INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'09
24 - 27 AUGUST 2009, STANFORD UNIVERSITY, STANFORD, CA, USA

ON RELATING FUNCTIONAL MODELING APPROACHES: ABSTRACTING FUNCTIONAL MODELS FROM BEHAVIORAL MODELS

Dingmar van Eck[1]
(1) University of Delft, The Netherlands

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

1 INTRODUCTION

As can be seen in a current review by Erden et al. [1] 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 [2,3,4]. 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 [5,6]. A part of this engineering ontology research consists of developing function ontologies, in which specific concepts of technical function are formalized [7,8,9,10]. These function ontologies prove useful in the storage, retrieval, and communication of functional information between engineers and engineering teams using the *same* ontology [9]. It is however commonplace that different meanings are attached to the concept of technical function in the engineering domain [1,11,12,13,14]. A methodology, developed by Kitamura et al. [15,16] and Ookubo et al. [17] 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 [18], the Reverse Engineering and Redesign methodology of Otto and Wood [19,20,21], the Functional Basis methodology of Stone and Wood [22], the Functional Reasoning methodology of Chakrabarti and Bligh [23], the Dual Stage methodology of Deng, Tor, and Britton [4,24,25], the Functional Concept Ontology methodology of Kitamura, Mizoguchi, and co-workers [9,26], and the Functional Interpretation Language methodology of Price, Bell, and Snooke [27]. 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 FUNCTIONAL MODELLING: 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.

2.1 multi level flow modeling methodology

The Multi Level Flow modeling methodology formulated by Lind [18] 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 [28] 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 [18,29]. 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 [18,29]. 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.

2.2 reverse engineering and redesign methodology

The Reverse Engineering and Redesign methodology formulated by Otto and Wood [19,20,21] 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 [20].

In this methodology, an overall product function is defined as a reproducible relationship between available input and desired output [21]. 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” [19,20,21]. 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. [30], 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 [21]. 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.

2.3 functional basis methodology

The Functional Basis methodology formulated by Stone and Wood [22] 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 [31]. 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 [32].

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 [22] 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.

2.4 functional reasoning methodology

The Functional Reasoning methodology developed by Chakrabarti and Bligh [23] is a methodology that is aimed at supporting engineering design of new products. They present what they call a "functional reasoning scheme" 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 [23]. Effects are defined as intended aspects of causal behavior [33]. 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 [34].

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 [23]. 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.

2.5 dual stage methodology

The Dual Stage methodology developed by Deng, Tor, and Britton [4,24,25], 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 [35,36,37] and to build function ontologies for support of knowledge exchange in collaborative design environments [8].

In the Dual Stage approach, following a distinction made by Chakrabarti [38], two types of functions are defined: purpose functions and action functions [4]. 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 [4]. 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" [4,24]. 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.

2.6 functional concept ontology methodology

The Functional Concept Ontology methodology developed by Kitamura and Mizoguchi [26] and Kitamura et al. [9] 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 [26].

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 [9,26].

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" [9].

2.7 functional interpretation language methodology

The Functional Interpretation Language methodology developed by Price, Bell, and Snooke [27] 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 [39]. 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 [27].

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" [27]. 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:

inoperative, expressed as: In $(f) \leftrightarrow \neg t \,\&\, \neg e$; failed, expressed as: Fa $(f) \leftrightarrow t \,\&\, \neg e$;
unexpected, expressed as: Un $(f) \leftrightarrow \neg t \,\&\, e$; 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 [27]. 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.

In a functional decomposition, an overall function is decomposed into either complete or incomplete sub functions, depending on the type of system analysed [27]. 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, and 4) functional decomposition in two TIF's.

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 [15,16,17].

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 [4] and input-output flow transformations are interpreted as behavior representations [24]. 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. [24] 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 [9]. 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 [17]. 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. [17] 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 [27] remark that what they call low-level functions, referring to an application of the Functional Basis [40], 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 [41] 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.

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 [21]. 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 [cf. 42].

In the Dual Stage, Functional Concept Ontology and Functional Interpretation Language methodologies it is not required that functional descriptions obey conservation laws. Deng [4], for instance, describes the function of a flywheel as providing mechanical energy and the function of a battery as providing electricity. Kitamura et al. [43] 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 [27]. In these functional descriptions energy is created, violating conservation laws [42]. 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 [17]. In the Dual Stage approach, behaviors are connected in sequences in which the output of preceding behaviors provides the input to succeeding behaviors [4]. In the Functional Interpretation Language approach, behaviors are represented as sequentially ordered state transitions [44]. 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.

4 ABSTRACTING FUNCTIONAL MODELS FROM BEHAVIORAL MODELS

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. [15,16], and Ookubo et al. [17]. 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 [45]. Most Functional Basis-functions are translated into Functional Concept Ontology-functions under the assumption that they match in meaning [15,16,17]. 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 [17]. 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 [17]. 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 [17].

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.

4.2 An alternative method

The strategy developed here incorporates a proposal by Garbacz [46]. Garbacz [46] 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.

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.

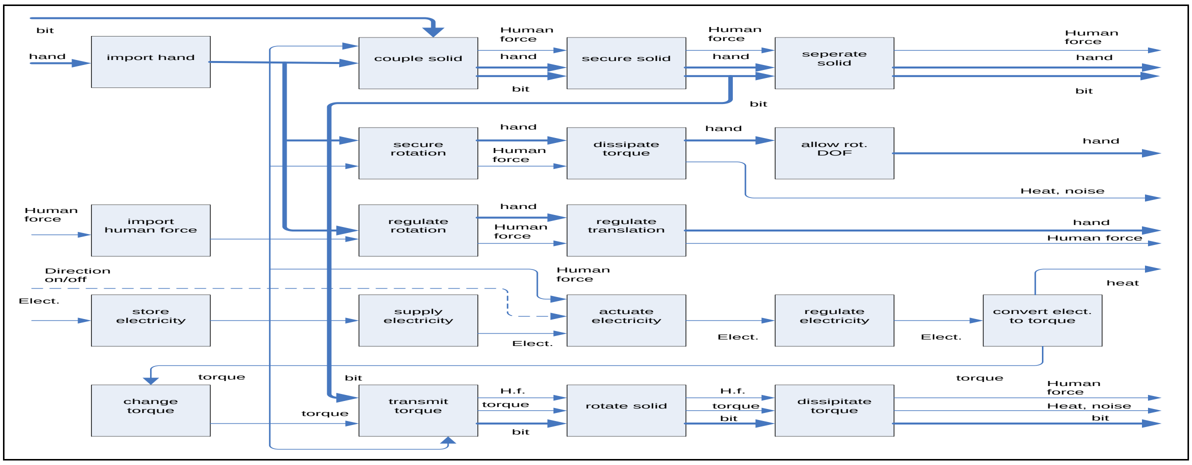

Figure 1. Functional Basis model of a power screwdriver, adopted from Stone et al. [24]

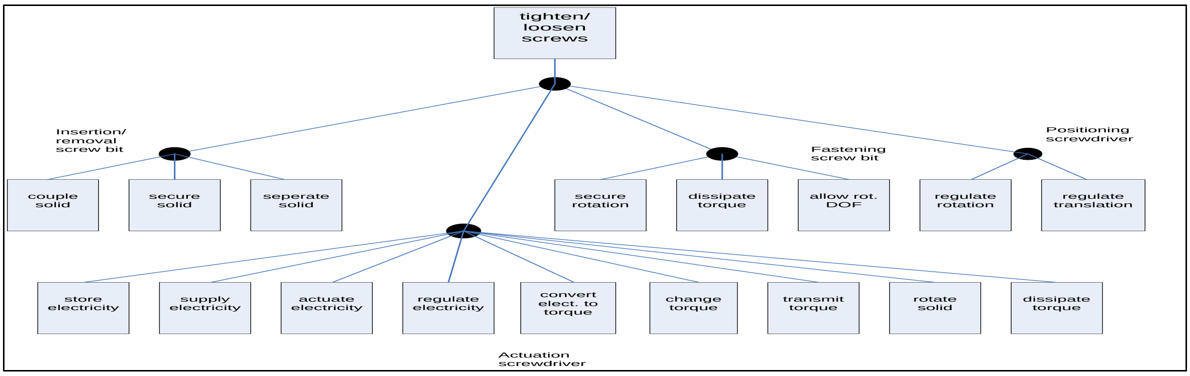

Figure 2. Functional Concept Ontology-inspired model of a power screwdriver. The overall function corresponds to the overall function of the Functional Basis-screwdriver [24].

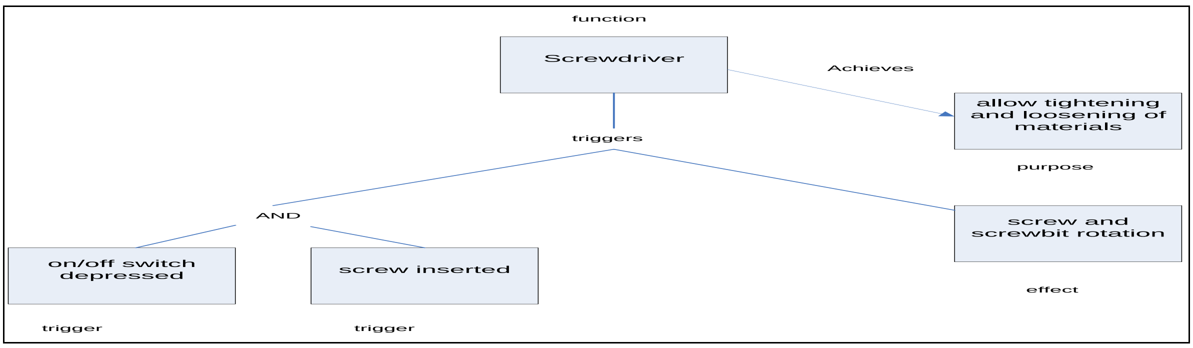

Figure 3. Functional Interpretation Language-inspired model of a power screwdriver, based upon the Functional Basis-overall function of the screwdriver. The sub functions share both an effect and a purpose, but have separate triggers.

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 [31]. Since Functional Basis "import" and "export" operations have no counterparts in the Functional Concept Ontology taxonomy [17], 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 [46], these functional modeling frameworks can be related and functional knowledge exchange established between them, without information loss.

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 referees for their helpful comments and suggestions. This research is supported by the Netherlands Organization of Scientific Research (NWO).

REFERENCES

[1] Erden M.S., Komoto H., Van Beek T.J., D'Amelio V., Echavarria, E, and Tomiyama T. A review of function modelling: approaches and applications. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 2008, 22, 147-169.

[2] Bucciarelli L.L. *Designing Engineers*, 1994 (Cambridge, MA: MIT Press).

[3] Rosenman M.A. and Gero J.S. Purpose and function in a collaborative CAD environment. *Reliability Engineering and System Safety*, 1999, 64, 167-179.

[4] Deng Y.M. Function and behavior representation in conceptual mechanical design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 2002, 16, 343-362.

[5] Ahmed S., Kim S. and Wallace, K.M. A methodology for creating ontologies for engineering design, *Journal of Computing Information Science and* Engineering, 2007, 7, 132-140.

[6] Kim S., Bracewell R. and Wallace, K. Some reflections on ontologies in engineering domain. In *proceedings of TMCE 2008 Symposium*, April 21-25, 2008, Izmir, Turkey, 1271-1284.

[7] Szykman S., Fenves S.J., Keirouz, W. and Shooter, S.B. A foundation for interoperability in next-generation product development systems. *Computer-aided Design*, 2001, 33, 545-559.

[8] Zhang W.Y., Lin L.F., Tong R.F., Li X. and Dong J.X. 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, 2005, pp. 1146-1151.
[9] Kitamura Y., Koji Y. and Mizoguchi, R. An ontological model of device function: Industrial deployment and lessons learned, *Applied Ontology*, 2005/2006, 1, 237-262.
[10] Borgo S., Carrara M., Garbacz P. and Vermaas, P.E. A formal ontological perspective on the behaviors and functions of technical artifacts. *AIEDAM*, 2009, 23, 3-21.
[11] Chittaro L. and Kumar A. N. Reasoning about Function and its Applications to Engineering. *Artificial Intelligence in Engineering*, 1998, 12, 331-336.
[12] Chandrasekaran B. and Josephson J.R. Function in device representation. *Engineering with Computers*, 2000, 16, 162-177
[13] Far B.H. and Elamy A.H. Functional Reasoning Theories: Problems and Perspectives. *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, 2005, 19, 75-88.
[14] Van Eck D., McAdams D.A. and Vermaas P.E. Functional decomposition in engineering: a survey. In *the International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE '07,* September 4-7, 2007, Las Vegas, Nevada, USA.
[15] Kitamura Y., Takafuji S. and Mizoguchi R. Towards a reference ontology for functional knowledge interoperability. In *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.
[16] Kitamura Y., Segawa S., Sasajima M., Tarumi S. and Mizoguchi, R. Deep semantic mapping between functional taxonomies for interoperable semantic search. In: *the Semantic Web*. Eds: Dominique j. and Anutariya C.,2008, 137-151. (Berlin: Springer)
[17] Ookubo M., Koji Y., Sasajima M., Kitamura Y. and Mizoguchi R. Towards interoperability between functional taxonomies using an ontology-based mapping. In *International Conference on Engineering Design, ICED '07, August 28-31, 2007, Paris, France.*
[18] Lind M. Modeling goals and functions of complex industrial plants, *Applied Artificial Intelligence,* 1994, 8, 259-283.
[19] Otto K. N. and Wood K.L. 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.
[20] Otto K.N and Wood K L. Product evolution: a reverse engineering and redesign methodology. *Research in Engineering Design*, 1998, 10 (4), 226-243.
[21] Otto K.N. and Wood K.L. *Product Design: Techniques in Reverse Engineering and New Product Development*, 2001 (Upper Saddle River, NJ: Prentice Hall).
[22] Stone R. B. and Wood K.L. Development of a Functional Basis for design, *Journal of Mechanical Design,* 2000, 122, 359-370.
[23] Chakrabarti A. and Bligh T.P. A Scheme for functional reasoning in conceptual design. *Design Studies,* 2001, 22, 493-517.
[24] Deng Y.M., Tor S.B. and Britton G.A. A dual-stage functional modelling framework with multi-level design knowledge for conceptual mechanical design. *Journal of Engineering Design,* 2000, 11 (4), 347-375.
[25] Deng Y.M., Tor S.B. and Britton G.A. Abstracting and exploring functional design information for conceptual mechanical product design. *Engineering with computers*, 2000, 16, 36-52.
[26] Kitamura Y. and Mizoguchi R. Ontology-based description of functional design knowledge and its use in a functional way server, *Expert Systems with Applications,* 2003, 24, 153-166.
[27] Bell J., Snooke N. and Price C. A language for functional interpretation of model based simulation. *Advanced Engineering Informatics,* 2007, 21, 398-409.
[28] Larsson J.E. Diagnostic reasoning based on means-end models: experiences and future prospects. *Knowledge-Based Systems*, 2002, 15, 103-110.
[29] Larsson J.E. Diagnosis based on explicit means-end models. *Artificial Intelligence,* 1996, 80, 29-93.
[30] Little A., Wood K. and McAdams D. Functional analysis: a fundamental empirical study for reverse engineering, benchmarking, and redesign. In *the 1997 ASME Design Theory and Methodology Conference*, 1997, New York, USA.
[31] Stone R.B., Wood K.L. and Crawford R.H. Using quantitative functional models to develop product architectures. *Design Studies*, 2000, 21, 239-260.

[32] http://function.basiceng.umr.edu/delabsite/repository.html
[33] Chakrabarti A., Sarkar P., Leelavathamma B. and Nataraju B.S. A functional representation for aiding biomimetic and artificial inspiration of new ideas. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 2005, 19, 113-132.
[34] Chakrabarti A. and Bligh T.P. An Approach to functional synthesis of solutions in mechanical conceptual design. Part1: introduction and knowledge representation. *Research in Engineering Design*, 1994, 6, 127-141.
[35] Zhang W.Y., Tor S.B., Britton G.A. and Deng Y.M. EFDEX: a knowledge-based expert system for functional design of engineering systems. *Engineering with Computers*, 2001, 17, 339-353.
[36] Zhang W.Y., Tor S.B. and Britton G.A. Automated functional design of engineering systems. *Journal of intelligent manufacturing*, 2002, 13, 119-133.
[37] Zhang W.Y., Tor S.B. and Britton, G.A. A graph and matrix representation scheme for functional design of mechanical products. *International Journal for Advanced Manufacturing Technology*, 2005, 25, 221-232.
[38] Chakrabarti, A. Supporting two views of function in mechanical designs. *In 15th National Conference on Artificial Intelligence, AAAI'98*, July 26-30, 1998, Madison, Wisconsin, USA.
[39] Price C.J. Function-directed electrical design analysis. *Artificial Intelligence in Engineering*, 1998, 12 (4), 445-456.
[40] Van Wie M., Bryant C.R., Bohm M.R., McAdams D.A. and Stone R.B. A model of function-based representations. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 2005, 19, 89-111.
[41] Chandrasekaran B. Representing function: relating functional representation and functional modeling research streams. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 2005, 19, 65-74.
[42] Vermaas P.E. 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.
[43] Kitamura Y., Sano T., Namba K. and Mizoguchi R. A functional concept ontology and its application to automatic identification of functional structures. *Advanced Engineering Informatics*, 2002, 16 (2), 145-163.
[44] Bell J. and Snooke N. Describing system functions that depend on intermittent and sequential behaviour. In *Proceedings of 28th International Workshop on Qualitative Reasoning*, 2004, Wales, UK, 51-57
[45] Van Eck D. On the conversion of functional models: bridging differences between functional taxonomies in the modeling of user actions. Delft University of Technology manuscript, 2009.
[46] Garbacz P. 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.

Contact: Dingmar van Eck
Delft University of Technology
Department of Philosophy
Jaffalaan 5, 2628 BX, Delft
The Netherlands
Phone: Int + 31 152785417
Fax: Int + 31 152786233
E-mail: D.vanEck@tudelft.nl

Dingmar van Eck conducts PhD research on functional modeling in engineering. His research is focused on analyzing engineering models for functional decomposition and engineering methods for functional knowledge exchange.