Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2017), *31*, 476–500.
© Cambridge University Press 2017 0890-0604/17
doi:10.1017/S0890060417000403

Function modeling combined with physics-based reasoning for assessing design options and supporting innovative ideation

HOSSEIN MOKHTARIAN,[1,2] ERIC COATANÉA,[1] AND HENRI PARIS[2]

[1]Tampere University of Technology, Department of Mechanical Engineering and Industrial Systems, Tampere, Finland
[2]University Grenoble Alpes, CNRS, G-SCOP Laboratory, Grenoble, France

(Received October 14, 2016; Accepted May 25, 2017)

Abstract

Functional modeling is an analytical approach to design problems that is widely taught in certain academic communities but not often used by practitioners. This approach can be applied in multiple ways to formalize the understanding of the systems, to support the synthesis of the design in the development of a new product, or to support the analysis and improvement of existing systems incrementally. The type of usage depends on the objectives that are targeted. The objectives can be categorized into two key groups: discovering a totally new solution, or improving an existing one. This article proposes to use the functional modeling approach to achieve three goals: to support the representation of physics-based reasoning, to use this physics-based reasoning to assess design options, and finally to support innovative ideation. The exemplification of the function-based approach is presented via a case study of a glue gun proposed for this Special Issue. A reverse engineering approach is applied, and the authors seek an incremental improvement of the solution. As the physics-based reasoning model presented in this article is heavily dependent on the quality of the functional model, the authors propose a general approach to limit the interpretability of the functional representations by mapping the functional vocabulary with elementary structural blocks derived from bond graph theory. The physics-based reasoning approach is supported by a mathematical framework that is summarized in the article. The physics-based reasoning model is used for discovering the limitations of solutions in the form of internal contradictions and guiding the design ideation effort.

Keywords: Dimensional Analysis Conceptual Modeling Framework; Function Modeling; Function Reasoning

1. INTRODUCTION

Tomiyama et al. (2013) provide a solid analysis of the reasons behind the important gap that exists between the study and usage of function modeling in academia and among industrial practitioners. The concept of a function can be used for several purposes within the engineering design process. It is, for example, used in requirements engineering. Requirements templates such as boilerplates (Dwyer et al., 1999) are often formed using simple subject–verb–noun triplets. Requirements do not exclusively represent functions, but a great part of them are functional requirements (to do something). The function is a classical way to describe the overall purpose of a system, to describe the internal structure, architecture, and behavior of a system. Different aspects of function thinking or modeling have already been used and taught for a long time as an important part of the engineering design process (Pugh, 1991; Otto & Wood, 2001; Pahl & Beitz, 2013). For example, for development phases and tools in requirements engineering to describe the functional requirements, quality function deployment to allocate customer needs to functions, system engineering to represent the system architecture, and also for system development management purposes, value engineering (Miles, 1967). Therefore, why, despite its presence all over the engineering design process, is function modeling not more widely used by design practitioners in several of the design and engineering communities? Several reasons exist that can explain the limited usage of function modeling in industry (Tomiyama et al., 2009, 2013). First, the engineering academic community, which is trying to promote function modeling, often presents studies related to new product development, even though most of the design activities inside companies are routine or incremental design tasks. The academic community seldom studies incremental or routine design tasks. Consequently, practitioners often

Reprint requests to: Hossein Mokhtarian, Department of Mechanical Engineering and Industrial Systems, Korkeakoulunkatu 6, P.O. Box 589, FI-33101 Tampere, Finland. E-mail: hossein.mokhtarian@tut.fi

consider that their everyday activities cannot be supported by function modeling. Tomiyama et al. refer to this as a "not practical" syndrome among practitioners (Tomiyama et al., 2009, 2013). Second, the added value of function modeling is often not immediately perceived by practitioners. It is often considered more efficient and more immediately rewarding to represent a solution quickly in a three-dimensional computer-aided design software tool, instead of taking time to abstract the solution in the form of a function model. In addition, function models, when developed, can quickly explode and become difficult to manage; function modeling is often seen as a source of wasted time. Third, there are few professional software tools capable of representing big function models efficiently on a computer screen of limited size. It is particularly difficult to get an overall picture of a complex function model on a computer screen. Another element limiting the impact of function analysis is the abstraction gap that exists between function models and design structures. In the literature, several models have been proposed to bridge this gap. The function–behavior–structure (FBS models; Gero, 1990) and the requirement–function–behavior–structure model (Christophe et al., 2010) are both attempts to connect functions to behaviors, states, and structures. Nevertheless, those models provide little operational support for function-level or qualitative simulation of system behaviors (Tomiyama et al., 2013). The qualitative simulation should improve the product development process (Sen & Summers, 2013; Tomiyama et al., 2013).

The present article proposes to integrate function modeling into a broader framework to achieve three concrete goals: first, to support the representation of physics-based reasoning; second, to use this physics-based reasoning to assess design options; and third, to use the framework to support innovative ideation. In this article, the exemplification of the functional-based approach is performed via the use of a case study proposed for this special issue: a glue gun. A reverse engineering approach is applied, and the authors seek an incremental improvement of the solution. The approach follows an iterative process to break the functions down from a black box model to a functional model with the desired level of detail. The approach aims at converting the function models to a list of governing equations and a causal graph between the variables in the system.

The rest of this paper is organized as follows: a state-of-the-art analysis of the concept of functions and of different functional techniques is presented in Section 2. This section ends with a description of the approaches that can be used to limit the variability of function modeling. In Section 3 and Section 4, the successive modeling steps and theoretical aspects of the dimensional analysis conceptual modeling (DACM) framework are presented. This is followed by the case study in Section 5, in which DACM framework modeling is applied to model the glue gun, to illustrate the several different modeling options, and to demonstrate the added value of the framework. In the discussion/conclusion Section 6, the capabilities, current limitations, and future developments of the DACM framework are discussed further.

2. BACKGROUND

2.1. Nature of functions in different methodologies and theories

To introduce the concept of functions, the definition of an artificial system proposed by Le Moigne is relevant. According to Le Moigne (1994), influenced by Von Bertalanffy and other systems theorists, a general system is an artifact (i.e., an artificial object) evolving in a certain environment to fulfill a purpose (i.e., a finality). This artifact functions (i.e., does activities) and its internal structure evolves over time, without losing its structure. Artifact or natural systems do activities. Consequently, they exhibit functions. These functions are an abstract concept describing the activities of a system. The concept of a function is present in sciences such as biology (Dusenbery, 1992), economics (Stahel, 1997), and systems theory (Le Moigne, 1994; Luhmann, 2013). The concept is extremely useful in analyzing complex systems. To quote from Herbert Simon (1996): "We define a polar bear by the conjunction of a project: survive by functioning, an environment: The Arctic continent, then by analysing the structural anatomy of this bear. . . ."

The importance attributed by systems theory and systems engineering to the concept of a function is also present in engineering design. Nevertheless, the concrete usage made of the concept depends greatly on the authors and the viewpoints they adopt. In the work of Pahl and Beitz (2013), a functional structure is defined as "a meaningful and compatible combination of sub-functions into an overall function." The functions are classified as main and auxiliary functions. Main functions are those subfunctions that serve the overall function directly, and auxiliary functions are those that contribute to it indirectly. The definition of a function and the relations between functions and design parameters are general, and the final decision about the meaningful and compatible combination of the function depends uniquely on the designer's personal preference. In axiomatic design (Suh, 1990), functional requirements are defined as "the minimum set of independent requirements that completely characterize the design objective for a specific need." The concept of a function is still fuzzy, and no distinction between main and auxiliary functions is made by the author. In the initial general design theory in 1981, Yoshikawa defines a function thus: "When an entity is exposed to a circumstance, a peculiar behavior appears corresponding to the circumstance. This behavior is called a visible function. Different behaviors are observed for different circumstances. The total of these behaviors is called a latent function. Both are called function inclusively" (Yoshikawa, 1981). In the NFX50-151 standard (NFX50-151, 1991), a function is defined as "an action of a product or one of its components expressed in terms of finality." The standard also distinguishes two types of functions. The first one is called a service function. The service function is "the actions expected of the product in order to answer the user's needs." The second type of function is a constraint

https://doi.org/10.1017/S0890060417000403 Published online by Cambridge University Press

function, which is the "limitation of the designer's freedom considered to be necessary for the applicant."

In most of the design methodologies or theories, the arguments about functions are not intended to give a clear definition of the function itself, but to show how desired overall functions are decomposed into identifiable subfunctions until they correspond to certain entities or design objects. Often implicitly (and in disagreement with principles from value analysis or system engineering), the designer has a solution in mind and maps this solution with a function decomposition matching this representation. This aspect often remains implicit and is rarely studied in functional analysis. In particular, when incremental innovation or routine design tasks are taking place, a functional model is the result of an iterative process that ends when the functional model matches the physical solution. This interplay between function/structure–behavior requires further research. Functions are usually used in two ways, for analyzing an existing object by discovering "*How does this object function?*" or to design a new service or artifact by answering the question "*What are the artifact's functions?*" In other terms, function modeling can be used to perform reverse engineering analysis, as per its main use in Otto and Wood (2001) or to create new artifacts. The nature of the day-to-day design activity characterized by routine or incremental design tasks is better grasped by answering the question "*How does this object function?*" and by performing reverse engineering. The question nevertheless remains of "*How is value for the designer to be generated with the reverse engineering approach?*" There is a need for a methodology based on reverse engineering and capable of analyzing the weaknesses of existing solutions but also exploring the design space and evaluating the potential design directions. The concept of value is addressed in the DACM framework developed in the article. The selection of colors for the causal graphs in DACM is a way to represent the values and viewpoints of the designers.

Currently, function modeling offers few methods to provide solid support to those objectives. The methods briefly presented above allow the gap between functions on one side and structure and behavior on the other side to be bridged to some extent but are not capable of providing simulation capabilities and especially capabilities to support physics-based reasoning to assess design options and for ideation purposes. A fundamental paradox emerges. Function modeling is an attempt to abstract and formalize design problems in order to understand the nature of the problem better, but at the same time, there is a need for early concretization and validation via prototyping and/or simulation. How can this be done quickly and easily in early stages without the need for complex prototyping or simulations? The article aims to reconcile these viewpoints by associating function models and early simulations.

Functions are used at different stages of the design process for different purposes. Specific processes and tools reflecting the different viewpoints and usages have been developed. This section provides a rapid overview of the most common processes and function representations. In systems engineering (INCOSE, 2012; System Engineering Fundamentals, 2013), a specific effort is made to identify the functional requirements, to decompose them to lower function levels, to allocate performances and limitations to the different functional levels, to define functional interfaces, to develop functional architectures, and to transform functional architectures into physical architectures. Figure 1 illustrates this rich usage of functions in systems engineering.

Figure 2 presents the different steps of the systems engineering process where the concept of functions is used. The functionalities of a system are considered throughout the life cycle of a design project. On the contrary, the functional modeling occurs only at the beginning of a development process. At this early stage too, verification is needed, and instead of considering the development process as a single V model (VDI, 1993), it might be more appropriate to consider multiple imbricated V cycles where verifications take place at each stage of the process. This article aims to provide such a type of verification capability for function modeling. Figure 3 summarizes a few of the most common forms used for the decomposition of a function or functional architecture, such as the functional tree, the functional structure, and the coupling matrix (NFX50-151, 1991; Le Moigne, 1994). The functional structure is the most commonly used.

The functional boxes themselves can be depicted using three colors to characterize the level of knowledge associated with those boxes. Figure 4 presents those colors. The inputs and outputs of the functional boxes also have different forms. The representation from Pahl and Beitz shown in Figure 5 is the most commonly used.

Another way to represent a function is the octopus diagram in Figure 6. The diagram presents the different elements of the environment and the system to be designed. The diagram allows the listing of the different service functions of systems, as well as the constraint functions. It can also be replaced in the system modeling language using a use case diagram (Friedenthal et al., 2008).

A design structure matrix (DSM) is a way of representing a graph but also a functional architecture. A good overview of DSM usage is provided by (Eppinger & Browning, 2012). A DSM is especially used to model the structure of systems or processes. A DSM lists all the constituent parts of a system or the activities of a process and the corresponding information exchange, interactions, or dependency patterns. DSMs compare the interactions between elements of a similar nature. Figure 7 presents an example of a DSM used to map functions together. A DSM is a square matrix (i.e., it has the same number of columns and lines) that maps elements of the same domain. A domain mapping matrix associates elements of a different nature in a matrix format and can also be used to map, for example, functions and components. The DACM framework presented in Section 3 utilizes DSM as an efficient way to automatize the physics-based reasoning approach.

The architecture of a system is usually represented using the functional structure or a rich representation language

https://doi.org/10.1017/S0890060417000403 Published online by Cambridge University Press

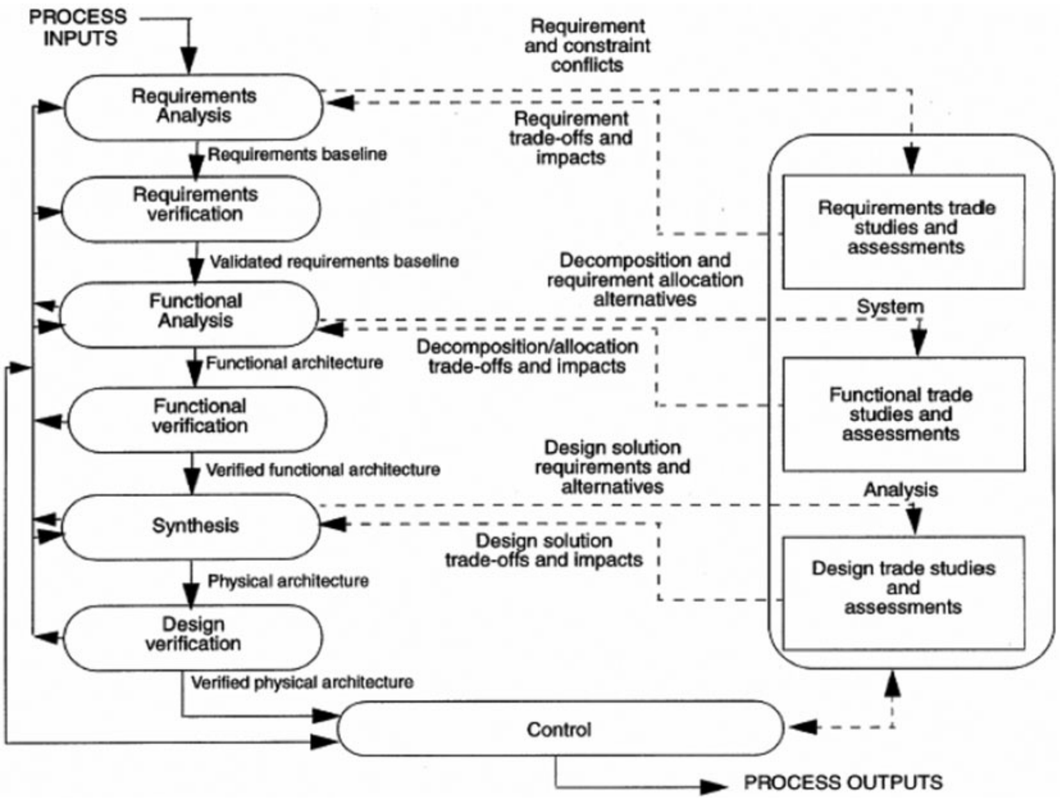

PROCESS INPUTS
Requirements Analysis
Requirements baseline
Requirements verification
Validated requirements baseline
Functional Analysis
Functional architecture
Functional verification
Verified functional architecture
Synthesis
Physical architecture
Design verification
Verified physical architecture
Control
PROCESS OUTPUTS
Requirement and constraint conflicts
Requirement trade-offs and impacts
Decomposition and requirement allocation alternatives
Decomposition/allocation trade-offs and impacts
Design solution requirements and alternatives
Design solution trade-offs and impacts
Requirements trade studies and assessments
System
Functional trade studies and assessments
Analysis
Design trade studies and assessments

Fig. 1. Design process in the IEEE 1220 Standard (IEEE, 2005).

such as Integrated Definition Methodology (IDEF; Hanrahan, 1995) or one of the diagrams from system modeling language (Friedenthal et al., 2008), such as an activity diagram or a sequence diagram. Through all its variants the IDEF representation language allows multiple aspects of functions to be represented. The sequence of functions can also be represented using languages such as Petri nets or Grafcet (NFC03-190+R1, 1995). Figure 8 represents the two-function modeling of a hybrid vehicle. In the hybrid series architecture, only electrical energy is used to generate the final mechanical energy required for the wheels, while in the parallel architecture both mechanical and electrical energy can be used simultaneously. Each of those powertrain configurations offers specific advantages. The parallel configuration can generate more instant power, while the series architecture is more fuel efficient.

Functional modes existing in complex systems can also be represented via a hybrid representation model such as that presented in Figure 9. These modes each represent the activations of different functions in a single-function model of a hybrid vehicle. This presentation is not exhaustive, but the aim was to present the richness of the functional description where multiple modes of functional representations have been developed over time. As a summary of this description, the usage of functions within design methods as a concept and as a design technique representation varies significantly. In general, the literature related to complex systems advises making intensive use of function modeling (Hmelo-Silver et al., 2008). On the contrary, the field of product development and design thinking (Bowler, 1976; Rowe, 1991) seldom refers to the concept of functions, probably because of the scope of the problems tackled by those approaches. This significant difference between systems engineering and design thinking is a source of a major question for the authors of this article. Tension exists between the need to abstract and the parallel need to prototype. Prototyping seems more rewarding for designers than abstracting. In particular, the benefit of physical prototypes is immediately visible and provides a feeling of concrete achievement. This impression of achievement is also present with digital prototypes.

2.2. Physics-based reasoning in function modeling

The early stages of designing a product usually involve the activity of proposing a variety of possible solutions to satisfy the design requirements. The designer might analyze the feasibility of the initial solutions. He might consider the trade-offs and select one solution among the variety of possible solutions. He might analyze how changing a design parameter affects the overall performance of a proposed solution. Therefore, the selection of one from among the possible existing solutions is very dependent on the experience of the designer of providing rationales. In order to enhance such analysis in the early stage of design, one possible direction

https://doi.org/10.1017/S0890060417000403 Published online by Cambridge University Press

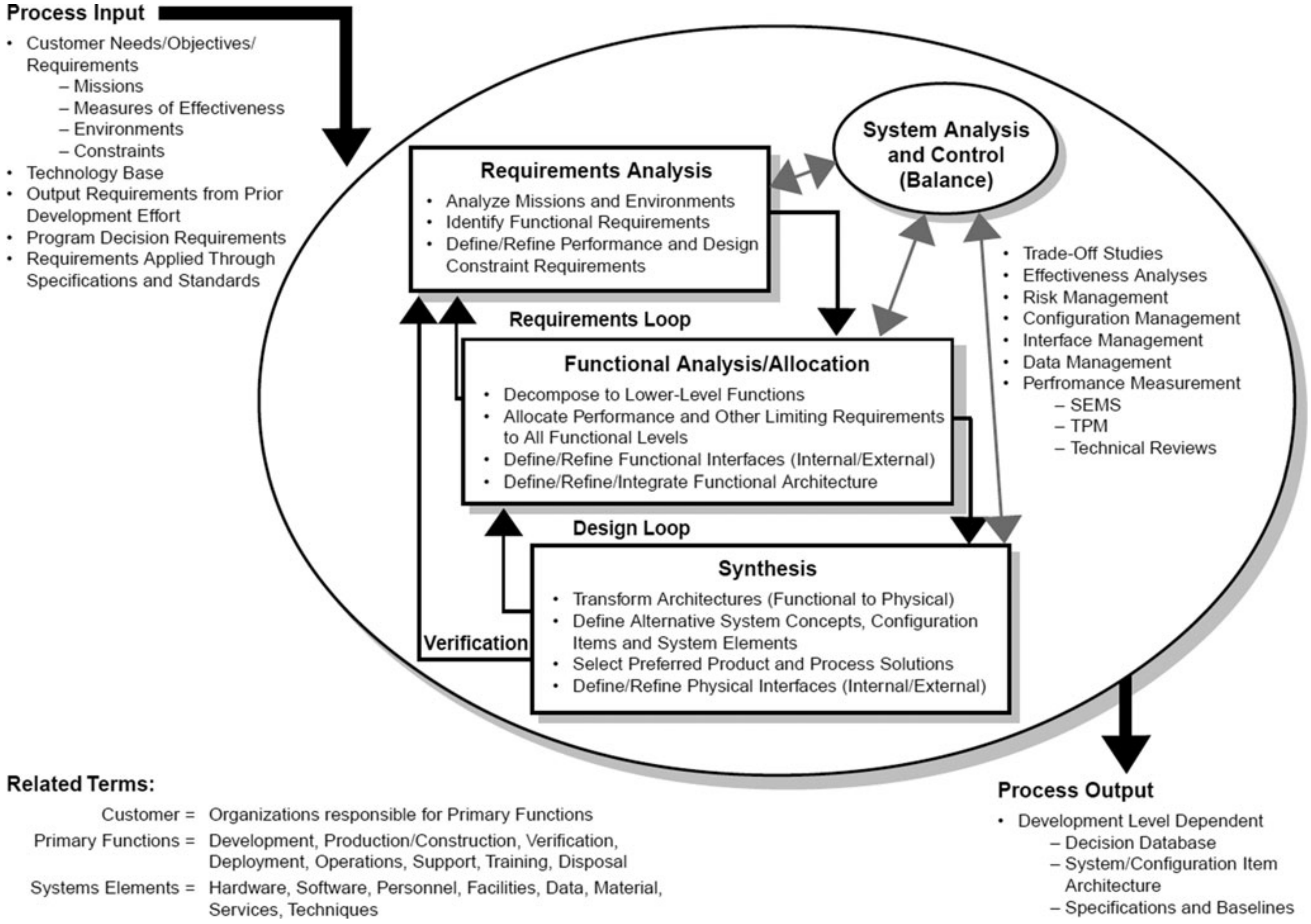

Process Input
• Customer Needs/Objectives/ Requirements
– Missions
– Measures of Effectiveness
– Environments
– Constraints
• Technology Base
• Output Requirements from Prior Development Effort
• Program Decision Requirements
• Requirements Applied Through Specifications and Standards
Requirements Analysis
• Analyze Missions and Environments
• Identify Functional Requirements
• Define/Refine Performance and Design Constraint Requirements
System Analysis and Control (Balance)
• Trade-Off Studies
• Effectiveness Analyses
• Risk Management
• Configuration Management
• Interface Management
• Data Management
• Perfromance Measurement
– SEMS
– TPM
– Technical Reviews
Requirements Loop
Functional Analysis/Allocation
• Decompose to Lower-Level Functions
• Allocate Performance and Other Limiting Requirements to All Functional Levels
• Define/Refine Functional Interfaces (Internal/External)
• Define/Refine/Integrate Functional Architecture
Design Loop
Synthesis
• Transform Architectures (Functional to Physical)
• Define Alternative System Concepts, Configuration Items and System Elements
• Select Preferred Product and Process Solutions
• Define/Refine Physical Interfaces (Internal/External)
Verification
Related Terms:
Customer = Organizations responsible for Primary Functions
Primary Functions = Development, Production/Construction, Verification, Deployment, Operations, Support, Training, Disposal
Systems Elements = Hardware, Software, Personnel, Facilities, Data, Material, Services, Techniques
Process Output
• Development Level Dependent
– Decision Database
– System/Configuration Item Architecture
– Specifications and Baselines

Fig. 2. The systems engineering process (Systems Engineering Fundamentals, 2013).

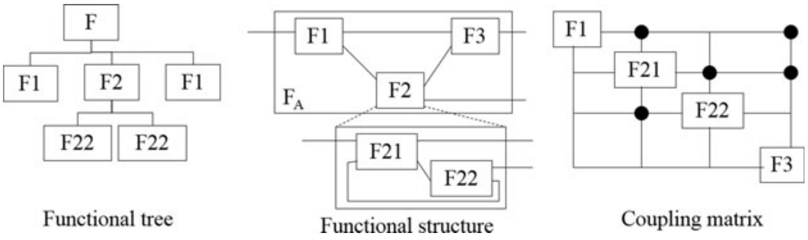


Fig. 3. Different functional representations.

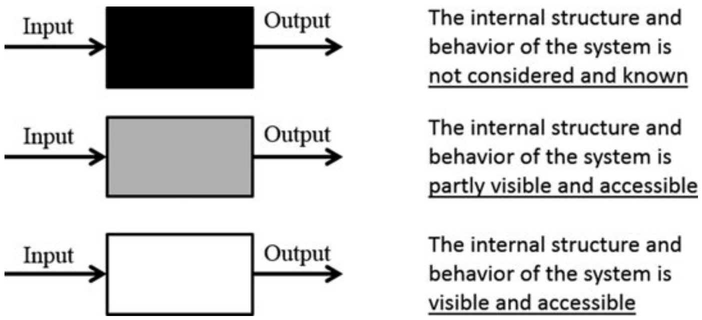

Input
Output
The internal structure and behavior of the system is not considered and known
Input
Output
The internal structure and behavior of the system is partly visible and accessible
Input
Output
The internal structure and behavior of the system is visible and accessible

Fig. 4. The color code used in the standard IEEE 1220 Standard (IEEE, 2005) to represent a system.

https://doi.org/10.1017/S0890060417000403 Published online by Cambridge University Press

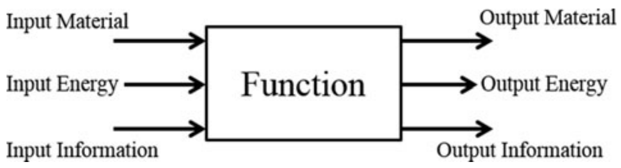


Fig. 5. Inputs and outputs in functional boxes.

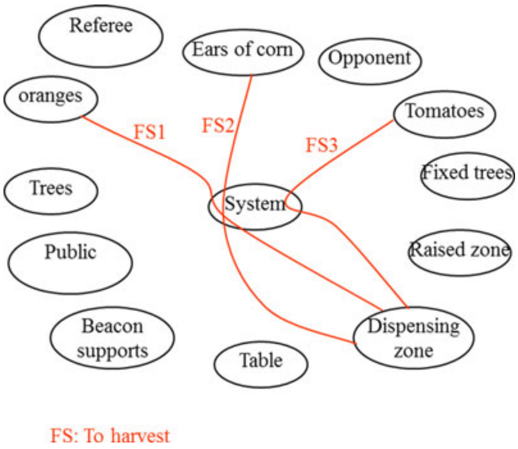
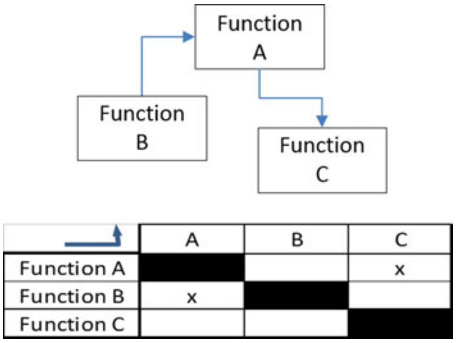


Fig. 6. Functional representation using an octopus diagram (de la Bretesche, 2000).

is to enable physics-based reasoning on function models. In his dissertation Sen (2011) stated that a function-based representation is suitable for supporting early design analysis reasoning. Sen and Summers (2013) identified requirements to enable physics-based reasoning from a function model. They extracted the following requirements:

1. *Coverage:* This is the ability to cover the knowledge and principles of various domains and their interactions, such as electrical, mechanical, thermal, and chemical engineering.
2. *Consistency:* Consistency is an internal property in the representation to prevent internal conflict.
3. *Validity against the laws of physics:* The function model representation should remain valid against the existing laws of physics in each domain.
4. *Physics-based concreteness:* The functions should be defined in terms of physical actions.
5. *Normative and descriptive modeling:* The representation should support both developing a function model for new product design (so-called normative modeling) and the function modeling of an existing artifact, concepts, or physical principles (so-called descriptive modeling).
6. *Qualitative modeling and reasoning:* It must enable support for both qualitative and quantitative reasoning.



	A	B	C
Function A	■		x
Function B	x	■	
Function C			■

Fig. 7. Example of a simple design structure matrix (function to function mapping).

The first three requirements were stated to be generic requirements, and the others are based on their study of identified gaps in function-based design (Summers & Shah, 2004). Those requirements were taken into account in developing the DACM framework and user interface.

2.3. Variability in function modeling, DACM scope, and relevant approaches in literature

The function modeling process is a source of great variability, which can be caused by multiple factors, such as the modeler's preferences and experience, for example, the functional vocabulary used by the modelers, the level of abstraction, and the level of detail selected. Variability can also exist at the functional architecture level. Variability as such is not a source of problems if the goal is to generate a large number of solutions. These sources of variability can be a source of creativity during the early stages during the divergent thinking process. Nevertheless, when repeatability in the models is required in order to communicate function modeling without ambiguity, the variability of function modeling is a source of problems for the modelers and for any physics-based reasoning analysis.

A mechanism ensuring convergence in the modeling process is required in this work if we want to generate a repeatable physics-based reasoning approach. Two qualities of models are important to fulfill for functional models in this work. Those characteristics are known as abstraction and fidelity. The fidelity of a model refers to the degree of exactness of the model compared to the real world (Roza, 2005). Moving from a high-level model of a system to a more detailed level containing more functions will increase the fidelity of the model. Increasing the fidelity of the model might be useful when the simulation of an existing system is intended. Abstraction is the selection of essential functions and neglects the unnecessary functions when modeling a system (Roza, 2005). From a value analysis perspective, unnecessary functions are functions that do not contribute directly to the global service function of the system (NFX50-151, 1991). Reducing the abstraction by considering more functions of the system will increase the comprehensiveness of the model. To reduce the variability in functional models, an initial approach

https://doi.org/10.1017/S0890060417000403 Published online by Cambridge University Press

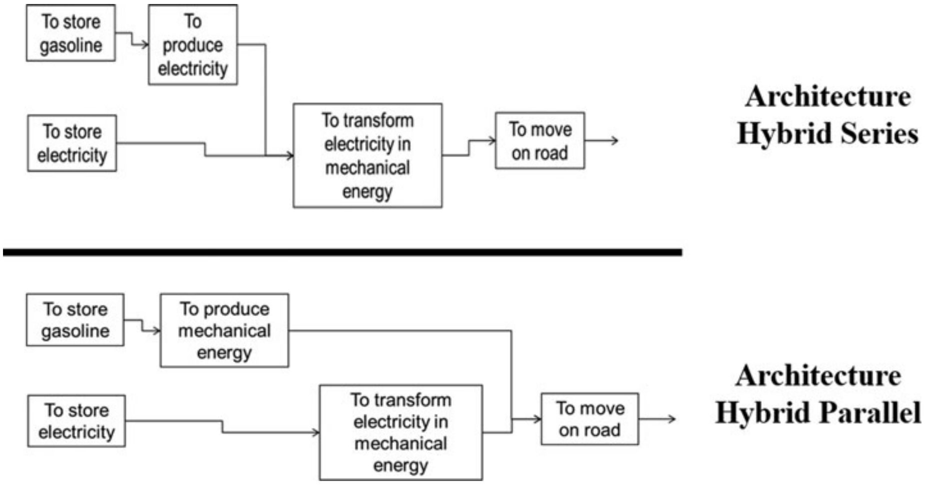

To store gasoline
To produce electricity
To store electricity
To transform electricity in mechanical energy
To move on road
Architecture Hybrid Series
To store gasoline
To produce mechanical energy
To store electricity
To transform electricity in mechanical energy
To move on road
Architecture Hybrid Parallel

Fig. 8. Two functional architectures of a hybrid vehicle.

consists of limiting the functional vocabulary to be used. A significant effort was made in this direction by Hirtz et al. (2002) in their development of a reconciled functional basis. They provided a reconciled list of functional vocabulary but also a list of fundamental energies conveyed by functions, as well as their names in the form of generalized effort and flow. Several authors in the community indicate the benefits of the usage of the vocabulary in Hirtz' functional basis vocabulary (Ahmed & Wallace, 2003; Kurfman et al., 2003; Sen et al., 2010; Helms et al., 2013).

The design process requires multiple iterations between functional abstraction and concretization to be able to efficiently converge toward a validated function model. The concretization part is not covered by the work of Hirtz et al. (2002). The concretization does not necessarily require concrete components, but more abstract elementary bricks are needed. Those bricks can be the elementary energy sources, transformation processes, and storage processes existing in nature or in artificial artifacts. Bond graph theory provides a compact list of those elementary bricks (Karnopp et al., 2012). Those bricks are universal and present in each energy domain with different names. They fulfill elementary functions belonging to the functional basis. A combination of these elementary functions allows more complex functions to be developed. These bricks support the iterative movement between function abstraction and concretization. They help to close the gap between abstraction and validation, but this is not sufficient. Variables and equations are also needed to create simulations. Some features of bond graph theory are included in DACM. Nevertheless, the purpose of the DACM framework is different from bond graph theory because DACM includes engineering design specifics. These specifics are the necessity to have criteria to detect weaknesses in design solutions, the need to support exploration of the design space, and the need to direct the design process toward innovative solutions. It should be noted that DACM is also different from the dissertation of Coatanéa (2005). The usage of dimensional analysis was already present in Coatanéa (2005) but not the colored causal graph reasoning associated with function modeling. DACM expands this initial work greatly.

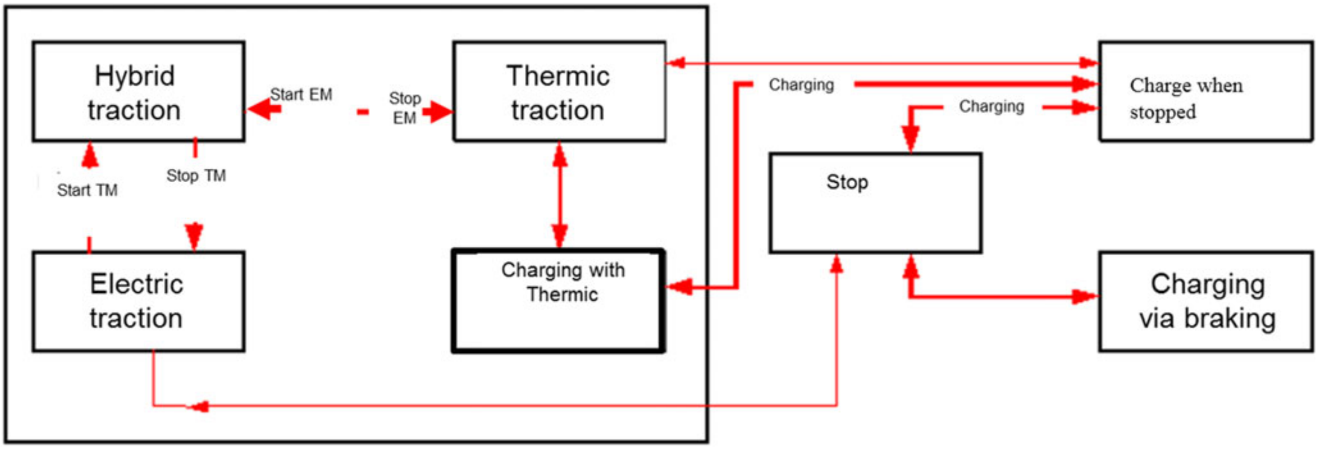


Fig. 9. Six functioning modes of a hybrid vehicle.

https://doi.org/10.1017/S0890060417000403 Published online by Cambridge University Press

In the other relevant works, researchers focus on using functional models in supporting computational design activities and innovation. For instance, Helms et al. (2013) aimed at developing a computational approach to support designers in the innovation process by introducing an approach to map the physical effects with the bond graph theory.

The research of Lucero et al. (2016) is focused on developing a framework to support producing analogies and different design solutions based on performance metrics related to functionality. They investigated how analogies can be implemented using performance metrics instead of linguistics. The framework proposed by Lucero et al. shares some similarities with the framework proposed in this research. Those similarities are limited to use of functional basis vocabulary (Hirtz et al., 2002) in developing the function model and mapping the function model to the bond graph elements. However, the two frameworks are different regarding the usage and capabilities. Here are some of those differences. Lucero et al. use the bond graph theory to group the performance metrics in functions in the functional basis, while the fundamental reason of using bond graph theory in DACM is being able to extract the causality of variables defining the functions. The framework of Lucero et al. seeks the innovative design solution by analogy generation across different domains, while DACM also enables the incremental innovation by providing simulation capabilities and systematic contradiction analysis. The simulation capability is not addressed in their research (Lucero et al., 2016). Generation of the cause–effect network among the variables describing the functions, qualitative and quantitative simulations, and contradiction analysis are of the most important capabilities provided in DACM.

3. METHODS

As mentioned above, the objectives pursued by the authors in this article are to tackle some of the perceived and probably real limitations of function modeling, for example, its inability to bridge the gap between function modeling and prototyping or simulation. The authors introduce the DACM framework to link the abstract representation of a system in the form of functional modeling with the behavior of this system. The DACM framework was developed to add a physics-based reasoning capability to functional modeling. This physics-based reasoning capability is used in the DACM framework to assess design options and support innovative ideation and strategic design decisions. In other words, DACM should reinforce the reflective analysis capability in the early phases of engineering development. Kahneman and Tversky (Kahneman, 2011) demonstrated that cause–effect analysis (Kistler, 2006) is the most common mechanism used by humans to react and act in the physical world. One idea supporting the development of DACM is that well-informed causal analysis can efficiently support conceptual modeling and analysis of design solutions and facilitates the use of the reflexive mode. DACM should be able to favor the slow and reflective mode of the brain and its natural tendency to classify information in the form of cause–effect relationships. DACM should offer concrete mechanisms to organize and simplify the complexity of the representation of a problem, and to propose a mechanism to simulate behavior using qualitative information analysis in early design phases.

4. THE DACM FRAMEWORK

DACM was initially developed as a specification and verification approach for complex systems (Coatanéa, 2015). The methods and theories contributing to the framework are articulated around fundamental pillars such as functional modeling, dimensional analysis, bond graph theory, causal rules, and colored hypergraphs. The DACM framework follows a step-by-step modeling and transformation process. Figure 10 shows the sequence of steps in DACM and related theoretical methods.

4.1. Steps of DACM framework

4.1.1. Step 1: Indicating the objectives of the model and defining the system of interest and its border

In the first step, the modeler explicitly provides rationales regarding the aim of the model and defines the system of interest. The approach is especially adapted to a context when the functional model is the result of a reverse engineering process. The aim of the model, in this case, is to favor incremental innovation.

4.1.2. Step 2: Function modeling

As presented in the Section 2, function modeling integrates multiple possible facets and usages. In the present article, the authors are especially interested in function modeling as a tool to represent the system architecture (INCOSE, 2012). Starting from one overall system functionality (or several functionalities) representing the system's intended objective(s), the specific usage of function modeling used in this article is to describe the sequence of the associated functions of a system or process. The approach considered in this article voluntarily takes the perspective of incremental design as an existing solution. The authors propose an interplay between function and functional structures like in the FBS models (Gero, 1990; Umeda et al., 1995) or the requirement–function–behavior–structure model (Christophe et al., 2010). From this viewpoint, the functional models result from an iterative process where functions and functional architectures are refined progressively using an existing artifact structure as a reference. The advantage of this approach is to propose an initial mechanism to limit the variability of the modeling. An element proposed by the authors in this article is to use only a limited set of functional vocabulary for modeling in the DACM framework developed in this article. The functional modeling process is controlled in this phase by using a normalized set of functional terms that directly use the functional basis introduced by Hirtz et al. (2002). Table 1 presents the selected vocabulary and the existing mapping with

https://doi.org/10.1017/S0890060417000403 Published online by Cambridge University Press

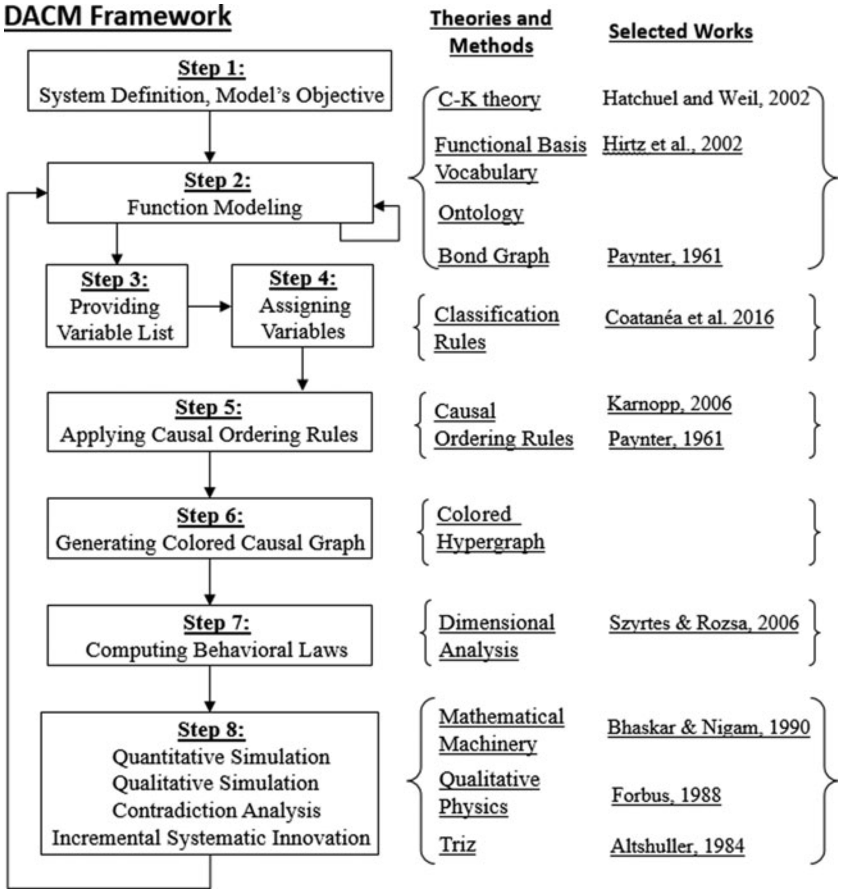

DACM Framework
Theories and Methods
Selected Works
Step 1: System Definition, Model's Objective
Step 2: Function Modeling
Step 3: Providing Variable List
Step 4: Assigning Variables
Step 5: Applying Causal Ordering Rules
Step 6: Generating Colored Causal Graph
Step 7: Computing Behavioral Laws
Step 8: Quantitative Simulation Qualitative Simulation Contradiction Analysis Incremental Systematic Innovation
C-K theory
Hatchuel and Weil, 2002
Functional Basis Vocabulary
Hirtz et al., 2002
Ontology
Bond Graph
Paynter, 1961
Classification Rules
Coatanéa et al. 2016
Causal Ordering Rules
Karnopp, 2006
Paynter, 1961
Colored Hypergraph
Dimensional Analysis
Szyrtes & Rozsa, 2006
Mathematical Machinery
Bhaskar & Nigam, 1990
Qualitative Physics
Forbus, 1988
Triz
Altshuller, 1984

Fig. 10. Modeling steps in dimensional analysis conceptual modeling framework.

Table 1. *Elementary bond graph elements used for modeling and limited associated functional basis*

Possible Names of Functions to Describe Organs	Bond Graph Elements
To transform effort into flow or flow into effort	Resistor (R)
To resist effort or flow	Resistor (R)
To transform flow into displacement	Capacitor (C)
To store displacement	Capacitor (C)
To transform displacement into effort	Capacitor (C)
To provide effort	Capacitor (C)
To transform effort into momentum	Inertia (I)
To store momentum	Inertia (I)
transform momentum into flow	Inertia (I)
To provide flow	Inertia (I)
To transform input effort into output effort of another magnitude	Transformer (TF)
To transform input flow into output flow of another magnitude	Transformer (TF)
To transform input effort into output flow of another magnitude	Gyrator (GY)
To transform input flow into output effort into output effort of another magnitude	Gyrator (GY)
To connect efforts of different magnitudes when flows are similar	Junction (JE/JF)
To connect flows of different magnitudes when efforts are similar	Junction (JE/JF)
To provide a constant effort	Source (SE/SF)
To provide a constant flow	Source (SE/SF)

https://doi.org/10.1017/S0890060417000403 Published online by Cambridge University Press

elementary building blocks from bond graph theory (Karnopp et al., 2012) used to represent the structure.

The bond graph modeling approach is a method conceived by Paynter (1961). It is a domain-independent graphical description of the dynamic behavior of physical systems. A classical bond graph model is expressed via a set of nine elementary elements. The nine elements are as follows: effort source (Se), flow source (Sf), inertial elements (I), capacitive elements (C), resistive elements (R), transformer elements (TF), gyrator elements (GY), effort Junction (0), and flow junction (1). Each of those nine elements has a predefined causality. Figure 11 represents the predefined causality of the main bond graph elements. By analyzing the causality and nature of each bond graph element, we extracted the list of possible functions in Table 1. In resistor elements, for instance, the nature of the element indicates the function "To resist effort or flow" and the predefined causality satisfies the function of "To transform effort into flow or flow into effort."

4.1.3. Step 3: Providing fundamental variable list

In the context of bond graph theory, the variables, regardless of the energy domain, are categorized into three main categories. Table 2 shows these main categories, together with their associated secondary categories of variables. The mathematical relation between generic variables describes how those variables relate to each other (Karnopp et al., 2012).

In each energy domain, "displacement" is the result of integration of the "flow" over time. For example, in the electrical domain, the electrical current is measured in amperes, which is equal to the charge per second. Equation (1) indicates that the integration of the electrical current over time is equal to the charge (q). The charge is equivalent to the "displacement" in the electrical domain.

Table 2. *Fundamental categories of variables*

Primary Category of Variables	Secondary Category of Variables
Overall system variables	Energy (En) Efficiency rate (η)
Power variables (P)	Generalized effort (E) Generalized flow (F)
State variables	Generalized displacement (D) Generalized momentum (M) Connecting variables (C)

$$\int I.dt = q. \tag{1}$$

The generalized *momentum* is the result of integration of *effort* over time. As an example, the flux linkage (known as momentum), is defined as in Eq. (2), where U (known as effort) is the potential difference between two terminals of an electrical element.

$$\int U.dt = \lambda. \tag{2}$$

The *connecting* variables proposed by Coatanéa et al. (2016) cover the other variables that are not in the four mentioned categories (effort, flow, momentum, and displacement) and are used to describe the material properties, geometry, dimensions, and so on. The connecting variables are often the design variables that a designer can select to influence the design. The connecting variable relates, for example, *effort* and flow together. For instance, consider Ohm's law in Eq. (3), which indicates the relation between voltage and cur-

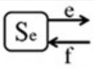
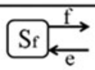
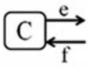
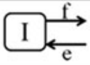
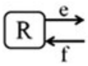
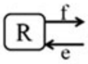
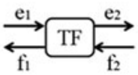
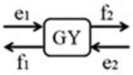
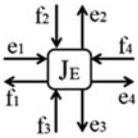
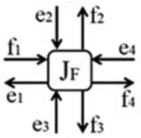
Bond Graph Element	Schematic view	Bond Graph Element	Schematic view
Source of effort (Se) Fixed effort-out causality		**Source of flow (Sf)** Fixed flow-out causality	
Capacitor (C) Fixed effort-out causality		**Inertia (I)** Fixed fow-out causality	
Resistor (R) Preferable effort-out causality (Resistive)		**Resistor (R)** Preferable flow-out causality (Conductive)	
Transformer (TF) Maintain incoming causality (two-port element)		**Transformer (TF)** Switch incoming causality (two-port element)	
Effort Junction (JE) or (0) (multi-port element) Effort Conservative $e_1 = e_2 = e_3 = e_4$ $f_1 + f_2 + f_3 + f_4 = 0$		**Flow Junction (JF) or (1)** (multi-port element) Flow Conservative $f_1 = f_2 = f_3 = f_4$ $e_1 + e_2 + e_3 + e_4 = 0$	

Fig. 11. Causality in the main bond graph elements.

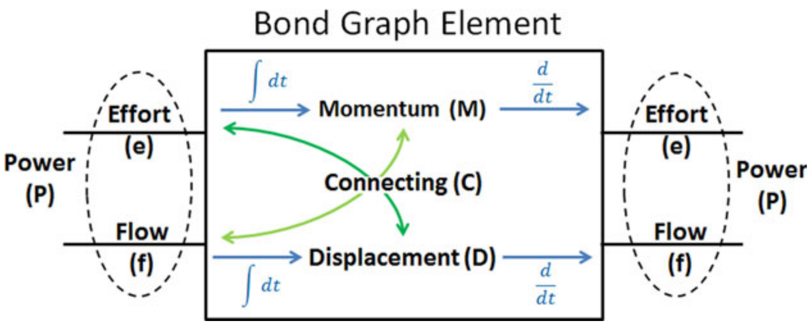


Fig. 12. Representation of the generic variables and their interconnections in the bond graph context.

rent in a conductor. The potential difference (U) is proportional to the product of electrical current (I). The connecting variable (R), known as resistance, creates this relation between effort and flow.

$$U = IR. \tag{3}$$

The efficiency is a dimensionless variable (the so-called Pi-number in Step 7). It is defined between input and output variables that have the same dimensions. Consider the power efficiency, which is defined as the ratio of output power divided by the input power. Equation (4) shows this relation.

$$\pi_{\eta p} = P_o \, . \, P_o^{-1}. \tag{4}$$

Figure 12 visualizes these relations, where the state variables, such as momentum, connecting, and displacement, are located inside the elements and the power variables are located outside.

The categories of and relations between the variables explained above are domain independent. Table 3 illustrates the mapping between the different types of energies and the categories of generic variables. The complementary information on the dimensions of the variables is also represented in the table. For each energy domain, the pair of effort and flow defines the power. In other terms, the effort multiplied by the flow produces the power. For instance, in the translational mechanical domain, *force*" and *linear velocity* define the mechanical power, while in the hydraulic domain, *pressure* and *volumetric flow rate* characterize the hydraulic power.

4.1.4. Step 4: Assigning variables

In order to be able to present the causal graph based on the function model, we need first to assign variables to the functional structure. In this step, the fundamental variables provided in the last section are assigned to the functional structure. The categorization of the variables proposed in Step 3 facilitates the systematic assignment of variables. The power variables are located outside the bond graph "boxes," and the state variables are located inside the boxes.

4.1.5. Step 5: Applying causal ordering rules

This step of the DACM process is fundamental. During this phase, the cause–effect relationships between variables are defined in the form of a causal graph. The algorithm presented in Figure 13 generates the cause–effect relationships between variables by considering multiple causal rules derived from bond graph theory (Karnopp et al., 2012). The principle of the algorithm is detailed in Figure 13. The algorithm starts by identifying the modeling problem and the initial function model proposed by the modeler. A one-to-one mapping (so-called bijective mapping) maps each function in the functional structure with one of nine bond graph elements. If this mapping is not bijective, it means the functional

Table 3. *Mapping table between the types of energies and specific names of the variables with the associated units and dimensions*

Emergy Domain	Generalized Effort	SI Units	Dimensions	Generalized Flow	SI Untts	Dimensions
Human	Force	Newton	MLT^{-2}	Velocity	m/s	LT^{-1}
Biological	Pressure	Pascal	$ML^{-1}T^{-2}$	Volumetric flow rate	m^3/s	L^3T^{-1}
Electrical	Voltage	Volt	$ML^2T^{-3}A^{-1}$	Current	ampere	A
Hydraulic	Pressure	Pascal	$ML^{-1}T^{-2}$	Volumetric flow rate	m^3/s	L^3T^{-1}
Mechanical (rotational)	Torque	Nm	ML^2T^{-2}	Angular velocity	rad/s	T^{-1}
Mechanical (translational)	Force	Newton	MLT^{-2}	Linear velocity	m/s	LT^{-1}
Chemical	Affinity	J/mol	$M^2L^2T^{-2}mol^{-1}$	Reaction rate	mol/L/s	$L^{-1}T^{-1}mol$
Pneumatic	Pressure	Pascal	$ML^{-1}T^{-2}$	Volumetric flow rate	m^3/s	L^3T^{-1}
Optical	Intensity	W/m^2	MT^{-3}	Velocity	m/s	LT^{-1}
Magnetic	Magnetomotive force	A-turns	A	Magnetic flux rate	Wb/s	$ML^2T^{-2}A^{-1}$
Thermal	Temperature difference	Kelvin	t	Entropy flow rate	J/ks	$ML^2T^{-3}t^{-1}$

https://doi.org/10.1017/S0890060417000403 Published online by Cambridge University Press

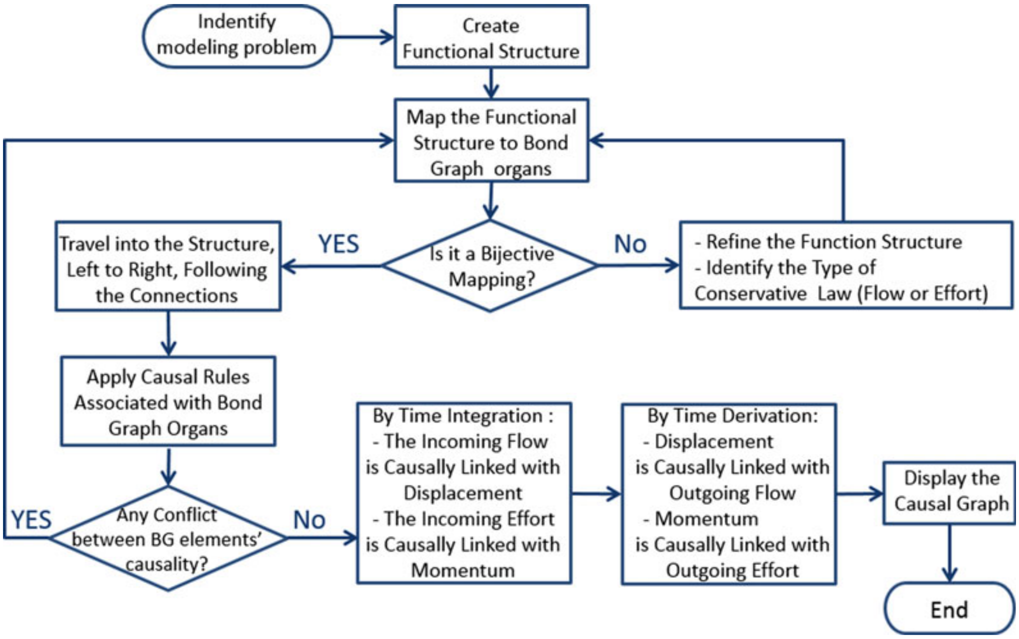

Indentify modeling problem
Create Functional Structure
Map the Functional Structure to Bond Graph organs
Is it a Bijective Mapping?
YES
No
Travel into the Structure, Left to Right, Following the Connections
- Refine the Function Structure
- Identify the Type of Conservative Law (Flow or Effort)
Apply Causal Rules Associated with Bond Graph Organs
Any Conflict between BG elements' causality?
YES
No
By Time Integration :
- The Incoming Flow is Causally Linked with Displacement
- The Incoming Effort is Causally Linked with Momentum
By Time Derivation:
- Displacement is Causally Linked with Outgoing Flow
- Momentum is Causally Linked with Outgoing Effort
Display the Causal Graph
End

Fig. 13. Description of the causal ordering algorithm.

structure requires more refinement. An iterative process is considered until it ensures that each function is mapped with one, and only one, bond graph element among the possible elements. Afterward, the algorithm applies the causal rules to the element one by one to cover the functional structure completely. If any conflict between causalities is detected, the algorithm goes back to the step where the bond graph elements were attributed to the functional structure. Otherwise, the process continues to complete the causality between variables. The algorithm finally generates a causal graph from the extracted cause–effect relationships between the variables.

4.1.6. Step 6: Generating a colored causal graph

The DACM framework colors the causal graph generated in the previous step. The variables are classified into four main classes, depending on the border of the system of interest. Colors are associated with each variable. The color code follows.

Exogenous variables. They are imposed onto the system. They are part of the environment of the system. The exogenous variables also cover the variables with no degree of freedom in changing their values. Therefore, the designer cannot modify them unless the border of the system is changed. In general, the physical and mechanical properties of the material are examples of exogenous variables. The exogenous variables are shown in black in a causal graph.

Independent design variables. This variable is not influenced by any other variable in the system. Designers can modify the value of design variables before the other type of variables. This variable can be selected during the design process. The independent variables are shown in green.

Dependent design variables. This variable, colored blue, is influenced by other variables and is thus more difficult to control than independent design variables. This variable can be selected during the design process.

Performance variables. They are a special class of dependent design variables. They are important for the overall performance evaluation of the system. The designers try to optimize them by minimizing (min.), maximizing (max.), or obtaining a target value (target). The performance variables are shown in red.

4.1.7. Step 7: Computing behavioral laws

In Step 7, two types of behavioral laws are computed. The first type of behavioral laws is equations in the junctions in the form of algebraic summation and equality between variables. A template for this kind of equation for the junction is shown in Figure 11. These equations are extracted from the detailed functional structure. The other equations are calculated on the basis of the causal graph using dimensional analysis, described below.

Dimensional analysis. Dimensional analysis proposes an approach that reduces the complexity of modeling problems to the simplest form before going into more details with any type of qualitative or quantitative modeling or simulation (Bridgman, 1969). Dimensional analysis (DA) theory has been developed over the years by an active research community including prominent researchers in physics and engineering (Maxwell, 1954; Matz, 1959; Barenblatt, 1996). The fundamental interest of DA is to deduce certain constraints on the form of the possible relationship between variables from the study of the dimensions of the variables (i.e., length, mass,

https://doi.org/10.1017/S0890060417000403 Published online by Cambridge University Press

time, and the four other dimensions of the international system of units) used in models. For example, in the most familiar dimensional notation, learned in high school or college physics, force is usually represented as [MLT^{-2}]. Such a dimensional representation is a combination of mass (M), length (L), and time (T). Newton's Law $F = m \cdot a$ with F (force), m (mass), and a (acceleration) is constrained by the dimensional homogeneity principle. This dimensional homogeneity is the most familiar principle of the DA theory and can be verified by checking the dimensions on both sides of Newton's law. The other result widely used in DA is Vashy–Buckingham's Π theorem, stated and proved by Buckingham in 1914 (Barenblatt, 1996). This theorem identifies the number of independent dimensionless numbers that can characterize a given physical situation. The method offers a way to simplify the complexity of a problem by grouping the variables into dimensionless primitives. Every law that takes the form $y_o = f(x_1, x_2, x_3, \ldots, x_n)$ can take the alternative form:

$$\prod_0 = f\left(\prod_1, \prod_2, \ldots, \prod_n\right). \tag{5}$$

Here, Π_i are the dimensionless products. This alternative form is the final result of the DA and is the consequence of the Vashy–Buckingham theorem. A dimensionless number is a product that takes the following form:

$$\pi_k = y_i . x_j^{\alpha_{ij}} . x_l^{\alpha_{il}} . x_m^{\alpha_{mi}}, \tag{6}$$

where x_i are called the repeating variables, y_i are named the performance variables, and α_{ij} are the exponents. Equation (6) presents the dimensionless form of reusable modeling primitives, used intensively to develop the framework presented in this research work. Examples of those primitives are present in multiple domains of science. For example, the efficiency rate, the Reynolds number, and the Froude number are some example of dimensionless primitives. As a result of the last step, the DACM software tool generates the governing laws of the system automatically. The example below briefly exemplifies the construction of a governing law for a small causal graph. From the causal graph shown in the figure below, it is possible to construct the matrix in Table 4. This initial matrix contains all the influencing variables, together with their associated dimensions. The target variable (power) is in the first column and the entire set of cause variables in the other columns.

Table 4. *Matrix derived from the causal graph shown in Figure 14*

Π Power	Power	T	ω
Mass	1	1	0
Length	2	2	0
Time	−3	−2	−1

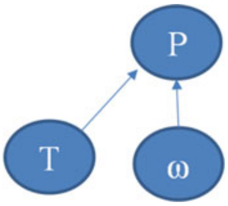


Fig. 14. Causal graph.

The initial matrix should be separated into two submatrices [**A**] and [**B**] in a manner that [**A**] always remains a nonsingular square matrix, and [**B**] contains the variable for which we are seeking a dimensionless product equation. The condition of the nonsingularity of matrix [**A**] necessitates excluding any nondimensional variables from the matrix, and combining columns or rows to create a nonsingular square matrix. If a line of [**A**] + [**B**] is totally null, then this line can be removed and the rank of [**A**] then diminishes. If the linear composition affects the lines, the system of the unit can be changed to move to composed units. These are usually sufficient to remove the problem. For example, in this case, we need to combine two rows of the initial matrix in order to have a square matrix [**A**]. Table 5 shows the split matrices where two rows are combined to make [**A**] a square matrix.

The next step consists of computing the exponent of the dimensionless numbers presented in Eq. (2). To achieve this task, the following formula taken from Szirtes and Rozsa (2006) is used:

$$[\mathbf{C}] = -([\mathbf{A}]^{-1}.[\mathbf{B}])^{T}. \tag{7}$$

Matrix [**C**] is the vector matrix representing the exponents $\alpha_{i1}, \alpha_{i2}, \alpha_{i3}$. Figure 15 illustrates the algorithm for computing the behavioral laws from the causal graph. The algorithm identifies the dependent and performance variables in the causal graph and creates the initial matrix containing its influencing variables. It follows the same principles explained above to present the behavioral laws in the form of Pi-numbers. The iterative process continues to cover all the dependent and performance variables of the causal graph.

$$\mathbf{C} = -(\mathbf{A}^{-1}.\mathbf{B})^{\mathbf{T}} = -\left(\begin{bmatrix} \frac{1}{3} & 0 \\ \frac{-2}{3} & -1 \end{bmatrix} . \begin{bmatrix} 3 \\ -3 \end{bmatrix}\right)^{T} = [-1 \quad -1], \tag{8}$$

Table 5. *Split matrices containing influencing variables*

	[B]	[A]	
Π Power	Power	T	ω
Mass × Length	3	3	0
Time	−3	−2	−1

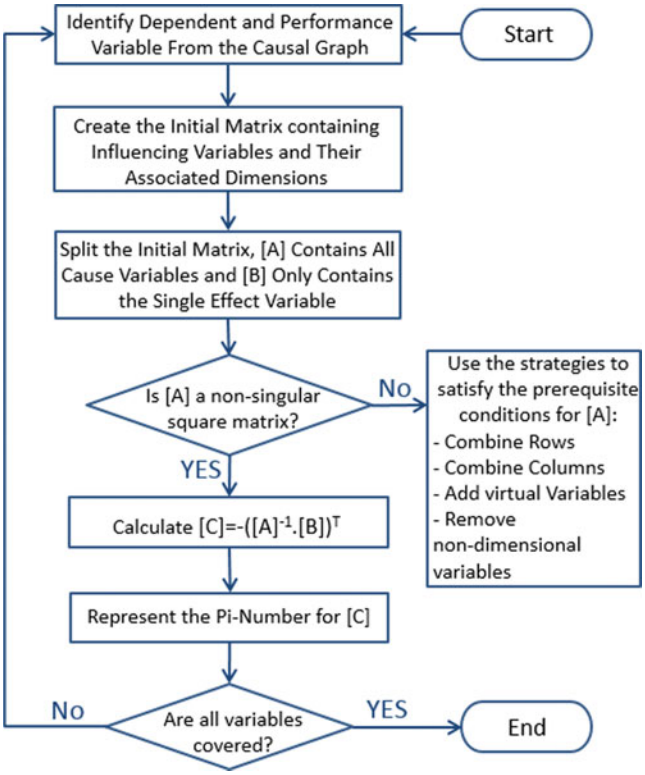

Start
Identify Dependent and Performance Variable From the Causal Graph
Create the Initial Matrix containing Influencing Variables and Their Associated Dimensions
Split the Initial Matrix, [A] Contains All Cause Variables and [B] Only Contains the Single Effect Variable
Is [A] a non-singular square matrix?
No
Use the strategies to satisfy the prerequisite conditions for [A]:
- Combine Rows
- Combine Columns
- Add virtual Variables
- Remove non-dimensional variables
YES
Calculate [C]=-([A]-1.[B])T
Represent the Pi-Number for [C]
No
Are all variables covered?
YES
End

Fig. 15. Description of the behavioral law computation algorithm.

$$\pi_{\text{Power}} = Power.T^{(-1)}.w^{(-1)}. \tag{9}$$

4.1.8. Step 8: Analysis: qualitative simulation, quantitative simulation, contradiction analysis, and incremental systematic innovation

By the end of Step 7, the functional structure has been transferred to the casual graph between the influencing variables and a set of governing equations. This step is dedicated to analyzing the whole model. The DACM framework enables quantitative and qualitative simulations (Forbus, 1988). It can be used as a systematic approach to find the weaknesses and contradictions of the system, which facilitates the incremental systematic innovation. This step assigns objectives to the performance variables colored in red in the causal graph generated in Step 6. Using the simulation machinery, those objectives are propagated backward in the causal graph. The objectives can be qualitative (i.e., maximizing or minimizing). The propagation of the objectives in the causal graph may generate contradictions (Ring, 2014). For example, the resulting objective of the propagation can lead to variables that should simultaneously answer contradictory objectives (Warfield, 2002). In order to understand how the dimensionless primitives and causal graphs that are generated are used in this work in qualitative simulation, let us consider the causal relations between energy (E) in joules, J; power (P) in watts; and time (t) in seconds. A causal graph can be established between those variables considering the relations presented in Figure 16.

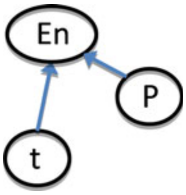


Fig. 16. A small causal graph representing the relation between energy, time, and power.

From this causally oriented graph, a dimensionless product can be constructed using Eq. (2) and Eq. (10) can be formed. The mathematical machinery developed by Bhaskar and Nigam (1990) to reason about a system of causal relationships is used in this article. A dimensionless product can be expressed in the general form, below. Equation (11) can be divided by x_j to form Eq. (12).

$$\pi_{En} = En.t^{-1}.P^{-1}, \tag{10}$$

$$y_i = \pi_k.x_j^{-\alpha_{ij}}.x_l^{-\alpha_{il}}.x_m^{-\alpha_{mi}}, \tag{11}$$

$$\frac{y_i}{x_j} = \pi_k.\frac{x_j^{-\alpha_{ij}}}{x_j}.\frac{x_l^{-\alpha_{il}}}{x_j}.\frac{x_m^{-\alpha_{mi}}}{x_j}. \tag{12}$$

From Eq. (12), a partial derivative can be written involving the variable y_i and the variable x_j and taking the following form:

$$\frac{\partial y_i}{\partial x_j} = -\pi_k.\alpha_{ij}\frac{x_j^{-\alpha_{ij}}}{x_j}.x\frac{x_l^{-\alpha_{il}}.x_m^{-\alpha_{mi}}}{x_j}. \tag{13}$$

The partial derivative can be reformulated and simplified by replacing Eq. (12) into Eq. (13); we then obtain Eq. (14):

$$\frac{\partial y_i}{\partial x_j} = -\alpha_{ij}\frac{y_i}{x_j}. \tag{14}$$

From Eq. (14), the sign of the derivative $(\partial y_i)/(\partial x_j)$ can be determined by simply verifying the sign of the exponent α_{ij}. This simple machinery provides a powerful approach for propagating qualitative optimization objectives (maximize, minimize) in a causal network. Let us take the small example shown in Figure 16, in which we define the initial objective of minimizing the energy (E_n). What should the resulting objectives for the power (P) and the time (t) be? By using Eq. (10), we can obtain two partial derivatives:

$$\frac{\partial E_n}{\partial P} = 1\frac{E_n}{P}, \tag{15}$$

$$\frac{\partial E_n}{\partial t} = 1\frac{E_n}{t}. \tag{16}$$

https://doi.org/10.1017/S0890060417000403 Published online by Cambridge University Press

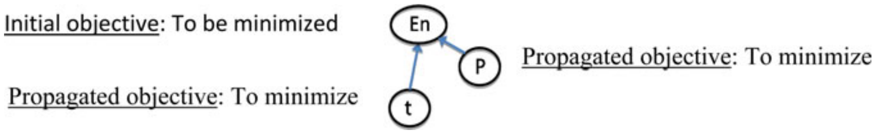

Initial objective: To be minimized
En
P
Propagated objective: To minimize
Propagated objective: To minimize
t

Fig. 17. Backward propagation of objectives in a causal graph representing the relation between energy, time, and power.

From Eqs. (15) and (16), it is possible to deduce that both P and t vary in the same direction as E_n as a result of the sign of the partial derivative. Consequently, if E_n needs to be minimized, it also requires P and t to be minimized. This is summarized in Figure 17. This process is named backward propagation in the article.

The principle described in this section is used to propagate qualitative objectives in a network and is exploited in the DACM framework to discover design weaknesses and to propose inventive solutions.

4.2. What are the possible design directions and their potential added values?

Once the contradictions and weaknesses of the system have been detected, one possible direction is to apply the innovative design principles to remove design contradictions (Fig. 18). The figure at the end of the article illustrates some of these innovative principles in the causal graph between variables. Some of these principles can be directly mapped with the TRIZ inventive principles (Altshuller, 1984), but not all of them. They were developed during the course of the research by analogy with historical situations from design, but also from history and biology. Another possible direction is to generate a virtual design for an experiment that takes advantage of the simulation machinery developed in the previous steps. The impact of the different variables influencing the performance variables is computed on the basis of their order of magnitude, and the variables are ranked according to their impact level. It helps in the later selection of the potentially most valuable design directions for innovation.

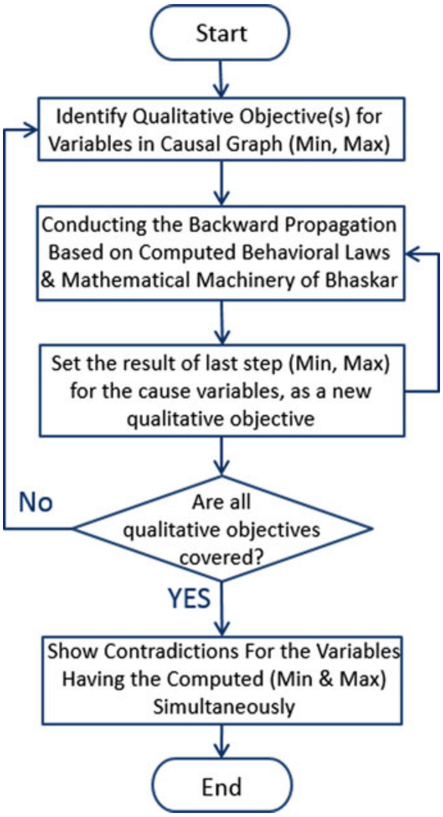

Start
Identify Qualitative Objective(s) for Variables in Causal Graph (Min, Max)
Conducting the Backward Propagation Based on Computed Behavioral Laws & Mathematical Machinery of Bhaskar
Set the result of last step (Min, Max) for the cause variables, as a new qualitative objective
No
Are all qualitative objectives covered?
YES
Show Contradictions For the Variables Having the Computed (Min & Max) Simultaneously
End

Fig. 18. Contradiction detection algorithm.

5. APPLICATION: GLUE GUN CASE STUDY

In this section, a glue gun was selected as a case study. Two basic approaches are compared initially to present the function model of the glue gun. The first approach attempts to build a function model in such a way as to avoid as far as possible having any existing architecture or design solution in mind for the modeler. This approach is presented first and it aims at demonstrating that it can be used in a new product development context too. Another interest for the author lay in demonstrating how it can lead to a different function modeling result when compared later with the purely reverse engineering approach. The reverse engineering approach is preferred in an incremental innovation process, and for this reason, the reverse engineering approach is also applied for the function modeling of the glue gun. The differences in the architectures obtained from both approaches are discussed. This article focuses in its second phase exclusively on the reverse engineering approach to demonstrate the scope of the entire DACM framework using the glue gun as an example.

In the first approach, the modeling begins with defining the boundaries of the system or artifact to be designed, recognizing different elements of the systems' environment in order to satisfy the final aim of the system. In the glue gun case, the final aim is to deliver a controlled amount of molten glue. The input material is in the form of solid glue and the output material is the molten glue. The system requires thermal energy to melt the glue. To start with similar initial conditions for both our both modeling approaches, it is assumed at first glance that the primary energy used to provide heat is electrical energy, and the mechanical energy to feed the glue stick is provided by human energy. The solid glue is used in two functions, to provide hydraulic energy to push the liquid

https://doi.org/10.1017/S0890060417000403 Published online by Cambridge University Press

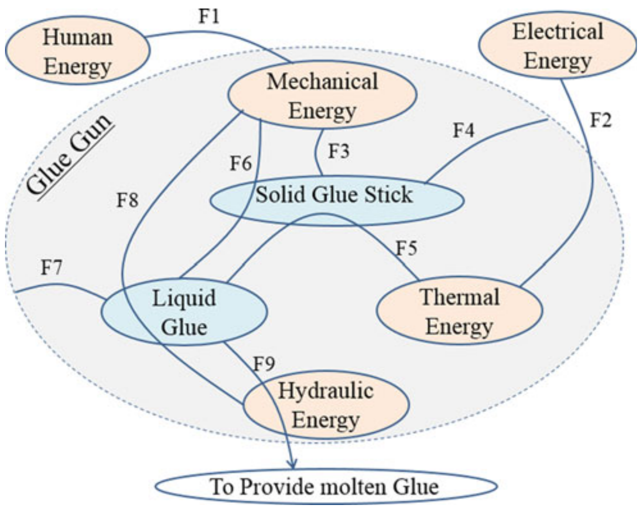


Fig. 19. Schematic view of associated functions in the glue gun.

glue out of the nozzle and also to change the state of the solid glue into liquid glue. The associated energy domains and materials in the glue gun are shown in different colors in Figure 19 (i.e., orange for energies and blue for materials). Afterward, the necessary functions are defined between different energies and materials. While some functions can only be defined between energies or between materials, some other functions need to use an energy domain to act between two materials. For example, as shown in Figure 19, the solid glue stick is transformed into a liquid glue stick (F5) by using thermal energy. The functions shown in Figure 19 are systematically defined in Table 6. Each function is also given an approximate sequence and sorted according to the order in which this function becomes active. The active functions in the same time interval in Table 6 are represented in parallel in Figure 20. The function schematic shows each function in the form of input and output. Having the input, output, and time sequence interval for each function helps us to know how the functions are connected. To relate two functions together, we need to match the output of the function to the input of the function in the next time sequence interval.

The function model based on this initial analysis is represented in Figure 20. It should be mentioned that the modeler has a significant impact on the nature of the model. For example, one might think that the pressure on the liquid glue stick can be provided by directly pushing the glue stick by hand or by an indirect action performed on the glue stick, or even by a specific device generating pressure on the liquid glue.

The second approach used to present the model of the functional architecture is a purely reverse engineering approach. In this approach, existing design architecture is available, and the role of the modeler is to represent the functional model using the existing system as a reference. Later in this section, the function model will be used in the DACM framework to take the function modeling a step forward toward increasing the usability of the function modeling. The sequence of the required steps in the DACM framework was explained in the Section 3 (see Fig. 10).

5.1. Step 1: System definition

The modeling begins with systems definition and the purpose of the modeling. In this case study, the system is the whole glue gun, including its components, and the purpose of the modeling is to present a model supporting the later physics-based reasoning.

5.2. Step 2: Function modeling

A black box model is considered; the solid glue stick is the input material, and the human energy and electrical energy are the energy inputs. The human and electrical energies are used to push and melt the glue. At the other side of the black box model of the glue gun, we have the melted glue. After the trigger has been pushed, the human force is converted to mechanical work belonging to the mechanical energy domain. The mechanical force activates a mechanism to guide the glue stick and to grip it. The glue stick is melted in a dedicated

Table 6. *Function definition for schematic view of the glue gun's associated functions*

Function	Subject	Object	By	Function Schematic	Sequence Interval
F1 (to transform)	Human energy (HE)	Mechanical energy (ME)		$\mathrm{HE} \xrightarrow{F1} \mathrm{ME}$	T1
F2 (to transform)	Electrical energy (EE)	Thermal energy (TE)		$\mathrm{EE} \xrightarrow{F2} \mathrm{TE}$	T1
F3 (to grip and move)	Mechanical energy (ME)	Solid glue stick (SG)		$\mathrm{SG, ME} \xrightarrow{F3} \mathrm{SG, ME}$	T2
F4 (to guide)	Solid glue stick (SG)		Body component (BC)	$\mathrm{SG} \xrightarrow{F4} \mathrm{BC}$	T2
F5 (to transform)	Solid glue stick (SG)	Liquid glue (LG)	Thermal energy (TE)	$\mathrm{TE, SG} \xrightarrow{F5} \mathrm{TE, LG}$	T3
F6 (to create pressure)	Solid glue stick (SG)	Liquid glue (LG)	Mechanical energy (ME)	$\mathrm{ME, SG} \xrightarrow{F6} \mathrm{ME, LG}$	T3
F7 (to guide and contain)	Liquid glue (LG)		Container (C)	$\mathrm{LG} \xrightarrow{F7} \mathrm{C}$	T3
F8 (to transform)	Mechanical energy (ME)	Hydraulic energy (HyE)		$\mathrm{ME} \xrightarrow{F8} \mathrm{HyE}$	T4
F9 (to provide)	Liquid glue (LG)	Liquid glue (LG)	Hydraulic energy (HyE)	$\mathrm{LG, HyE} \xrightarrow{F9} \mathrm{LG}$	T5

https://doi.org/10.1017/S0890060417000403 Published online by Cambridge University Press

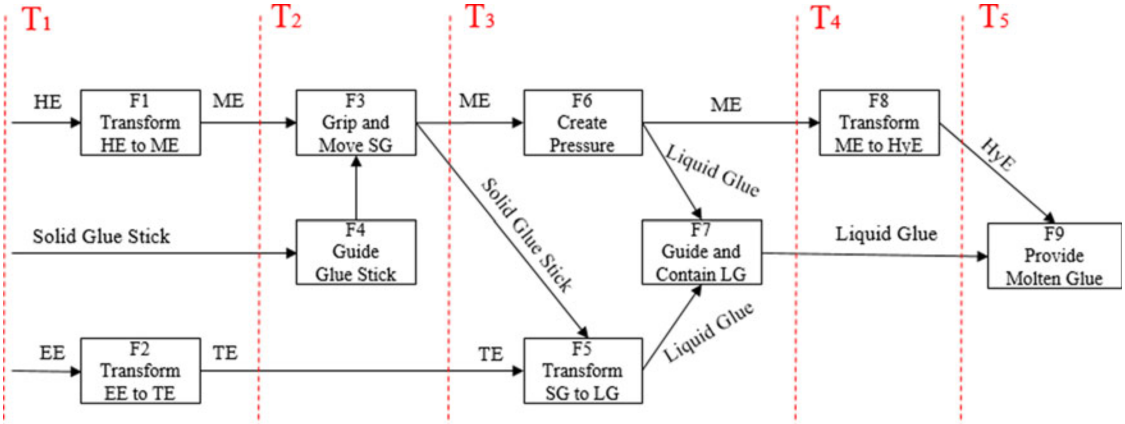


Fig. 20. Glue gun function model based on function schematic interaction.

area of the glue gun, and a compressed spring provides the backward movement of the trigger, allowing a new push on the trigger. On the other side, the electrical energy is converted into thermal energy, and this thermal energy melts the glue stick. A part of this thermal energy is also dissipated in the environment. The modification of the state of the glue from solid to liquid leads the energy to change from mechanical energy to hydraulic energy. The initial function model is represented in Figure 21. In the preparation of the function model that follows, a limited set of vocabulary is used. This vocabulary was presented in Table 1 in the Section 3. Figure 21 depicts a model resulting from numerous iterations. The limited vocabulary is used to converge in terms of representation details. It should be noted that the function model shown in Figure 21 is slightly different from the function model shown in Figure 20. The model is different in its architecture, and in Figure 20, the system used to pull back the trigger is not present. This main difference is that Figure 20 was not trying to abstract from any specific solution. The architecture in Figure 21 is also more detailed. This is due to the forced usage of a limited set of function vocabulary pushing the modeler to detail more the model.

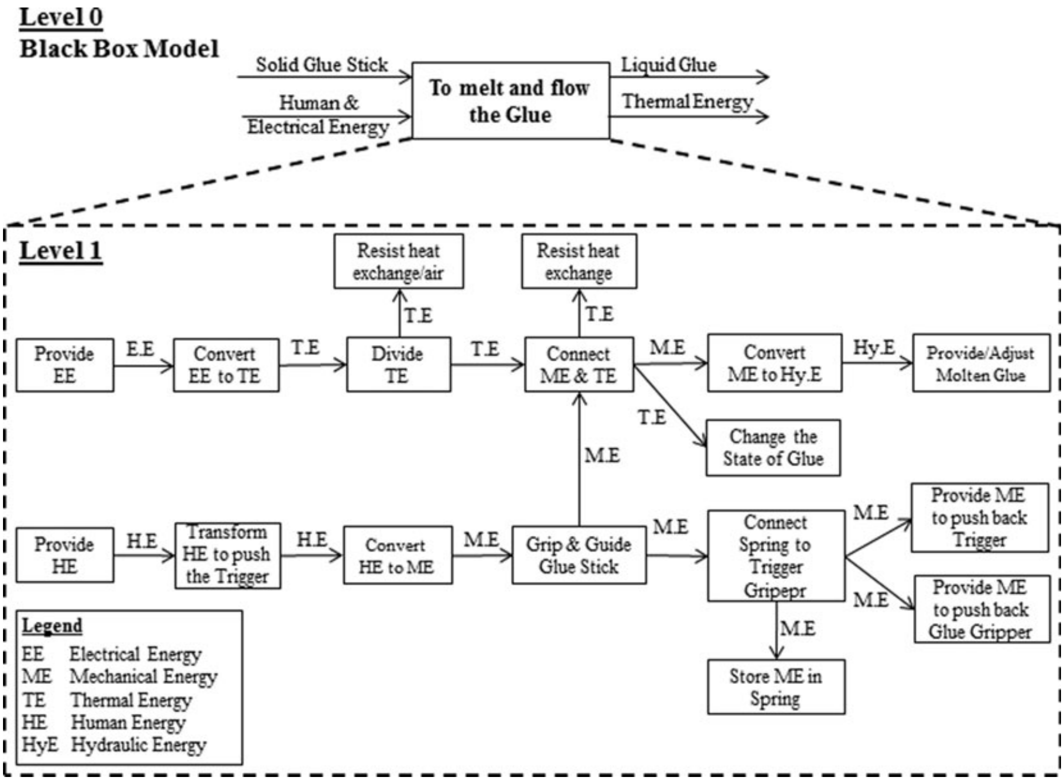


Fig. 21. Initial function model of the glue gun.

https://doi.org/10.1017/S0890060417000403 Published online by Cambridge University Press

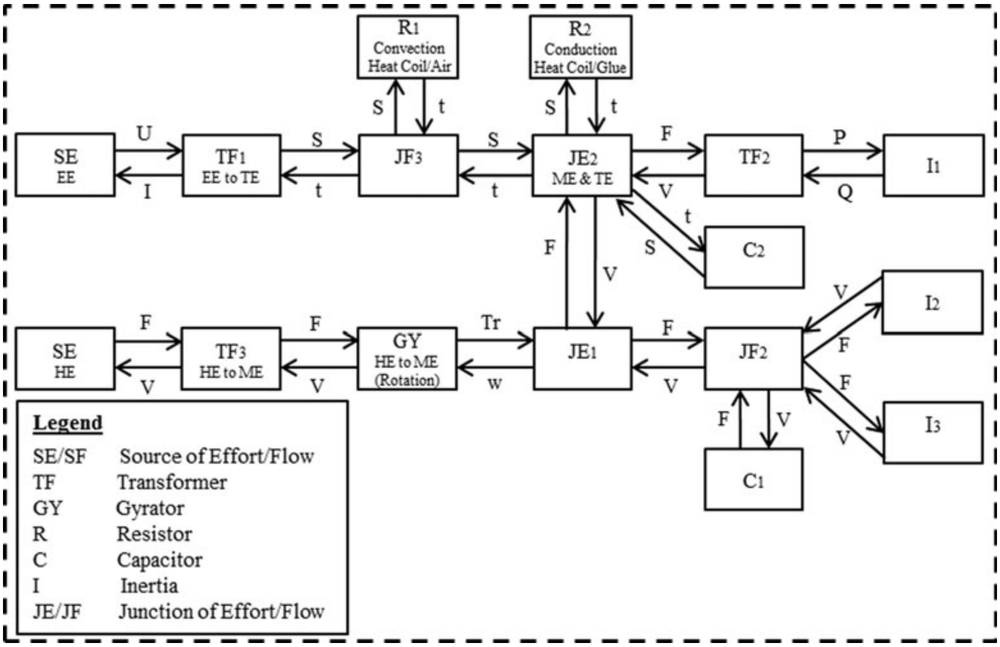


Fig. 22. The initial bond graph model mapped from the initial function model of the glue gun.

The model in Figure 21 is used as a reference, and the DACM framework is now developed further. The functional boxes in the function model are mapped to the bond graph elements (Fig. 22). This is done by performing a one-to-one mapping between the functional representation shown in Figure 21 and the bond graph elements presented in Table 1. Knowing the primary function name provides a limited set of choices between bond graph elements. For instance, "connect" offers two possible bond graph elements: "flow junction" and "effort junction." There is a clear difference between these junctions. A flow junction is used when we have flows of equal values between different "pipes" connecting at junctions. An effort junction is applied if the efforts arriving at an interface are equal. This kind of physics-based reasoning should be performed for each functional box to decide between the types of junctions to be integrated. These choices are fundamental to the validity of the final model. These choices are automated in the online platform developed to model the use of the DACM framework.

Table 7. *Mapping from functional vocabulary to possible choice of bond graph elements*

Verbs Used in Glue Gun FM	Primary Function Vocabulary	Possible Choice in Bond Graph Elements
To connect	Connect	JE, JF
To convert	Convert	TF, GY
To divide	Branch	JE, JF
To guide	Channel	JE, JF
To provide	Provision	SE, SF, I, C
To resist	Magnitude	TR, R, I, C
To store	Provision	SE, SF, I, C
To transform	Convert	TF, GY
To change	Magnitude	TR, R, I, C
To absorb	Provision	SE, SF, I, C

Table 1 and Table 7 represent the one-to-one mapping of the initial function model into a bond graph representation. Analyzing the initial bond graph reveals that the conversion of human force to mechanical force is performed by a "transformer" (TF_4) element. The transformer converts translational force and velocity to torque and angular velocity, while we will need translational force and velocity after JE_1. So the use of an additional transformer between the (TF_4) and (JE_1) elements is essential. Having this kind of critical physics-based reasoning in the bond graph can show if any required element is missed. Moreover, it will give an insight into the design solution even before attributing a physical artifact to the functions. It should be noted that the human force could be transferred to mechanical force with a single transformer if a direct human force is applied in the same direction as the movement of the glue stick. This is not the case in the architecture of the glue gun. This is consequently requiring a transformation of a linear movement into a rotational movement using a second transformer. This process requires satisfying the coherence of the model with the real glue gun architecture, and when this is obtained, it is possible to move to the next phase.

Now let us have a more detailed look at the thermal elements of the model (Fig. 22). The effort and flow variables are temperature and entropy flow rate, respectively. The multiplication of flow by effort shows the instantaneous power. Nevertheless, the entropy is not conservative and not directly measurable either. For those reasons, applying a true bond

https://doi.org/10.1017/S0890060417000403 Published online by Cambridge University Press

graph in the thermal domain is not straightforward. A pseudo bond graph, initially developed by Karnopp (1979), offers the possibility of using the heat flow rate instead of the entropy flow rate to characterize the flow. It provides more flexibility for presenting equations without losing the advantages of bond graph elements. Thus, another modification of the initial function model is to present the thermal aspect of the model in a pseudo bond graph. In a pseudo bond graph, the "heat flow rate" and "temperature" characterize the flow and effort, respectively. A pseudo bond graph can be systematically applied to a thermal process. With the hypothesis that the temperature is distributed nonhomogenously in the parts, each part is represented in a flow junction or with a divided flow function. The temperature remains constant at each contact surface. The interaction or the contact surface between parts or between parts and the surroundings is shown by the effort junction or with the connect function. In other words, we will have an effort junction between the flow junctions. The resistor and capacitor elements are connected via a flow junction and effort junction, respectively. The resistor element characterizes the heat transfer by conduction and convection.

Once again, the latest function model should be mapped to the bond graph elements using Table 7. In the next step, the variables are assigned to the bond graph representation.

5.3. Steps 3 and 4: Providing and assigning variables

Each energy flow between two functional boxes is mapped to variables in two categories: flow and effort. The flow and effort variables are selected on the basis of the energy domain in Table 1. In order to be able to distinguish easily between the different variables in the same category, a number is attributed to the repetitive variables. In addition to the flow and effort variables, some of the bond graph elements require the variables to be assigned inside the elements to generate the causality and define the characteristic of the element. As mentioned in the Section 3, the inside variables are categorized into the three categories of "displacement," "momentum," and "connecting." The use of connecting variables is a major difference with the bond graph approach. This type of variable is allowing to really designing a system and not simply modeling the dynamic of a system. Table 8 represents the influencing variables of the model, together with their associated dimensions and categories.

Figure 23 depicts the pseudo bond graph representation, mapped from the modified function model of the glue gun shown in Figure 24. The influencing variables are assigned to the pseudo bond graph representation. The transformer TF_5 is added between TF_4 and JE_1 on the mechanical side of the model in comparison with the last bond graph represen-

Table 8. *System variables with associated dimensions and categories*

Parameters	Symbol	Unit	Dimension	Category
Electric potential	U	Volt	$ML^2T^{-3}A^{-1}$	Effort
Electric current	I	Ampere	A	Flow
Velocity (feed rate)	V	m/s	LT^{-1}	Flow
Force	F	N	MLT^{-2}	Effort
Volume flow rate	Q	m^3/s	L^3T^{-1}	Flow
Pressure	P	N/m^2	$ML^{-1}T^{-2}$	Effort
Torque	Tr	N m	ML^2T^{-2}	Effort
Angular velocity	w	rad/s	T^{-1}	Flow
Melted glue viscosity	μ	Pa.s	$ML^{-1}T^{-1}$	Momentum
Temperature difference	t	°C	t	Effort
Entropy flow rate	S	W/°C	$ML^2T^{-3}t^{-1}$	Flow
Heat flow rate	f	J/s	$ML^{-2}T^{-3}$	Flow
Temperature	e	°C	t	Effort
Stiffness coefficient	K	Kg/s^2	MT^{-2}	Connecting
Glue gun nozzle diameter	d	m	L	Displacement
Coefficient of conduction	K_c	W/(m°C)	$MLT^{-3}t^{-1}$	Connecting
Glue stick diameter	D	m	L	Displacement
Coefficient of convection	H	$W/(m^2°C)$	$MT^{-3}t^{-1}$	Connecting
Coil heat exchange surface	A	m^2	L^2	Displacement
Glue gripper mass	M_1	kg	M	Connecting
Trigger mass	M_2	kg	M	Connecting
Mass of glue stick in coil	M_3	kg	M	Connecting
Glue stick and glue density	ρ	kg/m^3	ML^{-3}	Connecting
Mass flow rate	MFR	kg/s	MT^{-1}	Flow
Transformation modulus	n	—	—	—
Specific heat capacity	C_p	J/kg k	$L^2T^{-2}\ t^{-1}$	Connecting
Duration of function	ΔT	s	T	Connecting
Ambient temperature	t_a	°C	t	Effort

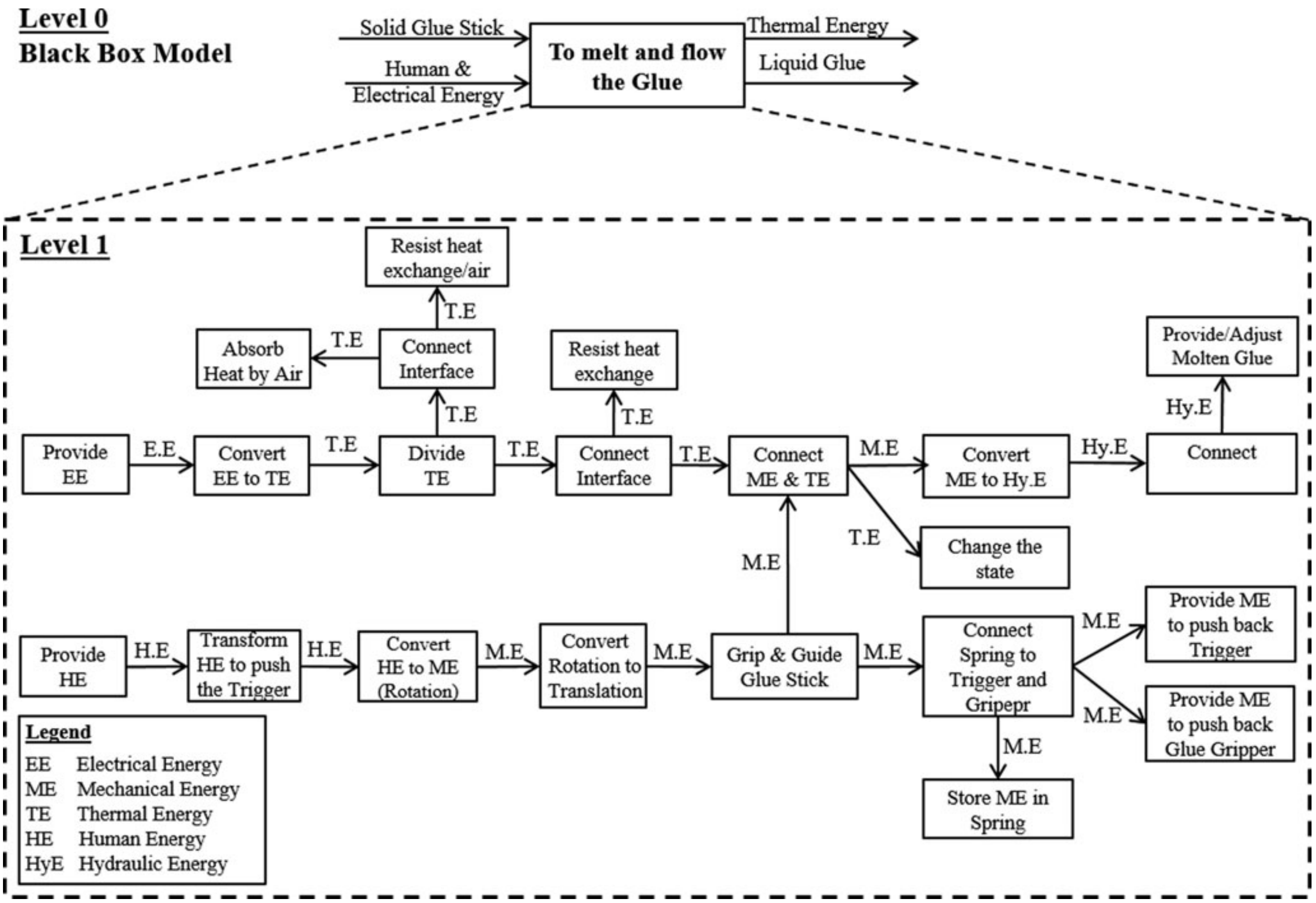


Fig. 23. Pseudo bond graph representation filled with variables.

tation. The transformer (TF_4) transforms the translational force (F) and velocity (V) into torque (Tr) and angular velocity (w). From the other side of the model, the electrical energy and the difference in temperature between adjacent parts cause the heat flow. JE_3 and JE_2 represent the heating coil and glue stick. JF_4 maps the heat exchange surface between “outer surface of coil and surroundings,” and JF_3 shows the heat exchange surface between “the inner surface of the heating coil and the glue stick.” In addition, (e_5) shows the temperature difference between the outer surface of the coil (e_2) and the ambient temperature ($e_4 = t_a$). The convection coefficient (H), heat surface exchange (A), and temperature difference (e_5) characterize the heat convection between the heat coil and the surroundings in the (R_1) element. In the same manner, the difference in temperature between the inner surface of the coil and the glue stick (e_6), conduction coefficient (K), and glue stick diameter (D) characterize the heat conduction between the heating coil and the glue stick in the (R_2) element.

5.4. Steps 5, 6, and 7: Generating a colored causal graph and computing behavioral laws

Using the predefined causality rules for each bond graph element (see Fig. 11), a general causal graph and the governing equations are generated. This is a second major difference with the traditional bond graph approach. Colors are used, and the equations are derived using the DA theory (Coatanéa et al., 2016). Identifying incoming and outgoing variables will enable us to form the causal graph. An additional rule is that the exogenous variables are always the cause of outgoing variables. The causal graph shows the relation between variables in terms of cause and effect in a visual manner, and the set of equations relate the variables in a mathematical manner.

Figure 25 represents the causal graph of the glue gun model. The governing equations extracted are the causal graph and are listed in Table 9. DA is used to present equations between variables in the form of the product (Coatanéa et al., 2016). The other equations, in the form of equalities and summation of variables, are extracted from the junctions.

5.5. Step 8: Qualitative simulation, contradiction analysis, and incremental innovation

Once the causal graph and set of governing equations of a problem are ready, the contradiction analysis can be performed. The contradiction analysis starts with choosing the qualitative objective of the performance variables (i.e., red variables) of the system. In the case of the glue gun, finding

https://doi.org/10.1017/S0890060417000403 Published online by Cambridge University Press

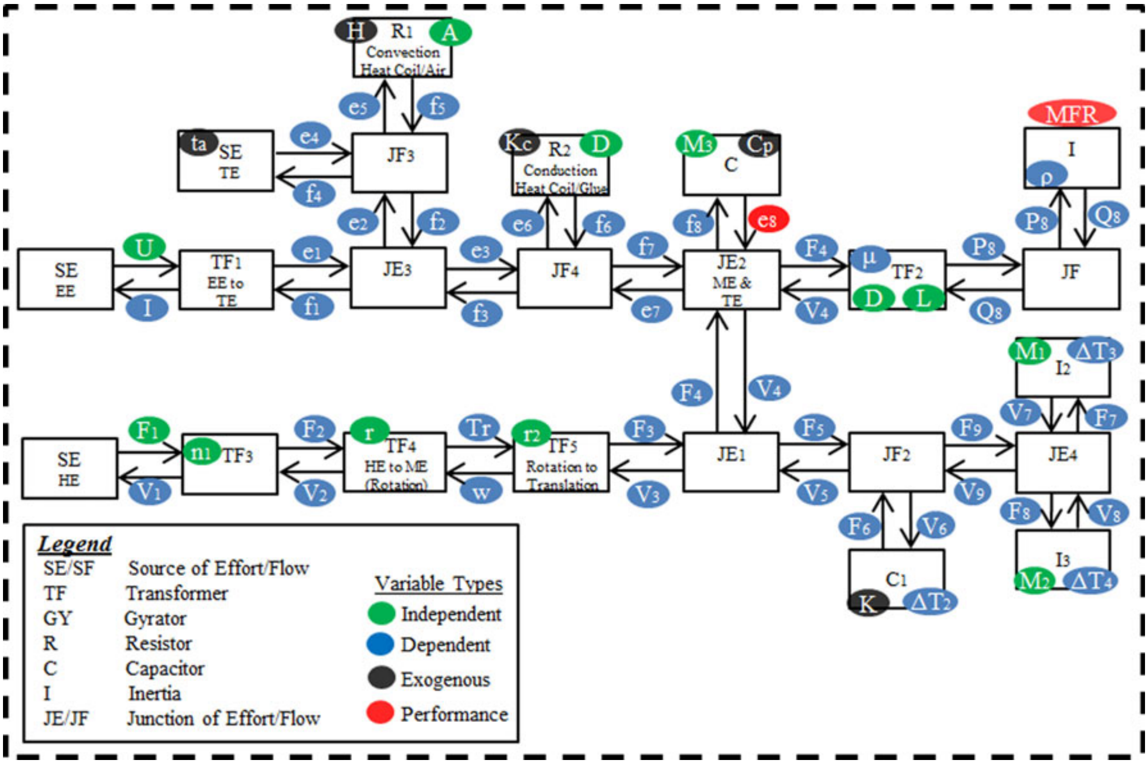


Fig. 24. Modified function model of the glue gun.

a design solution that lets the user provide molten glue with less effort and less energy consumption is desirable. Less energy consumption is partly related to the insulation condition of the system. However, it is also related to the final temperature of the output molten glue. Therefore, minimizing the output temperature of the molten glue (i.e., minimizing "e_8") to a few degrees above its melting point can be considered to be the first qualitative performance. If the user can have a higher material flow rate while pressing the trigger, this satisfies the desired need to have molten glue with less human effort. Increasing the output material flow rate (i.e., maximizing MFR) is considered to be the second qualitative performance in this study. It should be mentioned that different qualitative performances can be considered that are based on different aims. The backward propagation as the result of considering the two above-mentioned performances is shown in the causal graph in Figure 26.

Let us analyze the result of backward propagation shown in Figure 26. Maximizing the flow rate of molten glue (MFR) requires the volume flow rate (Q) to be maximized. Volume

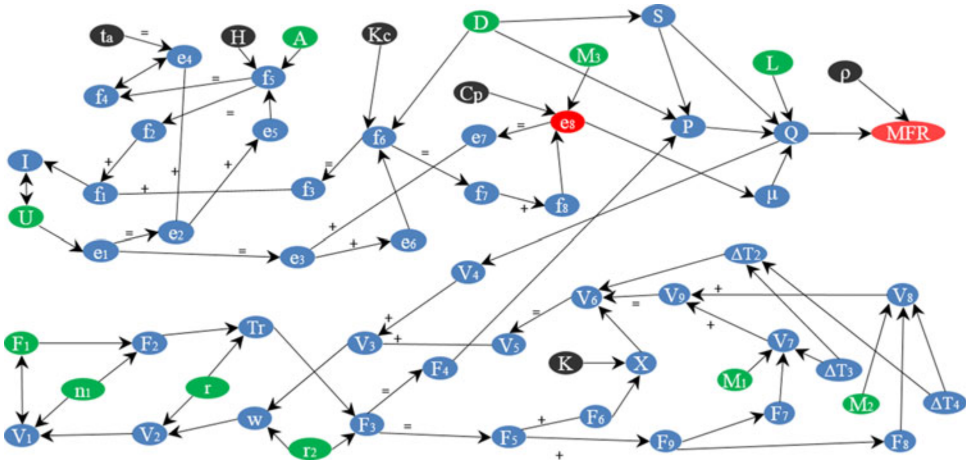


Fig. 25. Causal graph of the glue gun.

Table 9. *Governing equations extracted from causal graph and dimensional analysis*

$\pi_{f_1} = C_1 . f_1 . e_1 . I^{-1} . U^{-1}$	$\pi_{Tr} = C_5 . \mathrm{Tr} . F_2^{-1} . r^{-1}$
$e_1 = e_2 = e_3$	$\pi_{V_2} = C_6 . V_2 . w^{-1} . r^{-1}$
$f_1 = f_2 + f_3$	$\pi_{F_3} = C_7 . F_3 . \mathrm{Tr}_2^{-1} . r_2^{-1}$
$e_1 = t_1$	$F_3 = F_4 = F_5$
$f_2 = f_4 = f_5$	$V_3 = V_4 + V_5$
$e_2 = e_5 + e_4 \quad \text{OR} \quad e_5 = e_2 - e_4$	$F_5 = F_6 + F_9$
$e_4 = t_a$	$V_5 = V_6 = V_9$
$\pi_{f_5} = C_2 . f_5 . e_5^{-1} . H^{-1} . A^{-1}$	$F_7 = F_8 = F_9$
$f_3 = f_6 = f_7$	$V_9 = V_7 + V_8$
$e_3 = e_6 + e_7 \quad \text{OR} \quad e_6 = e_3 - e_7$	$\pi_{V_7} = C_8 . V_7 . F_7^{-1} . M_1^{-1} . \Delta T_3^{-1}$
$\pi_{f_6} = C_3 . f_6 . e_5^{-1} . K^{-1} . D^{-1}$	$\pi_{V_8} = C_9 . V_8 . F_8^{-1} . M_2^{-1} . \Delta T_4^{-1}$
$f_7 = f_8$	$\pi_P = C_{11} . P . F_4^{-1} . D^2$
$\pi_{f_8} = C_4 . e_8 . f_8^{-1} . M . C_p$	$\pi_Q = C_{12} . Q . \mu . P^{-1} . D^{-2} . L^{-1}$
$F_2 = n_1 . F_1$	$\pi_{\mathrm{MFR}} = C_{13} . \mathrm{MFR} . \rho^{-1} . Q^{-1}$
$V_2 = V_1 / n_1$	

flow rate depends on multiple variables such as pressure (P), viscosity (μ), and the volume of molten glue [cross section (S) and length (L)]. Maximizing the flow rate requires maximizing or minimizing one or several of these variables. To increase the volume of molten glue (Q), we need to increase the pressure on the glue stick (P), and/or the cross section of the glue stick (S). Increasing the pressure has a direct relation with applied force, and a reverse direction with the glue stick cross section. Therefore, the first contradiction is detected, as we need to minimize and maximize the cross section (S) simultaneously (see Fig. 26). In contrast, based on the casual graph, to minimize the viscosity (μ), the temperature of molten glue (e_8) should be increased. The latter is in contradiction with the second objective of the study, which is to decrease the temperature (e_8). The backward propagation of the second objective (minimizing e_8) also indicates that the glue stick diameter (D) should be minimized to melt the glue stick faster and reduce the energy consumption. Figure 26 illustrates the backward propagation of the two qualitative objectives on the causal graph. The contradictions are highlighted on the causal graph. The result of the contradiction analysis and the visual causal relation between variables will guide where the designer should search for an idea to innovate or to improve the performance of the system. In addition to the principles of TRIZ (Altshuller, 1999), the inventive principles based on the causal graph are presented in Figure 27. To suppress the contradiction and to present an innovative solution in the current case study, we used Principle 9. Principle 9 exists among the TRIZ principles and basically suggests dividing the object into other objects. Therefore, using several glue sticks with a small diameter can solve the contradictions in this case study. The pressure is applied to an area, which is equal to the sum of the cross sections of the glue sticks. Smaller diameter of glue sticks enables the faster melting and the sum of the cross sections can be increased without interfering with the fast melting condition. As a result of providing a new solution for suppressing the contradiction, the causal graph should be updated to see if the proposed solution causes another contradiction in the system or not.

6. CONCLUSION

The paper presented an approach developed in order to tackle some of the issues limiting the usage of functional modeling in the engineering design world. The key objective of this article was to demonstrate that an extension of the capability domain of function modeling can be developed to build physics-based reasoning models. The DACM framework presented in this paper is a concrete method to implement the theoretical FBS models presented in the literature. Other characteristics associated with function modeling, such as the variability of the models produced by different modelers, have been analyzed in this paper. In this article, since the DACM framework predominantly starts from a reverse engineering

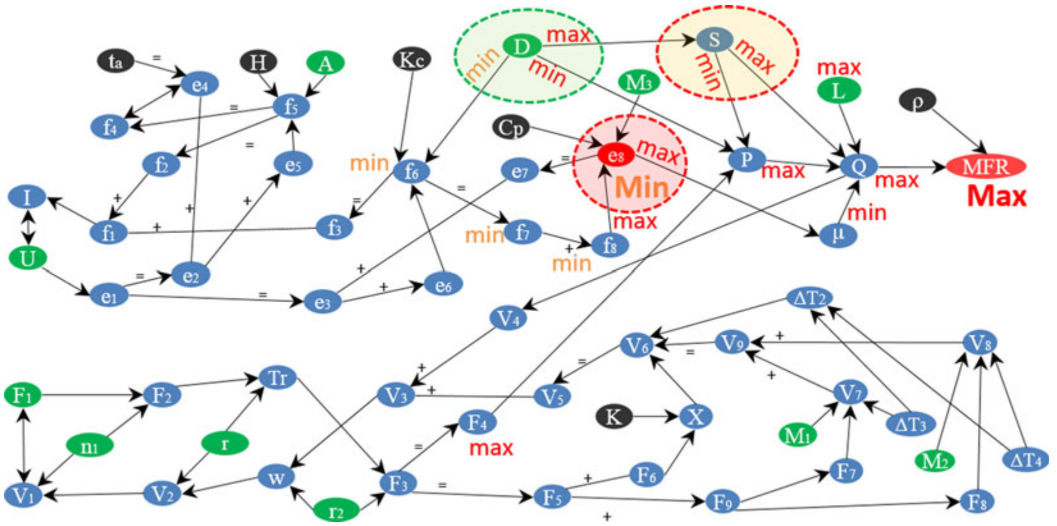


Fig. 26. Contradiction analysis in the causal graph.

https://doi.org/10.1017/S0890060417000403 Published online by Cambridge University Press

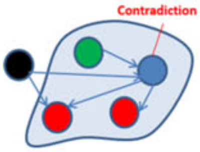
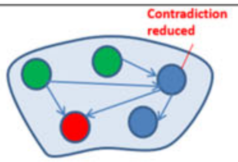
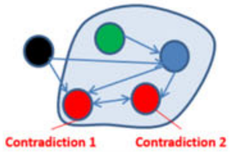
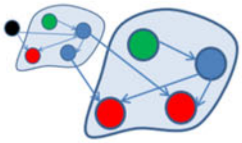
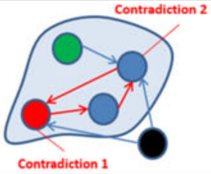
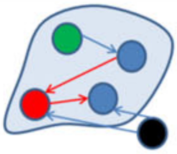
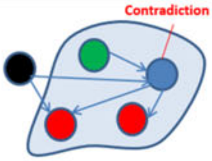
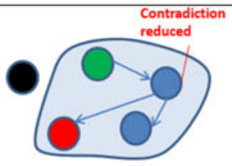
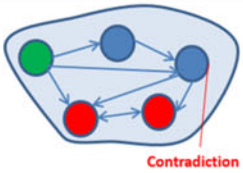
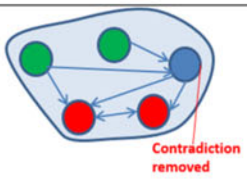
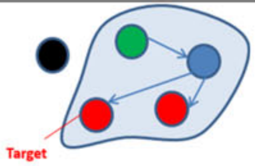
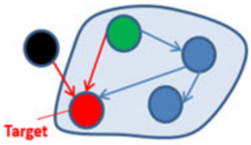
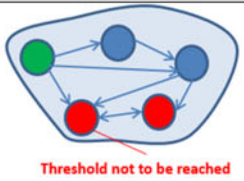
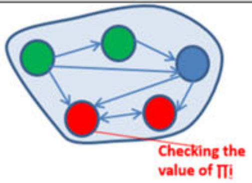
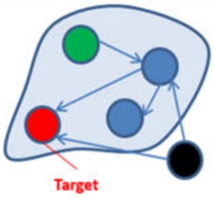
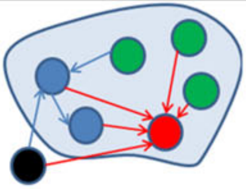
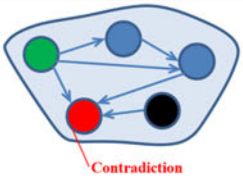
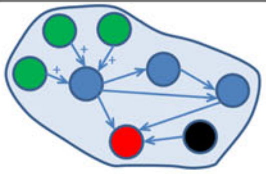
	Before applying the principle	After applying the principle	Principle
Principle 1: Border expansion			The borders of the system are expanded by transforming exogenous variables into design variables.
Principle 2: System integration or cooperation			Combining multiple systems to act on the contradictions or conflicts.
Principle 3: Redesign of graph for decoupling			Redesigning the graph to remove loops that cause contradictions or conflicts.
Principle 4: Avoid or limit the impact of exogenous variables			Limit or remove the impact of exogenous variables on the variables that cause the contradiction or conflicts.
Principle 5: Limit the number of dependent design variables (blue) that are difficult to control			Redesign the network to try to replace dependent design variables (blue) with independent design variables (green).
Principle 6: Develop connectivity			Develop connectivity to make the target more resilient and robust.
Principle 7: Check the potential threshold value in your network			Using power laws verifying that threshold values π_i that can change the behavioural domain of a system are not attained.
Principle 8: Develop diversity and redundancy			If the performance variables have to reach target values or if the performance variables are critical, the redundancy and multiplicity of the variables affecting the performance variables have to be designed.
Principle 9: Segmentation			The variables in the causal graph are segmented to remove the contradiction.

Fig. 27. Some inventive principles for the causal graph.

https://doi.org/10.1017/S0890060417000403 Published online by Cambridge University Press

situation, it is required that the function model that is generated matches the structure of the artifact being analyzed. The final model produced by DACM is heavily dependent on the quality of the functional model. Nevertheless, DACM is not limited to be used in incremental design innovations. For this reason, different mechanisms have been proposed to refine the functional model progressively and to insure convergence in the direction of a single functional model. The DACM approach cannot yet guarantee that different modelers will obtain a similar functional model at the end of the DACM process. A study of case studies involving several modelers is required as a continuation of the research to develop further the DACM approach. The authors of the article are also developing an online platform for DACM in which an ontology combined with AI-based tools has been developed to support the initial function modeling process.

The case study has exemplified the usage of DACM in the context of the glue gun example. The design objectives retained by the authors for the glue gun were to diminish the energy consumption and the manual effort on the trigger to be employed by the user of the glue gun. Other design objectives can be considered, such as, for example, increasing the output flow rate of the glue gun or adjusting the temperature of the glue precisely. We encourage readers to apply the approach in other case studies and to contact us if support is needed to use the approach. The DACM transformation and reasoning process of the initial function model are based on solid scientific grounds, as all the elements that combine to form DACM are validated approaches. The novel aspect of DACM has been integrated into a more global approach. The method has already been tested and validated in multiple case studies, and the method is currently being used in different design and manufacturing domains. The authors of the article are eager to test the approach in other fields as well.

REFERENCES

Ahmed, S., & Wallace, K. (2003). Evaluating a functional basis. *Proc. ASME Design Engineering Technical Conf.*, pp. 1–7, Chicago.

Altshuller, G. (1999). *The Innovation Algorithm: TRIZ, Systematic Innovation and Technical Creativity*, p. 312. Worcester, MA: Technical Innovation Center, Inc.

Altshuller, G.S. (1984). *Creativity as an Exact Science—Altshuller*. Philadelphia, PA: Gordon & Breach.

Barenblatt, G.I. (1996). *Scaling, Self-Similarity, and Intermediate Asymptotics: Dimensional Analysis and Intermediate Asymptotics*. Cambridge: Cambridge University Press.

Bhaskar, R., & Nigam, A. (1990). Qualitative physics using dimensional analysis. *Artificial Intelligence 45*, 73–111.

Bowler, L. (1976). Experiences and design education of librarians. *Knowledge Quest 42(5)*, 58–61.

Bridgman, P.W. (1969). *Dimensional Analysis in Encyclopaedia Britannica* (Haley, W., Ed.), Vol. 7, pp. 439–449. Chicago: Encyclopaedia Britannica.

Christophe, F., Bernard, A., & Coatanéa, E. (2010). A model for knowledge representation of conceptual design. *CIRP Annals Manufacturing Technology 59*, 155–158. doi:10.1016/j.cirp.2010.03.105

Coatanéa, E. (2005). *Conceptual modelling of life cycle design*. PhD Thesis. University of Aalto.

Coatanéa, E. (2015). *Dimensional Analysis Conceptual Modelling (DACM): A Comprehensive Framework for Specifying, Validating, and Analyzing System Models From a Model-Based System Engineering Perspective*. (Contract SOW 4.5, 4). Washington, DC: US Department of Defense, NAWCTSD Office.

Coatanéa, E., Roca, R., Mokhtarian, H., Mokammel, F., & Ikkala, K. (2016). A conceptual modeling and simulation framework for system design. *Computing in Science & Engineering 18(4)*, 42–52.

de la Bretesche, B. (2000). *La méthode APTE: Analyse de la valeur, analyse fonctionnelle*. Paris: Pétrelle.

Dusenbery, D.B. (1992). *Sensory Ecology*. New York: W. H. Freeman.

Dwyer, B., Avrunin, G.S., & Corbett, J.C. (1999). Patterns in property specifications for finite-state verification. *Proc. 1999 Int. Conf.. IEEE*, pp. 411–420, Los Angeles, May 11–14.

Eppinger, S.D., & Browning, T.R. (2012). *Design Structure Matrix Methods and Applications*. Cambridge, MA: MIT Press.

Forbus, K.D. (1988). Qualitative physics: past, present and future. In *Exploring Artificial intelligence* (Shrobe, H.E., Ed.), pp. 239–296. San Francisco, CA: Morgan Kaufmann.

Friedenthal, S., Moore, A., & Steiner, R. (2008). OMG systems modeling language. *Proc. INCOSE Int. Symp.*, 1731–1862, Utrecht, The Netherlands, June 15–19.

Gero, J.S. (1990). Design prototypes: a knowledge representation schema for design. *AI Magazine 11(4)*, 26.

Hanrahan, R.P. (1995). *The IDEF process modeling methodology*. Washington, DC: US Air Force Software Technology Support Center.

Helms, B., Shea, K., & Schulthesis, H. (2013). Automated mapping of physical effects to functions using abstraction ports based on bond graphs. *Journal of Mechanical Design 135(5)*, 1–12. doi:10.1115/1.4023923

Hirtz, J., Stone, R.B., Mcadams, D.A., Szykman, S., & Wood, K.L. (2002). A functional basis for engineering design: reconciling and evolving previous efforts. *Research in Engineering Design 13*, 65–82. doi:10.1007/s00163-001-0008-3

Hmelo-Silver, C.E., Rebecca, J., Liu, L., Gray, S., Demeter, M., Rugaber, S., Vattam, A., & Goel, A. (2008). Focusing on function: thinking below the surface of complex natural systems. *Science Scope 31(9)*, 27–35.

IEEE. (2005). *IEEE Standard 1220 for Application and Management of Systems Engineering Process*. Piscataway, NJ: IEEE Standards Association.

INCOSE. (2012). *System Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, version 3.2.2. San Diego, CA: Author.

Kahneman, D. (2011). *Thinking Fast and Slow*. New York: Macmillan.

Karnopp, D. (1979). State variables and pseudo bond graphs for compressible thermofluid systems. *Journal of Dynamic Systems, Measurement, and Control 101(3)*, 201–204.

Karnopp, D.C., Margolis, D.L., & Rosenberg, Ronald C. (2012). *System Dynamics: Modeling, Simulation, and Control of Mechatronic Systems*. Hoboken, NJ: Wiley.

Kistler, M. (2006). *Causation and Law of Mature*. London: Routledge.

Kurfman, M.A., Stone, R.B., Rajan, J., & Wood, K.L. (2003). Experimental studies assessing the repeatability of a functional modeling derivation method. *Journal of Mechanical Design 125*, *682*. doi:10.1115/1.1625400

Le Moigne, J.L. (1994). *La théorie du système général: Théorie de la modélisation*. Paris: Presses Universitaires de France.

Lucero, B., Linsey, J., & Turner, C.J. (2016). Frameworks for organising design performance metrics. *Journal of Engineering Design 27*, 175–204.

Luhmann, N. (2013). *Introduction to Systems Theory*. Cambridge: Polity Press.

Matz, W. (1959). *Le principe de similitude en Génie Chimique*. Paris: Dunod.

Maxwell, J.C. (1954). *A Treatise on Electricity and Magnetism*. Cambridge: Cambridge University Press.

Miles, L.D. (1967). *Value Engineering*. Lansberg, Germany: Verlag Moderne Industrie.

NFC03-190+R1. (1995). *NF C03-190+ R1 Norme française, Diagramme fonctionnel "GRAFCET" pour la description des system logique de commande, Union Technique de l 'électricité*. Paris: UTE Éditions.

NFX50-151. (1991). *Analyse de la valeur, analyse fonctionnelle, expression fonctionnelle du besoin et cahier des charges fonctionnel*. Accessed at http://perso.citi.insa-lyon.fr/sfrenot/cours/PI/AF_JFOREST.pdf

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

Pahl, G., & Beitz, W. (2013). *Engineering Design: A Systematic Approach*. London: Springer Science & Business Media.

Paynter, H.M. (1961). *Analysis and Design of Engineering Systems*. Cambridge, MA: MIT Press.

Pugh, S. (1991). *Total Design, Integrated Methods for Successful Product Engineering*. Boston: Addision-Wesley.

Ring, J. (2014). Discovering the real problematic situation: the first aspect of conceptual design. *Insight 17(4)*, 11–14.

Rowe, P.G. (1991). *Design Thinking*. Cambridge, MA: MIT Press.

Roza, Z. (2005). *Simulation fidelity theory and oractice*. PhD Thesis. Delft University of Technology.

Sen, C. (2011). *A formal representation of mechanical functions to support physics-based computational reasoning in early mechanical design*. PhD Thesis. Clemson University.

Sen, C., Caldwell, B.W., Summers, J.D., & Mocko, G. (2010). Evaluation of the functional basis using an information theoretic approach. *Artificial Intelligence for Engineering Design, Analysis & Manufacturing 24(1)*, 85–103.

Sen, C., & Summers, J.D. (2013). Analysis and identifying requirements for physics-based reasoning on function structure graphs. *Artificial Intelligence for Engineering Design 27*, 291–299. doi:10.1017/S0890060413000292

Simon, H.A. (1996). *The Sciences of the Artificial*. Cambridge, MA: MIT Press.

Stahel, W.R. (1997). The functional economy: cultural and organizational change. In *The Industrial Green Game: Implications for Environmental Design and Management*, pp. 91–100. Washington, DC: National Academies Press.

Suh, N.P. (1990). *The Principles of Design*. New York: Oxford University Press.

Summers, J.D., & Shah, J.J. (2004). Representation in engineering design: a framework for classification. *Proc. DETC'04 ASME 2004 Design Engineering Technical Conf. Computers and Information in Engineering Conf.*, pp. 1–10, Salt Lake City, UT.

System Engineering Fundamentals. (2013). *System Engineering Fundamentals—US Army*. Washington, DC: US Department of Defense, US Army.

Szirtes, T., & Rozsa, P. (2006). *Applied Dimensional Analysis and Modeling*, 2nd ed. Burlington, MA: Elsevier.

Tomiyama, T., Beek, V., Cabrera, T.J.A., Komto, A., & Hitoshi D'Amelio, V. (2013). Making function modeling practically usable. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 27(3)*, 301–309.

Tomiyama, T., Gu, P., Jin, Y., Lutters, D., Kind, C., & Kimura, F. (2009). Design methodologies: industrial and educational applications. *CIRP Annals Manufacturing Technology 58(2)*, 543–565. doi:10.1016/j.cirp.2009.09.003

Umeda, Y., Tomiyama, T., & Yoshikawa, H. (1995). FBS modeling: modeling scheme of function for conceptual design. *Proc. 9th Int. Workshop on Qualitative Reasoning*, pp. 271–278, Amsterdam.

VDI (1993). *VDI 2221: Systematic Approach to the Development and Design of Technical Systems and Products*. Düsseldorf: Author.

Warfield, J.N. (2002). *Understanding Complexity: Thought and Behavior*. Bentonville, AR: AJAR.

Yoshikawa, H. (1981). General design theory and a CAD system. *Man-Machine Communication in CAD/CAM* (Sata, T., & Warman, E., Eds.), pp. 35–58. Amsterdam: North-Holland

Hossein Mokhtarian is a joint doctoral student at Tampere University of Technology and Grenoble Alpes University. His research area is the modeling and simulation of additive manufacturing.

Eric Coatanéa is a Professor at Tampere University of Technology. He holds a doctorate from Aalto University in Finland and the University of West Brittany in France. Eric's research interests are system engineering, design methodologies, and manufacturing. Dr. Coatanéa is a member of the ASME society.

Henri Paris is a Professor at Grenoble Alpes University in France. His research interests are manufacturing and design methodologies. Dr. Paris is a member of the GSCOP laboratory and a member of CIRP.

https://doi.org/10.1017/S0890060417000403 Published online by Cambridge University Press