumenteer dat bepaalde modellen bijdragen aan het behalen van de doelen waarvoor ze worden gebruikt omdat ze voldoen aan bepaalde engineering waarden. Dat wil zeggen, dat gekozen model-

[1] De constraint dat alle structuren compatibel zijn (en derhalve alle functies te realiseren) is, uiteraard, een cruciale constraint die belangrijk is in alle functionele modelleer benaderingen. Modelleer benaderingen verschillen echter in welke ontwerp fase aan deze waarde voldaan moet worden. In sommige benaderingen moeten functionele decompositie modellen aan deze waarde voldoen, terwijl in andere hieraan voldaan moet worden in latere ontwerpfases. Alleen in sommige benaderingen is het dus een waarde die van toepassing is op functionele decompositie modellen.

len bijdragen aan het behalen van de doelen waarvoor ze worden gebruikt vanwege de (gewenste) kenmerken of eigenschappen waaraan ze voldoen (en op basis waarvan ze worden gekozen). Deze middel-doel interpretatie van engineering waarden laat zien dat het gebruik van concurrerende modellen door ingenieurs vanuit een praktisch oogpunt rationeel is. Ik beargumenteer dat engineering waarden geschikte middelen zijn voor het bereiken van doelen door onderscheid te maken tussen hoofddoelen en sub doelen van ingenieurs: ik beargumenteer dat modellen die voldoen aan bepaalde engineering waarden direct bijdragen aan het behalen van bepaalde sub doelen, en dat het behalen van die sub doelen vervolgens bijdraagt aan het behalen van hoofddoelen. Ik illustreer dit, onder andere, middels de eerder beschreven engineering waarde (in de context van routinematig ontwerpen) dat de realisatie van een functie niet afhankelijk is niet afhankelijk is van de (eerdere) realisatie van een andere functie door een andere structuur. Ik beargumenteer dat als een functioneel decompositie model (van doel functies) voldoet aan deze engineering waarde, het model bijdraagt aan het behalen van het sub doel een grote verscheidenheid aan functie-structuur connecties in ogenschouw te kunnen nemen. Het hebben van een grote verscheidenheid aan functie-structuur connecties is een sub doel dat, wanneer behaalt, vervolgens bijdraagt aan het hoofddoel van routinematig ontwerpen. Aan de andere kant beargumenteer ik dat als een model (van gedrag functies) voldoet aan de engineering waarde dat de ontwerp oplossingen of structuren gedefinieerd door het model compatibel zijn, het model bijdraagt aan het sub doel dat alle functies in het model gerealiseerd zijn. Behalen van dit sub doel draagt vervolgens bij aan het behalen van het hoofddoel van routinematig ontwerpen.

Het onderscheid tussen hoofd en sub doelen maakt dus inzichtelijk waarom variatie in engineering waarden optreedt in gevallen waar concurrerende modellen worden gebruikt voor het behalen van een gemeenschappelijk hoofddoel. Daarnaast maakt deze analyse het mogelijk modellen rationeel te vergelijken met betrekking tot een bepaalde engineering waarde. Bijvoorbeeld, als men compatibiliteit van structuren belangrijk acht, dan kiest men beter voor een model van gedragsfuncties; als men onafhankelijkheid van functie-structuur connecties belangrijk acht, dan opteert men beter voor een model van doel functies. Er is geen één model dat optimaal aan al van zulke engineering waarden voldoet. Derhalve claim ik dat het gebruik van verschillende functionele decompositie modellen door verschillende ingenieurs vanuit een praktisch oogpunt rationeel is.

Naast het verklaren van het naast elkaar bestaan van verschillende functionele decompositie modellen analyseer ik ook de communicatie problemen die deze co-existentie van modellen teweegbrengt (zie hoofdstukken 6-7). Een kwestie nauw gerelateerd aan methodologische incommensurabiliteit is de vraag of terminologie van concurrerende conceptuele kaders naar elkaar vertaald kan worden. Dit vraagstuk is uitvoerig geanalyseerd in de wetenschapsfilosofie onder de noemer van semantische incommensurabiliteit. In de hoofdstukken 1 en 5 (gebaseerd op onderzoek in hoofdstukken 2-4 en 6-7) beargumenteer ik dat semantische incommensurabiliteit niet van toepassing is op functionele decompositie in engineering. Ik beargumenteer dat er significante overlap is tussen conceptualiseringen die ten grondslag liggen aan de functionele termen in verschillende functionele decompositie modellen. Gegeven deze overlap is het tweede doel van dit proefschrift het verbeteren van de communicatie van functionele decompositie modellen tussen modelleer benaderingen.

In hoofdstukken 6 en 7 analyseer ik een engineering methodologie voor het vertalen van functionele decompositie modellen, en ontwikkel ik conceptuele verbeteringen voor de vertaal stappen in deze methodologie. In deze methodologie worden modellen vertaald met behulp van één specifieke (overkoepelende) conceptualisatie van functie en functionele decompositie. Deze overkoepelende conceptualisatie wordt gebruikt om de betekenis van functies en functionele decompositie modellen te identificeren. Vertalingen van modellen worden ontwikkeld op basis van deze identificaties. Ik beargumenteer, echter, dat deze overkoepelende conceptualisatie verschillende functiebegrippen en functionele decompositie modellen onder één noemer schaart. Als gevolg hiervan gaat functionele informatie verloren in het vertaalproces. De uitwisseling en hergebruik van ontwerp kennis tussen modelleer benaderingen wordt hierdoor bemoeilijkt. Ik beargumenteer dat mijn classificatie van functionele decompositie modellen in staat stelt tot het identificeren en omzeilen van dit vertaal probleem.

De kern idee van mijn alternatieve vertaalstrategie is om eerst functionele decompositie modellen van sets van gedrag functies te vertalen in modellen van sets van fysische gedragingen. Deze stap bewaart informatie over gedrag functies. In een tweede stap worden effecten van de fysische gedragingen in vertaalde modellen geabstraheerd. Deze effect beschrijvingen worden gebruikt om functionele decompositie modellen van sets van effect functies te construeren. Op overeenkomstige wijze kunnen in een derde stap doel functies geabstraheerd worden van effect functies en modellen van sets van doel functies geconstrueerd worden. Deze tweede en derde stap relateren verschillende functionele decompositie modellen. Ik beargumenteer in hoofdstuk 6 dat mijn vertaalstrategie de uitwisseling en hergebruik van ontwerpkennis tussen modelleer benaderingen mogelijk maakt.

About the Author

Dingmar van Eck was born on September 9th, 1976 in Gouda. He studied Theoretical Psychology at the Free University of Amsterdam from 1998 until 2003, specializing in history and philosophy of psychology. In 2003 he obtained his Master's degree with a thesis on cross-theory relations between psychological and neurobiological theories. From 2000 until 2006 he also held a part-time position as an assistant-lecturer at the Free University. After his graduation he continued research on philosophy of psychology, which resulted in two publications. He started his PhD research on *functional decomposition* in 2006 at the Philosophy section of the Department of Technology, Policy, and Management at Delft University of Technology. His research was part of the project *Functional Decomposition in Philosophy and Engineering*, funded by the Dutch Organization of Scientific Research (NWO).

Simon Stevin Series in the Philosophy of Technology
Delft University of Technology & Eindhoven University of Technology
Editors: Peter Kroes and Anthonie Meijers

Books and Dissertations

Volume 1: Marcel Scheele, '*The Proper Use of Artefacts: A philosophical theory of the social constitution of artefact functions*', 2005

Volume 2: Anke van Gorp, *Ethical issues in engineering design, Safety and sustainability*, 2005

Volume 3: Vincent Wiegel, SophoLab, *Experimental Computational Philosophy*

Volume 4: Jeroen de Ridder, *Technical Artifacts: Design and Explanation*, 2006

Volume 5: Melissa van Amerongen, *The Interpretation of artifacts; A critique of Dennett's design stance, 2008*

Volume 6: Krist Vaesen, *A Philosophical Essay on Artifacts and Norms,* 2008

Volume 7: Giacomo Romano, *Thoughtful Things. An investigation in the descriptive epistemology of artifacts,* 2009

Volume 8: Dingmar van Eck, *Functional Decomposition: on Rationality and Incommensurability in Engineering*, 2011

Research Documents

Peter Kroes and Anthonie Meijers (eds.), *'Philosophy of Technical Artifacts'*, 2005

Simon Stevin (1548-1620)

'Wonder en is gheen Wonder'

This series in the philosophy of technology is named after the Dutch / Flemish natural philosopher, scientist and engineer Simon Stevin. He was an extraordinary versatile person. He published, among other things, on arithmetic, accounting, geometry, mechanics, hydrostatics, astronomy, theory of measurement, civil engineering, the theory of music, and civil citizenship. He wrote the very first treatise on logic in Dutch, which he considered to be a superior language for scientific purposes. The relation between theory and practice is a main topic in his work. In addition to his theoretical publications, he held a large number of patents, and was actively involved as an engineer in the building of windmills, harbours, and fortifications for the Dutch prince Maurits. He is famous for having constructed large sailing carriages.

Little is known about his personal life. He was probably born in 1548 in Bruges (Flanders) and went to Leiden in 1581, where he took up his studies at the university two years later. His work was published between 1581 and 1617. He was an early defender of the Copernican worldview, which did not make him popular in religious circles. He died in 1620, but the exact date and the place of his burial are unknown. Philosophically he was a pragmatic rationalist for whom every phenomenon, however mysterious, ultimately had a scientific explanation. Hence his dictum 'Wonder is no Wonder', which he used on the cover of several of his own books.