

RESPONSE PAPER

Making function modeling practically usable

TETSUO TOMIYAMA,¹ THOM J. VAN BEEK,² ANDRÉS ALBERTO ALVAREZ CABRERA,²
HITOSHI KOMOTO,³ AND VALENTINA D'AMELIO²

¹Manufacturing and Materials Department, Cranfield University, Cranfield, UK

²Faculty of Mechanical, Maritime, and Materials Engineering, Delft University of Technology, Delft, The Netherlands

³National Institute of Advanced Industrial Science and Technology, Tsukuba, Japan

(RECEIVED March 1, 2012; ACCEPTED March 1, 2013)

Abstract

Function modeling is considered potentially useful in various fields of engineering, including engineering design. However, a close look at practices reveals that practitioners do not use formal function modeling so much, while the concept of “function” frequently appears in many practical methods without a vigorous definition. This paper tries to understand why formal function modeling is not practically utilized in industry by analyzing usage cases of function. By observing product development activities in industry, the paper identifies three problems that prevent formal function modeling from wider applications in practices, namely, practitioners’ neglect of function modeling, the lack of practically useful function reasoning, and the complexity of the methods and tools of formal function modeling that make them impractical. Finally, the paper proposes strategies to tackle these problems and illustrates some research efforts in this regard.

Keywords: Function; Function Modeling; Function Reasoning; Product Development; Systems Architecting

1. INTRODUCTION

As a researcher in design theory and methodology (DTM), in particular, function modeling, you might have been asked a question about the utility of the field: “How can I use your research results?”¹ Although some ask this purely out of curiosity, others try to imply that DTM is useless, because people can design even without knowing DTM. Some others try to mean that it might be useful, but there is no need to pay serious attention because it is too trivial.

Unfortunately, this tendency exists in industry as well. At educational institutions, in machine design courses, students are taught systematic design methodologies (such as Hubka & Eder, 1982; Verein Deutscher Ingenieure, 1993; Otto & Wood, 2001; Pahl et al., 2007; Ulrich & Eppinger, 2011) and function modeling. Any design should begin with defining functional requirements, followed by establishing functional structure through functional analysis. Conceptual design is largely driven by functional requirements. By following such a design methodology, the student is expected to become capa-

ble of doing good design, if not the best and most innovative. These students are disappointed when they discover real practices in industry, because design methodologies are not used at all and function modeling is rarely used or even mentioned. Design practices are often intuitive and not systematic. Conceptual design is paid little attention, and creative design is heavily relying only on brainstorming. Consequently, design outputs lack rational explanations and there is no guarantee that the “discovered” solution is “the best solution.” This situation sounds like an exaggeration, but more or less this is the observation of industrial practices made by the authors.

Typically, industrial practitioners do not regard function modeling as something very useful, particularly, for the purpose of design. They have been taught function modeling at school, so they know how to draw a function diagram, but they seldom make one in practice. However, this does not mean that the concept of function itself is useless for practicing designers and engineers in industry. There are a variety of engineering methods (many of which are related to quality, though) that require explicit representation of functions. These methods include quality function deployment (QFD; Mizuno & Akao, 1993), failure mode effect analysis (FMEA; McDermott et al., 1996), value engineering (Yunker, 2003), and design structure matrix (Steward, 1981; Browning, 2001) to name a few. Function here means almost

Reprint requests to: Tetsuo Tomiyama, Cranfield University, Building 50, Manufacturing and Materials Department, Cranfield, Bedfordshire MK43 0AL, UK. E-mail: t.tomiyama@cranfield.ac.uk

¹ The first author was asked this question during a conference in which his PhD student was presenting a theory about function modeling.

anything, ranging from “academic function” to physical behaviors or attributes, although almost all of these methods do not assume a special function modeling method but advance a classic, generic verb–noun pair form (“*to do something*”). As such, no coherent function modeling exists common to these engineering methods; in some cases, a function model is a simple collection of statements in a natural language, and in other cases, a graphical representation consisting of boxes and arrows.

Function modeling is a formal way to define and model functions. According to review articles on function modeling and its relevant fields (Umeda & Tomiyama, 1997; Erden et al., 2008; Borgo et al., 2009), besides the classic verb–noun pair representation (“to do something”), there are three major different formal function modeling methods, namely, transformational methods (Chakrabarti & Bligh, 1996; Stone & Wood, 2000; Chakrabarti & Bligh, 2001; Pahl et al., 2007); methods with physical behavior as medium including the function–behavior–state (FBS) modeling (Umeda et al., 1990, 1996) and the function–behavior–structure modeling (Gero, 1990; Gero et al., 1992), and their variations (e.g., Deng, 2002); and state transition based methods (Goel, 1992; Goel & Stroulia, 1996; Goel et al., 2009). Functional ontology has been also intensively studied (Kitamura et al., 2002, 2004; Kitamura & Mizoguchi, 2004; Borgo et al., 2009; Carrara et al., 2011). Because this paper does not aim at giving detailed comparative descriptions about these three methods, interested readers are invited to refer to the review survey articles mentioned above. As suggested above, however, none of these modeling methods is practically used by industry, in which, instead, function is defined in an ad hoc manner, depending on the usage without any formal model. This matches the observation made in Eckert (2013) that functions have different notions depending their usage context.

As Erden et al. (2008) point out, function is a medium that connects objective descriptions about the physical world in which an artifact operates with subjective descriptions given by its user about what the artifact does. Among these three formal function modeling methods, the transformational methods reject subjective descriptions by focusing on flows (and transformation) of material, energy, and information. In contrast the other two (i.e., the FBS² type and state transition type) permit subjective functional descriptions, because space for objective descriptions is separately given. Because these two types of function modeling allow subjective elements that cannot be unified, functions are now qualified to mean “anything,” depending largely on the context, such as purpose, goal, behavior, effects, and phenomenon (Eckert, 2013; Vermaas, 2013). In contrast, there is a theoretical possibility that their objective elements can be modeled and represented in a unified manner. Goel (2013) recommends functional descriptions be grounded to visuospatial reasoning.

² Here, FBS means Umeda et al.’s (1996) modeling. Gero’s function–behavior–structure modeling is also abbreviated as FBS, but in this paper FBS denotes the former.

Vermaas (2013) reports that even if limited only to objective descriptions, this is sufficiently difficult, if not impossible at all, and identifies four ways to cure the problem. The FBS modeling approach (Umeda et al., 1996) separates objective elements described with qualitative process theory (Forbus, 1984) from subjective functional descriptions. This is Vermaas’s (2013) fourth approach.

However, this paper tries to understand why academically developed formal function modeling methods are not used, but the concept of function itself is commonly used in practice. To do so, first it analyzes various types of engineering methods that rely on the concept of function. Then, observations of industrial product development activities will be made through being embedded within an actual product development environment and having discussions with team members, including system architects and domain experts. Based on these, the paper will identify four usage types of function, namely, to represent the purpose of the artifact; to explain the behavior, structure, or working principle of the target system; to capture customer requirements; and to illustrate the overview of the system. It will then try to understand problems associated with practical usage of function modeling. Function modeling is not used in practice, primarily because truly innovative new design that requires function modeling rarely happens. In addition, practitioners do not simply believe that function modeling is useful, described functions cannot be fully utilized due to the lack of function reasoning, and as the size and complexity of the target system grows, the function model quickly “explodes” and cannot be dealt with easily. Finally, the paper will illustrate some attempts to attack these problems.

2. USAGE OF FUNCTIONAL DESCRIPTIONS

In order to find out actual usage of the concept of function and formal function modeling methods in industry, we conducted a literature study and examined experiences the authors obtained during their interactions (see Table 1) with industry. While originally these interactions were meant to study, for example, system architecting methods, these experiences were sufficiently intensive to understand daily activities in real product development processes.

The following is a nonexhaustive list of engineering methods that incorporate functional descriptions, although they do not employ a specific function modeling method but a classic verb–noun pair form (“*to do something*”) or sometimes sentences in a natural language explaining such concepts as behaviors, working mechanisms, and inputs and outputs.

Requirement descriptions (e.g., INCOSE, 2010) are used during the product definition stage (i.e., before conceptual design) and are documents that describe and analyze customer requirements. These customer requirements include a variety of aspects (not only functional, behavioral, and attributive, but also commercial, legal, as well as descriptions related to all life cycle phases) and are broken down into technical specifications, stored, and reviewed. However, they are not really

Table 1. Interactions with industry

Case	Company	Project	Phase	Researcher' Role	Number of Researchers	Approximate Duration
1	Medical equipment	Product development	All through the whole process	Embedded as a mechanical engineer	1	2 years
2	Precision machinery	Prototyping	Lab prototyping	Internship as a control engineer (observer)	2	3 months
3	Material handling systems	Control software architecture	Data structure definition	Analyzed product life cycle management models	3	1 months
4	Information equipment	Control software architecture	System architecting	Observer	2	1 year
5	Information equipment	Product development (product A)	System architecting	Embedded as an assistant architect	1	6 months
6	Information equipment	Product development (product B)	System architecting	Embedded as an assistant architect	1	1 year
7	Information equipment	Product development (product C)	Lab prototyping	Observer	1	6 months

used in the succeeding stages throughout the product development process, especially within the core development team. Rather, these documents are used for interorganization communication purposes with different organizations (e.g., between the main product development team and an external software development team) and for validation/verification and review purposes at formal occasions such as design reviews. In addition, these documents serve certification and quality control purposes.

QFD (Mizuno & Akao, 1993) is a method to reflect customer's requirements in technical functions and further down in technical specifications. All of these descriptions can include functional descriptions. By creating a house of quality, one tries to balance and prioritize these elements. In QFD, functions are used, but these functions are not related to any technical functions used in, for example, conceptual design.

In *systems engineering*, systems-level design follows conceptual design. Systems architecture is usually functional decomposition in which subsystems are defined and grouped according to their functions. In other words, an appropriate function model helps practitioners to understand the architecture, divided tasks, and eventually points to improve (Alvarez Cabrera et al., 2011). Methods to overview systems architecture of complex systems were developed (Sobek & Smalley, 2008; Borches Juzgado, 2010) that use a schematic representation on an A3 sheet. In addition, systems engineering documents are full of behavior-level information that is connected to functions [e.g., imagine functions in Systems Modeling Language (SysML) diagrams (Object Management Group, 2010), although functional descriptions in SysML are a bundled set of input and output parameters and their transformation].

Product life cycle management (PLM) in one company involves multidisciplinary products, and the company creates documents containing functional and behavioral descriptions stored in a knowledge management system of PLM. Here, functions are common denominators for elements in different

domains. For instance, a “conveyor” has a “controller” to control a “motor.” The controller was a part of a system achieving functionality of transport and stored in three separate depositories of mechanical (three-dimensional geometric model), software (a piece of controller code), and control domains (systems dynamics model representing differential equations). Here a function is rather an identifier. More recent development of PLM software systems offers systems engineering features such as SysML diagrams.

Quality control (e.g., ISO 9000 series; International Organization for Standardization, 2008) in product development basically requests to document all processes. Within these processes, documents that describe functional requirements are made following predefined templates. These documents are used for the validation purposes during the development process and for certification after the development.

In *value engineering* (Younker, 2003), value is defined as “function divided by cost,” so in order to increase value, functions must be increased or costs decreased. It is a method to achieve higher quality and lower costs. Functions are represented in the classic verb–noun pair form and should be evaluated against requirements.

FMEA (McDermott et al., 1996) is used to increase the reliability of the product and is a frequently exercised technique. Failures are defined as malfunctioning of the product, and as such, FMEA requires first understanding the behavior and purpose of the system. FMEA uses functional descriptions but usually in a classic verb–noun pair form.

Through our experiences with industrial product development activities listed in Table 1, we observed the followings. (It must be noted that these are observations about those activities and are never meant to be universal conclusions.)

Most product development activities in industry are not new design but routine design or improvement design in which the system architecture is given or fixed, meaning function structure remains more or less the same. Due to in-

creasing time pressure (e.g., time to market) engineers tend to move on to later stages of design as quickly as possible. Completely new architecture is regarded highly risky. Consequently, function structure or function-level architecture is not really discussed. However, there are an increasing number of practitioners who find it useful to use those methods listed above as well as Suh's (1990) axiomatic design and TRIZ (Altshuller, 1984), both of which require functional descriptions (but in a classic verb–noun pair form). This simply indicates that functions as a concept are used very frequently in a wide variety of engineering, yet conceptual design that looks at functional structure or system architecture based on formal function modeling methods is less appreciated.

At the same time, behind these usage examples of functional descriptions, one can identify four major purposes. These usage types may overlap each other.

1. To represent the purpose of the artifact: This type is used during function-based conceptual design as a driving force of design (Gero, 1990; Umeda et al., 1990; Gero et al., 1992; Umeda et al., 1996; Chakrabarti & Bligh, 1996; Stone & Wood, 2000; Charabarti & Bligh, 2001; Deng, 2002; Pahl et al., 2007). The function in FMEA also belongs to this type. In many cases, these can eventually be associated with behaviors, physical principles, and attributes. Formal function modeling methods can be used here.
2. To explain the behavior, structure, or working principle of the target system: This type aims at giving an explanation (or understanding) of the artifact (for communication; in addition to those authors in (1), see Kitamura et al., 2002, 2004; Kitamura & Mizoguchi, 2004; Borgo et al., 2009; Carrara et al., 2011). For this purpose, formal function modeling and operation methods are assumed.
3. To capture customer requirements: This is obvious in the examples of requirement descriptions, QFD, and value engineering. For these, functional descriptions are often given in the verb–noun pair form or even in a natural language sentence.
4. To illustrate the overview of the system: This is perhaps a subcategory of (2), but, for instance, in systems engineering, systems architecture can be often explained through functional decomposition (Borches Juzgado, 2010; Alvarez Cabrera et al., 2011). The usage example in PLM belongs to this category as well. Here, a function is a common denominator that connects chunks of information that belong to different fields (or views) during communication among stakeholders.

3. WHY IS FUNCTION MODELING NOT USED IN INDUSTRY?

Practitioners use functions for a variety of occasions, yet they do not use formal function modeling methods or rely on functional

descriptions as a driving force for conceptual design. Still, they were often exposed to formal function modeling methods during their education, leaving three possible reasons to explore why they are not used: the formal modeling methods are known but neglected by practitioners; it is assumed that they are useless or that it is not necessary to use them; or practitioners want to use the methods yet practical issues obstruct usage.

1. “*Never used it*” syndrome (or simple neglect): Practitioners believe that they know how to model and represent functions, because it looks very easy or at least not difficult. They have seen or heard of function modeling but never received any formal training about a formal method together with its potential powerful applications (e.g., functional analysis with respect to quality, cost, and architecture). Perhaps, they have never explored beyond “transformational boxes” or “to do something” verb–noun pairs. Therefore, they know function modeling only superficially and cannot believe that it would bring in useful results.
2. “*No added value*” syndrome: Practitioners do not feel like going deeper than what they (must) do now. For instance, if documenting functional requirements is needed only for the certification purpose, they do not add more information than necessary and will not explore other applications. They feel their time and effort to complete function diagrams were wasted, if there is no practical added value of the diagrams. This simply means that even if one builds a function model, it becomes immediately useless.
3. “*Not practical*” syndrome: For practitioners, academically developed methods are too formal, abstract, and far away from real products, missing direct connections with any other information more frequently used (e.g., three-dimensional geometric models). In addition, because these modeling methods were not seriously used in industry, they have never been improved to become useful and practical. This leads to another problem, which is that a function model may easily explode if it is a complete picture of the entire artifact: even a simple mechanism can already contain numerous components, relationships, and other types of interrelated information about functions (see Fig. 1, which illustrates an example FBS diagram, displaying an overwhelming amount of information in a nonintuitive manner). Without a professionally developed tool that can handle so much information, and without proper training of the designer, function diagrams cannot practically fulfill their purposes of providing explanation and overview. In addition, it must be emphasized that often an appropriate function model helps practitioners to understand the architecture and divided tasks, and eventually points to improve. However, they also see too many new ideas. Without an appropriate method to reduce those generated ideas, the whole function modeling exercise often diverges rather than converges.

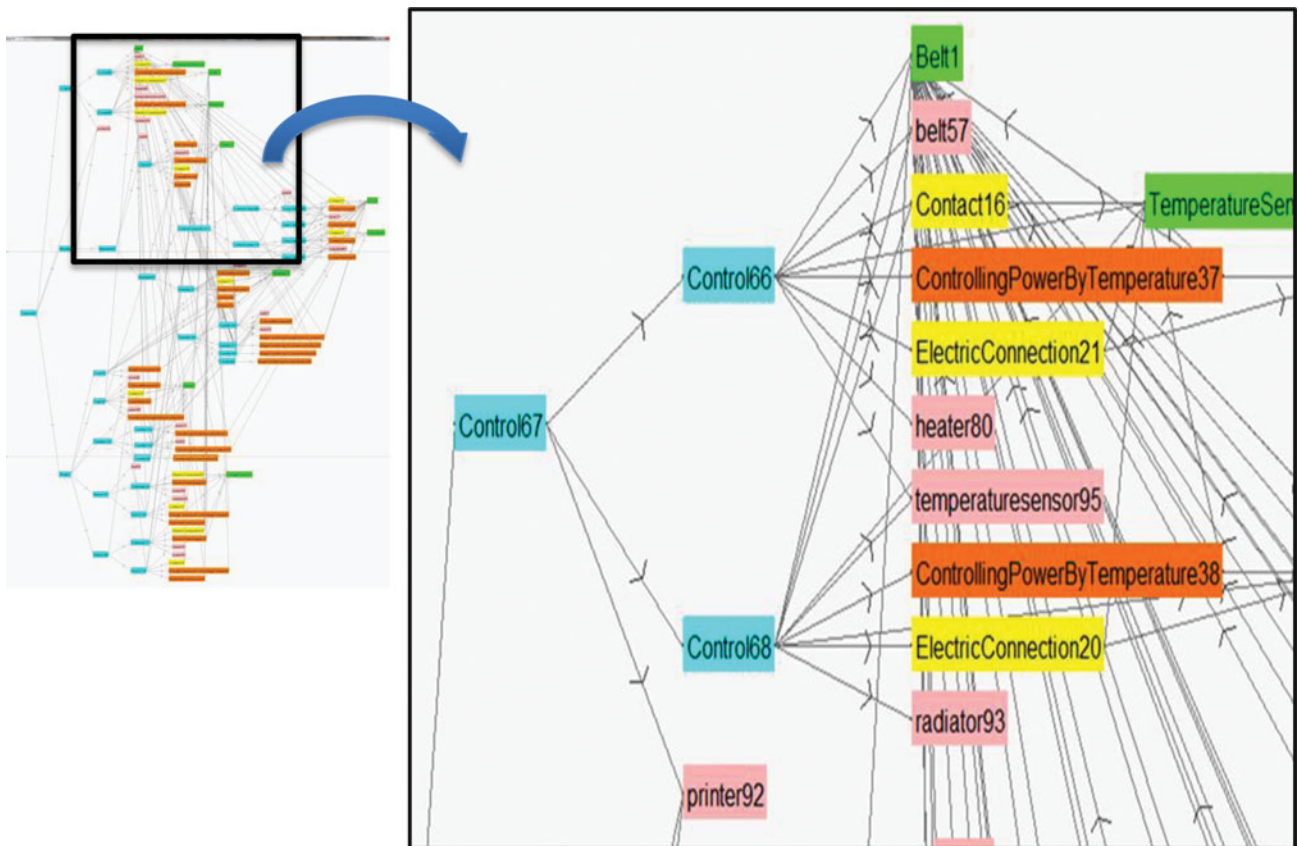


Fig. 1. A function–behavior–structure (FBS) model of a simple printer mechanism. The blue boxes in the online version are functions, orange boxes are physical phenomena, green boxes are entities, and yellow boxes are the relationships among the entities. [A color version of this figure can be viewed online at <http://journals.cambridge.org/aie>]

4. THREE POSSIBLE DIRECTIONS TO MAKE FUNCTION MODELING MORE PRACTICAL

Corresponding to the three syndromes listed above, we are now able to identify three possible directions for future development to make function modeling practically usable. However, these are merely suggestions and not answers.

One way to tackle the “never used it” syndrome would be to demonstrate that truly creative design becomes systematically feasible with conceptual design that involves deep functional-level analysis. For example, Komoto and Tomiyama (2010, 2011) have developed a computer-based tool to support system architecting, which will be illustrated later.

Against the second “no added value” syndrome, we argue that the lack of function reasoning to generate added value after a function model is created is the fundamental cause for the unpopularity of formal function modeling, rather than the lack of a unified function modeling method. The current research trends pay too much attention to modeling and representation of function. Therefore, it is critical to demonstrate that function modeling can generate added value in engineering design by deriving useful knowledge through function reasoning (Far & Elamy, 2005). This may include the following:

1. *Functional level simulation for validation:* During systems architecting, validation needs to be performed quantitatively, but this implies that validation can be performed only after some progress is made. Function-level (or qualitative) simulation would improve the efficiency of the whole product development process, but such a technique applicable to a wide range of models does not exist yet despite the progress in qualitative physics (Price et al., 2006). Instead, identifying overlooked design failures would be helpful at least to warn designers of potential design failures (D’Amelio et al., 2011).
2. *Function redundancy design:* FBS modeling was originally developed to look for redundant functions to improve “fault tolerance” and to design a “function-redundant type self-maintenance machine” (Tomiyama et al., 1993; Umeda et al., 1994).
3. *Identification of latent functions:* Similar to function-redundant design, this might be useful to find, for instance, a (sub)system that performs a required function. This could be used for indexing knowledge bases of past designs and prevent design repetition.
4. *Function-level techniques for systems architecting:* For instance, Stone et al. (2000) demonstrated that it is possible to generate product architecture from a function

diagram. The authors' group also developed some techniques and tools addressing this topic (Komoto & Tomiyama, 2010, 2011). These techniques could facilitate and improve "system-level design" (Sobek, 2006).

- Komoto and Tomiyama (2010, 2011) demonstrated that a computer-based tool can support the decomposition process during systems architecting (Fig. 2). Physical phenomena that embody the required functions are identified from functional descriptions in the FBS format (Umeda et al. 1996). These physical processes contain parameters, and by clustering these parameters, different decompositions are systematically generated. Subsequent quantitative performance evaluation facilitates choosing the best architecture.
- Alvares Cabarera et al. (2011) developed a method and a computer tool (AM tool) for architecture-centric model-based product development. Architecture defines subsystems and their relationships among them, and system architecting involves (hierarchical) decomposition, behavior definition of subsystems, and interface definition. Functions give an overview of subsystems (behaviors) and form the basis for cross-disciplinary communication (see Fig. 3, which illustrates the top-level system architecture). The tool can connect to further development stages, such as performance analysis or control software generation (Alvares Cabarera et al., 2010).
- Other possibilities would be connecting functional descriptions with user-level upfront information such as workflow that would eventually guarantee and increase the usability of the product in connection with functions (van Beek & Tomiyama, 2011).

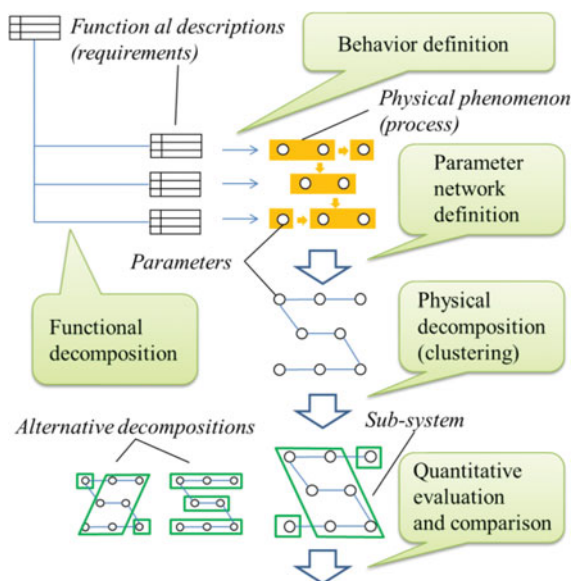


Fig. 2. The systems architecting process. [A color version of this figure can be viewed online at <http://journals.cambridge.org/aic>]

From functional descriptions, it is possible to generate subsystems that interface definitions as design structure matrix.

Against the third "not practical" syndrome, countermeasures include the development of easy-to-use tools and training methods that can still handle a reasonably complicated function structure (i.e., not a toy problem). Ideally, these tools and techniques must be extremely easy to use without even training. In addition, the tool should be equipped with knowledge bases that can be professionally useful in function modeling. It must be capable of dealing with large amounts of information (e.g., hierarchical treatment), versioning, and multiple user accesses. In addition, integration should not be forgotten. After all, because function is all about what the system does, functional information must be accessible and usable from not only mechanical design tools but also tools for electrical, electronics, control, and software design.

Finally, because function-level design can involve true end users (e.g., in the case of medical equipment, doctors, and nurses, not necessarily designers and engineers), tools like the FBS modeler (Fig. 1) are too difficult. From this point of view, end user level requirements, needs, wishes, and constraints should be described with a simple, focused method that allows linking to functional information. We have investigated the use of workflow modeling for this purpose (van Beek & Tomiyama, 2011).

5. CONCLUSIONS

This paper began with a proposition that function modeling is primarily researched by academics and taught at educational institutions but in reality not used by practitioners. The paper analyzed this situation and found that, while formal function modeling methods are not used, the concept of function itself appears at every corner of engineering in a very simple form (such as the classic verb–noun form or sentences in a natural language). One major reason why function modeling is not being used is that the majority of product development is routine design or improvement design and does not require function-level design in the first place. Another reason is that practitioners do not recognize very well the benefits of applying function modeling to (new) design.

To go deeper into this observation, the paper analyzed various types of engineering methods that rely on the concept of function. The paper identified four usage types of function, namely, to represent the purpose of the artifact; to explain the behavior, structure, or working principle of the target system; to capture customer requirements; and to grasp the overview of the system. In all these usage types, we could identify practical problems why function modeling is not used in practice: practitioners simply do not believe in the usefulness of function modeling, the lack of added values of function modeling, and the explosion of the model as the size and complexity increase.

At the end, the paper proposed strategies about how to tackle these problems. Against the lack of belief in utility of

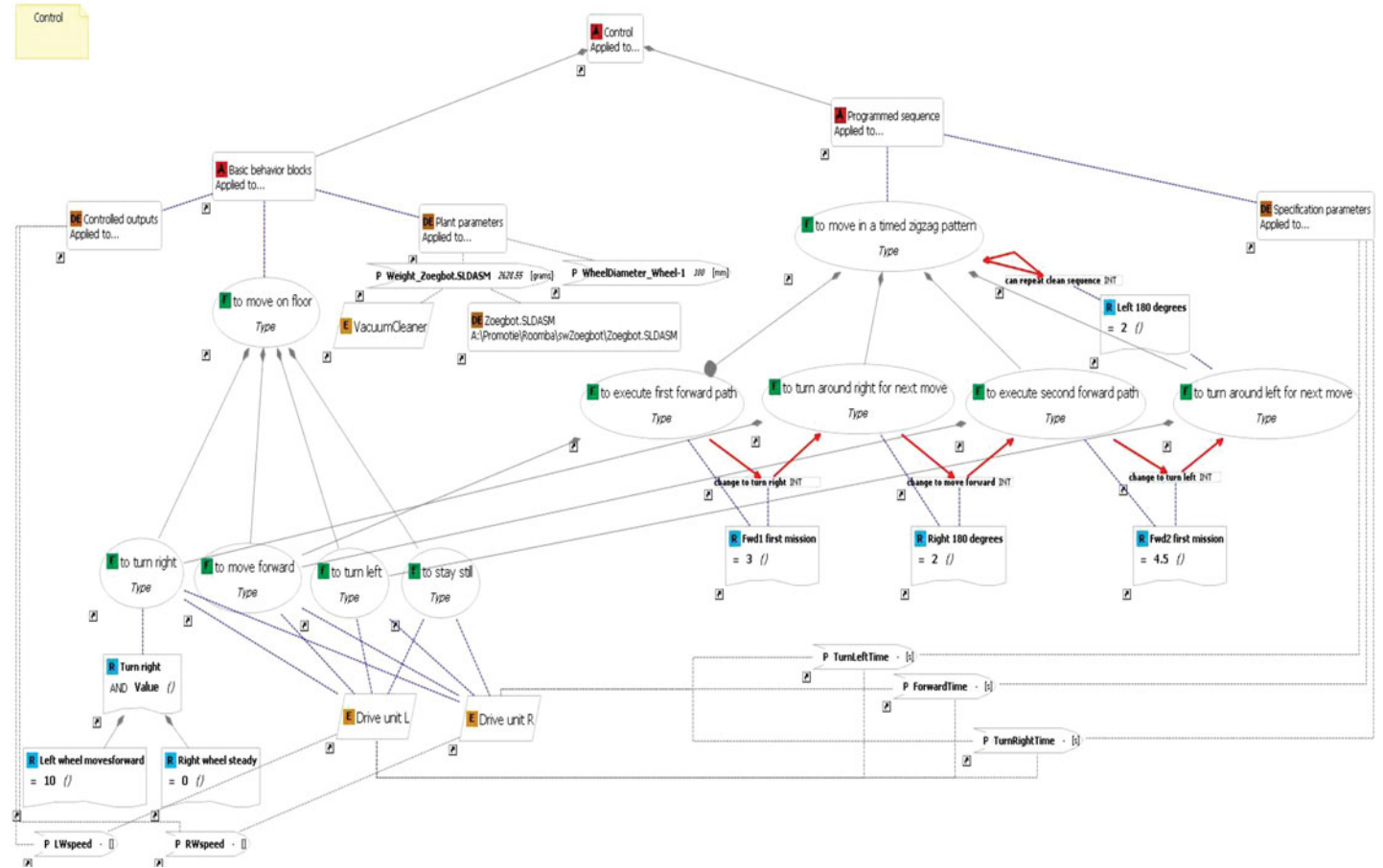


Fig. 3. Modeling an autonomous vehicle in the AM tool. A, aspect; SM, design task or “synthesis method”; DTR, design task relation; DE, domain entity; E, entity; Er, entity relation; Fo, formulae; F, function; Fr, function relation; P, parameter; R, requirement; V, view. [A color version of this figure can be viewed online at <http://journals.cambridge.org/aie>]

function modeling, one needs to demonstrate that creative design can be generated with deep functional-level analysis. Against the second lack of added value, we argued that various types of function reasoning should be developed. Among others, functional-level simulation for validation and functional methods for architecting seem promising. Against the third practicality (complexity) problem, the development of easy-to-use tools and training methods is a must. These tools should be equipped with a professional-level user interface and operational capabilities. In addition, for real end users, we may think about replacement of function models with much easier models.

Finally, we must conclude that function modeling itself is not a big issue. Bigger issues are how to advocate engineering design methods combined with function modeling and how to make better use of functional descriptions. To achieve this, we recommend three strategies: show the usefulness of function modeling, derive useful information from function modeling, and develop practically usable professional tools.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the following projects that supported the part of the research work: Automatic Generation of Control Software for Mechatronics Systems (supported by the Innovation-Oriented Research Programme, Integral Product Creation and Realization (IOP IPCR) of the Dutch Ministry of Economic Affairs, Agriculture and Innovation), and Darwin and Octopus (carried out under the responsibility of the Embedded Systems Institute in Eindhoven and partially supported by the same ministry under the BSIK program). Philips Healthcare and Océ Technologies were industrial partners of the Darwin and Octopus projects, respectively. The work was done while the authors were working at Delft University of Technology between 2004 and 2012.

REFERENCES

- Altshuller, G.S. (1984). *Creativity as an Exact Science*. New York: Gordon & Breach Science.
- Alvarez Cabrera, A.A., Foeken, M.J., Tekin, O.A., Woestenenk, K., Erden, M.S., De Schutter, B., van Tooren, M.J.L., Babuška, R., van Houten, F.J.A.M., & Tomiyama, T. (2010). Towards automation of control software: a review of challenges in mechatronic design. *Mechatronics* 20(8), 876–886.
- Alvarez Cabrera, A.A., Woestenenk, K., & Tomiyama, T. (2011). An architecture model to support cooperative design for mechatronic products: a control design case. *Mechatronics* 21(3), 534–547.
- Borches Juzgado, P.D. (2010). *A3 Architecture Overviews*. PhD Thesis. Twente University.
- Borgo, S., Carrara, M., Garbacz, P., & Vermaas, P.E. (2009). A formal ontological perspective on the behaviors and functions of technical artifacts. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 23(1), 3–21.
- Browning, T.R. (2001). Applying the design structure matrix to system decomposition and integration problems: a review and new directions. *IEEE Transactions on Engineering Management* 48(3), 292–306.
- Carrara, M., Garbacz, P., & Vermaas, P.E. (2011). If engineering function is a family resemblance concept: assessing three formalization strategies. *Applied Ontology* 6(2), 141–163.
- Chakrabarti, A., & Bligh, T.P. (1996). An approach to functional synthesis of mechanical design concepts: theory, applications, and emerging research issues. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 10(4), 313–331.
- Chakrabarti, A., & Bligh, T.P. (2001). A scheme for functional reasoning in conceptual design. *Design Studies* 22(6), 493–517.
- D'Amelio, V., Chmarra, M.K., & Tomiyama, T. (2011). Early design interference detection based on qualitative physics. *Research in Engineering Design* 22(4), 223–243.
- Deng, Y.M. (2002). Function and behavior representation in conceptual mechanical design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 16(5), 343–362.
- Eckert, C. (2013). That which is not form: the practical challenges in using functional concepts in design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 27(3), 217–232 [this issue].
- Erden, M.S., Komoto, H., van Beek, T.J., D'Amelio, V., Echavarría, E., & Tomiyama, T. (2008). A review of function modeling: approaches and applications. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 22(2), 147–169.
- Far, B.H., & Elamy, A.H. (2005). Functional reasoning theories: problems and perspectives. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 19(2), 75–88.
- Forbus, K.D. (1984). Qualitative process theory. *Artificial Intelligence* 24(3), 85–168.
- Gero, J. (1990). Design prototypes: a knowledge representation schema for design. *AI Magazine* 11(4), 26–36.
- Gero, J.S., Tham, K., & Lee, H. (1992). Behaviour: a link between function and structure in design. *IFIP Transactions B: Applications in Technology, Intelligent Computer-Aided Design B-4*, 193–225.
- Goel, A. (1992). Representation of design functions in experience-based design. *IFIP Transactions B: Applications in Technology, Intelligent Computer-Aided Design B-4*, 283–308.
- Goel, A.K. (2013). A 30-year case study and 15 principles: implications of an artificial intelligence methodology for functional modeling. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 27(3), 203–215 [this issue].
- Goel, A.K., Rugaber, S., & Vattam, S. (2009). Structure, behavior, and function of complex systems: the structure, behavior, and function modeling language. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 23(1), 23–35.
- Goel, A.K., & Stroulia, E. (1996). Functional device models and model-based diagnosis in adaptive design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 10(4), 355–370.
- Hubka, V., & Eder, W.E. (1982). *Principles of Engineering Design*. London: Butterworths Scientific.
- INCOSE. (2010). *INCOSE Requirements Management Tools Survey*. Accessed November 25, 2012, at <http://www.incose.org/productspubs/products/rmsurvey.aspx>
- International Organization for Standardization. (2008). *Quality Management Systems—Requirements*. ISOTC76, ISO 9001:2008. Geneva: International Organization for Standardization.
- Kitamura, Y., Kashiwase, M., Fuse, M., & Mizoguchi, R. (2004). Deployment of an ontological framework of functional design knowledge. *Advanced Engineering Informatics* 18(2), 115–127.
- Kitamura, Y., & Mizoguchi, R. (2004). Ontology-based systematization of functional knowledge. *Journal of Engineering Design* 15(4), 327–351.
- Kitamura, Y., Sano, T., Namba, K., & Mizoguchi, R. (2002). A functional concept ontology and its application to automatic identification of functional structures. *Advanced Engineering Informatics* 16(2), 145–163.
- Komoto, H., & Tomiyama, T. (2010). A system architecting tool for mechatronic systems design. *CIRP Annals—Manufacturing Technology* 59(1), 171–174.
- Komoto, H., & Tomiyama, T. (2011). Multi-disciplinary system decomposition of complex mechatronics systems. *CIRP Annals—Manufacturing Technology* 60(1), 191–194.
- McDermott, R.E., Mikulak, R.J., & Beaugard, M.R. (1996). *The Basics of FMEA*. New York: Productivity Press.
- Mizuno, S., & Akao, Y. (1993). *QFD: The Customer-Driven Approach to Quality Planning & Deployment*. Tokyo: Asian Productivity Organization.
- Object Management Group. (2010). *OMG Systems Modeling Language (OMG SysML)* Version 1.2, OMG Document Number: formal/2010-06-02. Accessed April 13, 2012, at <http://www.omg.org/spec/SysML/1.2/>
- Otto, K., & Wood, K. (2001). *Product Design: Techniques in Reverse Engineering and New Product Development*. Upper Saddle River, NJ: Prentice Hall.
- Pahl, G., Beitz, W., Feldhusen, J., & Grote, K.-H. (2007). *Engineering Design—A Systematic Approach* (3rd ed., Wallace, K., & Blessing, L. Trans. and Eds.). Berlin: Springer.

- Price, C.J., Travé-Massuyès, L., Milne, R., Ironi, L., Forbus, K., Bredeweg, B., Lee, M.H., Struss, P., Snooke, N., Lucas, P., Cavazza, M., & Coghill, G.M. (2006). Qualitative futures. *Knowledge Engineering Review* 21(4), 317–334.
- Sobek, D.K. (2006). System-level design: a missing link? *International Journal of Engineering Education* 22(3), 533–539.
- Sobek, D.K., II, & Smalley, A. (2008). *Understanding A3 Thinking: A Critical Component of Toyota's PDCA Management System*. New York: Productivity Press.
- Steward, D.V. (1981). The design structure system: a method for managing the design of complex systems. *IEEE Transactions on Engineering Management* 28(3), 71–74.
- Stone, R.B., & Wood, K.L. (2000). Development of a functional basis for design. *ASME Journal of Mechanical Design* 122, 359–370.
- Stone, R.B., Wood, K.L., & Crawford, R.H. (2000). A heuristic method for identifying modules for product architectures. *Design Studies* 21(1), 5–31.
- Suh, N.P. (1990). *The Principles of Design*. Oxford: Oxford University Press.
- Tomiyaama, T., Umeda, Y., & Yoshikawa, H. (1993). A CAD for functional design. *CIRP Annals—Manufacturing Technology* 42(1), 143–146.
- Ulrich, K.T., & Eppinger, S.D. (2011). *Product Design and Development* (5th ed.). New York: McGraw–Hill.
- Umeda, Y., Takeda, H., Tomiyama, T., & Yoshikawa, H. (1990). Function, behaviour, and structure. *Applications of Artificial Intelligence in Engineering: V. Proc. 5th Int. Conf.* (Gero, J.S., Ed.), Vol. 1, pp. 177–194. Boston: Computational Mechanics Publications.
- Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y., & Tomiyama, T. (1996). Supporting conceptual design based on the function–behavior–state modeler. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 10(4), 275–288.
- Umeda, Y., & Tomiyama, T. (1997). Functional reasoning in design. *IEEE Expert* 12(2), 42–48.
- Umeda, Y., Tomiyama, T., Yoshikawa, H., & Shimomura, Y. (1994). Using functional maintenance to improve fault-tolerance. *IEEE Expert* 9(3), 25–31.
- van Beek, T.J. & Tomiyama, T. (2011). Workflow modelling of intended system use. In *Views on Evolvability of Embedded Systems* (van de Laar, P., & Punter, T. Eds.), pp. 153–170. New York: Springer.
- Verein Deutscher Ingenieure. (1993). *Systematic Approach for the Design of Technical Systems and Products*. VDI 2221. Düsseldorf: Verein Deutscher Ingenieure.
- Vermaas, P.E. (2013). The coexistence of engineering meanings of function: four responses and their methodological implications. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 27(3), 191–202 [this issue].
- Younker, D.L. (2003). *Value Engineering: Analysis and Methodology*. Boca Raton, FL: CRC Press.

Tetsuo Tomiyama has been a Professor of life cycle engineering in the Manufacturing and Materials Department of Cranfield University since October 2012. Prior to this appointment, he was a Professor at Delft University of Technology between 2002 and 2012 and at the University of Tokyo between 1998 and 2002. He was awarded his doctoral degree in precision machinery engineering from the University of Tokyo in 1985. His research fields include DTM, function modeling, multidisciplinary product development, life cycle engineering, and maintenance engineering. He is currently a Royal Society Wolfson Research Merit Award Holder.

Thom van Beek is a part-time PhD candidate at Delft University of Technology. He received his MS degree from that same university in 2006. The topic of his PhD research is the evolvability of system architectures and how to improve the system architecting process by incorporating user workflow modeling combined with function modeling and modularization. Besides his job as a researcher, he cofounded the company Fietsenmakers (bicycle repairmen) in 2011. He is a system architect in engineering consultancy and product development with projects focused on bicycle simulator design, dynamics, and control.

Andrés A. Alvarez Cabrera is currently developing micro-acoustical products for Sonion A/S in The Netherlands. He obtained his PhD degree from the Delft University of Technology in 2011, researching the practical use of architecture-level models in multidisciplinary model-based design. Dr. Alvarez Cabrera holds an MS degree jointly awarded by the Institut national des sciences appliquées de Lyon (INSA Lyon) and Escola Tècnica Superior d'Enginyeria Industrial de Barcelona/Universitat Politècnica de Catalunya (ETSEIB/UPC) and a BS degree from Universidad Nacional de Colombia (UNAL).

Hitoshi Komoto has been a Research Scientist at the Advanced Manufacturing Research Institute of the National Institute of Advanced Industrial Science and Technology in Japan since 2011. He conducted research in system architecting of adaptable systems as a postdoc at Delft University of Technology between 2009 and 2011. He obtained a PhD from the same university in 2009. He received a Diplom Ingenieur from Karlsruhe Institute of Technology in Germany in 2004. Dr. Komoto's current research interests include computational support in various design domains such as product-service systems design and mechatronics design.

Valentina D'Amelio is an Aerospace Engineer working as a Product Manager at Hukseflux Thermal Sensors in The Netherlands. She graduated in 2005 from the Università degli studi di Roma La Sapienza with a final thesis concerning a haptic system for an unmanned aerial vehicle. She received her PhD in December 2010 from the Department of Biomechanical Engineering at Delft University of Technology. The topic of Dr. D'Amelio's research was design methodology for mechatronics systems.