

Function modeling using the system state flow diagram

UNAL YILDIRIM, FELICIAN CAMPEAN, AND HUW WILLIAMS

Automotive Research Centre, Faculty of Engineering and Informatics, University of Bradford, Bradford, United Kingdom

(RECEIVED November 20, 2016; ACCEPTED March 31, 2017)

Abstract

This paper introduces a rigorous framework for function modeling of complex multidisciplinary systems based on the system state flow diagram (SSFD). The work addresses the need for a consistent methodology to support solution-neutral function-based system decomposition analysis, facilitating the design, modeling, and analysis of complex systems architectures. A rigorous basis for the SSFD is established by defining conventions for states and function definitions and a representation scheme, underpinned by a critical review of existing literature. A set of heuristics are introduced to support the function decomposition analysis and to facilitate the deployment of the methodology with strong practitioner guidelines. The SSFD heuristics extend the existing framework of Otto and Wood (2001) by introducing a conditional fork node heuristic, to facilitate analysis and aggregation of function models across multiple modes of operation of the system. The empirical validation of the SSFD function modeling framework is discussed in relation to its application to two case studies: a benchmark problem (glue gun) set for the engineering design community; and an industrial case study of an electric vehicle powertrain. Based on the evidence from the two case studies presented in the paper, a critical evaluation of the SSFD function modeling methodology is discussed based on the function benchmarking framework established by Summers et al. (2013), considering the representation, modeling, cognitive, and reasoning characteristics. The significance of this paper is that it establishes a rigorous reference framework for the SSFD function representation and a consistent methodology to guide the practitioner with its deployment, facilitating its impact to industrial practice.

Keywords: Functional Architecture; Functional Requirements; Function Modeling; State Diagrams; Systems Engineering

1. INTRODUCTION

1.1. Challenges for function modeling of complex systems

The multidisciplinary complexity of technical systems, such as automotive vehicles, has increased significantly with the introduction of new technologies to address evolving customer needs and trends, including autonomy and environmental concerns (Lu & Suh, 2009). The multidisciplinary nature of technical systems introduces important integration challenges (Tomiya et al., 2007; Hehenberger et al., 2016) to overcome disciplinary boundaries associated with prevalent discipline-specific approaches in conventional engineering education and design tools and methods (D'Amelio & Tomiyama, 2007). It is particularly difficult to assess and predict system interactions early in the engineering design process, which makes the management of multidisciplinary

system integration a very difficult task (Lindemann et al., 2009; van Beek & Tomiyama, 2009). This becomes an even bigger challenge with the design analysis of cyberphysical systems (Hehenberger et al., 2016) that are expected to operate autonomously in nondeterministic conditions with open architectures. This has prompted an increased interest in both industry practice and academic communities for generic (solution-neutral) function modeling approaches that can be effectively deployed early in the design process, to efficiently manage cross-disciplinary modeling supporting multiple challenges from interoperability to innovation and design for adaptability of dynamic systems (Gericke & Eckert, 2015). The work presented in this paper is focused on this challenge of developing an effective function modeling framework that supports a generic, top-down solution independent analysis of a complex systems, which is scalable across the levels of abstraction of the system.

Systems engineering approaches, in particular model-based systems engineering, have evolved driven by the need to address the overall integration of operational and functional models within a system-of-systems context, with

Reprint requests to: Felician Campean, University of Bradford, Automotive Research Centre, Faculty of Engineering and Informatics, Bradford BD7 1DP, United Kingdom. E-mail: F.Campean@Bradford.ac.uk

strong traceability, interoperability, and reuse properties. The system modeling language (SysML), which has emerged as a prevalent model-based systems engineering modeling environment (Friedenthal et al., 2008; Hoffmann, 2011), provides a graphical descriptive “language” for system modeling, focused on capturing functional requirements from the operational concept of the system across its lifecycle use cases. However, the systems engineering methods, in particular function modeling, are not linked to the established technically oriented design methodologies (Tomiyama et al., 2009; Hehenberger et al., 2016; Tomiyama, 2016).

The function modeling viewpoint taken in this paper is driven by the industry need to manage the design process of large complex systems, which have multiple operating modes, often delivered through switchable technologies to address multiple functional requirements (Liu et al., 2015). An integrative approach to architecture analysis of multidisciplinary systems requires a focus on function modeling to support system decomposition analysis (Komoto & Tomiyama, 2011). Many of the well-established function modeling schemes, including Stone and Wood (2000), Ulrich and Eppinger (2003), and Ullman (2010) are rooted in the Pahl et al. (2007) flow-based thinking, underpinned by representations of the flows of material, energy, and information through the system. The authors’ extensive observation of current engineering systems analysis practice in the automotive industry, across a number of original equipment manufacturers (OEMs) and suppliers, has revealed that such methods are mostly used to describe and represent the function of a known configuration, that is, as reverse engineering, typically applied at the subsystem or component level. The application of these methods for the system-level analysis, linking between the customer and logical layers, and across multiple disciplines including software, has been very limited. Other well-known functional modeling methods, such as the function–behavior–structure (FBS) model of Umeda et al. (1990) and the structure–behavior–function (SBF) model of Goel et al. (2009), which focus on the linkage between function and the intended behavior and structure of the system, were not seen to have a consistent application in industrial practice, which is also confirmed by observations in literature (e.g., van Beek & Tomiyama, 2009; Tomiyama et al., 2013).

1.2. Background to system state flow diagram (SSFD)

The SSFD was initially introduced to support design failure mode avoidance methodologies in industrial practice, by promoting a consistent solution-neutral function decomposition across systems engineering levels (Campean et al., 2011; Campean, Henshall, Yildirim, et al., 2013). The SSFD is underpinned by the general principle of defining functions and functional requirements in relation to the state transitions of the operand (Albers et al., 2011); that is, a function is the relation between the input and the output state of the operand. The graphical representation of the SSFD is coherent with the statecharts (Harel, 1987), in that the states are represented in boxes, joined by arrows indicating the functions defined as state transitions, as illustrated in Figure 1a. SSFD analysis starts with the definition of input and output states of the operand, conceptualized as an object, in terms of the measurable attributes or properties that describe these states. The function is defined in relation to the transformation needed to change the attribute values from the initial input to the final output state. An SSFD model is developed by decomposing the function through identification of intermediate states of the flow between the input state and the output state, as illustrated in Figure 1b. Functional models of complex systems can be created by systematic mapping of flows of energy, material, and information through the system, as illustrated in Campean et al. (2011). The SSFD methodology has been deployed in industry over the past 6 years, in particular in conjunction with two major automotive OEMs, supported through a learning intervention for a design methodology focussed on robust engineering systems design analysis (Henshall et al., 2017). Through this deployment, many engineers have been exposed to the methodology, with several hundred case studies being developed. A perceived strength of the methodology stems from the fact that it promotes engineering systems analysis focused on the identification of the observable states of the flows through the system, described in terms of measurable attributes. The subsequent articulation of functions and the associated functional requirements, being defined in relation to the state transitions, tend to be more precise and maintain a solution-neutral articulation of the function. This addresses common difficulties observed in industrial practice associated with conventional methods, which require the function to be

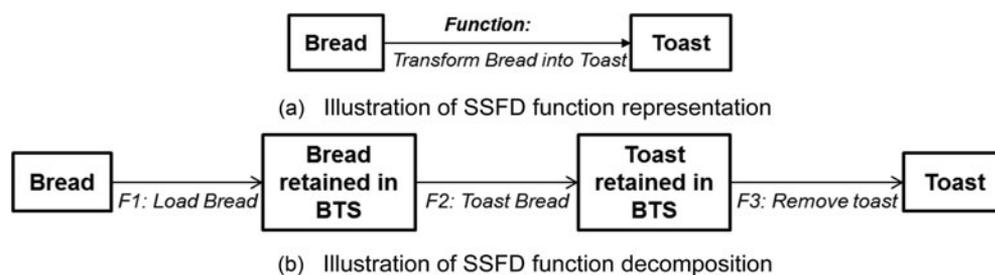


Fig. 1. Illustration of system state flow diagram for a bread toasting system.

defined first, typically using a verb–noun structure. This often leads to imprecise function articulation and modeling, with no consistent reference to measurable attributes. A further strength of the SSFD is that it can be applied to the function modeling and representation across different disciplines (including software), and allows combination of multiple flows, as illustrated in the case studies reported in the literature (Campean et al., 2011; Campean, Henshall, & Rutter, 2013; Henshall et al., 2014, 2015; Dobryden et al., 2017; Naidu et al., 2017).

With the growing evidence of its application, the need to enhance the rigor of the SSFD methodology has also become clear, to boost its broader academic and practical impact. Based on a critical review of the SSFD methodology against the established function modeling frameworks, Yildirim (2015) discussed some key aspects of the methodology that require improvement:

1. A rigorous definitions for the key elements of the SSFD methodology is required, in particular, “state” and “function,” along with consistent convention for their graphical representation for the SSFD functional model.
2. A consistent set of guidelines is needed to govern the representation of the combinations of flows and operation modes in the same diagram to support the compact representation of complex multidisciplinary systems with multiple states and modes of operation.
3. A set of consistent practitioner guidelines for the deployment of the SSFD to the analysis of complex systems needs to be provided to ensure that the effect of the analyst’s subjectivity on function modeling is minimized (Maier & Rechten, 2000).

1.3. Research aim and methodology

This paper presents underpinning research aiming to systematically address these requirements for the development of the SSFD methodology. The research aims to establish a rigorous, comprehensive, and coherent framework for function modeling of complex systems based on the SSFD, covering

1. definitions and conventions for the representation of the elements of the SSFD functional model,
2. a consistent methodology for the deployment of the SSFD,
3. empirical validation of the SSFD function modeling methodology via a benchmark problem and an industrial case study, and
4. a critical evaluation of the SSFD function modeling methodology based on the benchmarking framework introduced by Summers et al. (2013).

The significance of the research presented in this paper is that it establishes a rigorous academic basis for the SSFD function modeling, together with a coherent methodology to facilitate

and guide the application of the SSFD framework in engineering systems analysis and design practice.

This research was set in the context of the existing function modeling frameworks and methodologies, to provide the basis for a critical discussion and analysis underpinning SSFD methodology development. Driven by the focus of this paper on considering function modeling to support a systematic top-down decomposition analysis of a complex system, a framework for the classification of function modeling methods is introduced focusing on the approaches to developing functional chains (defined as a set of connected functions that capture the decomposition structure at a given hierarchical level). This analysis is presented in Section 2 of the paper.

A generic issue noted in relation to the existing functional modeling approaches is that they generally focus on modeling of one mode of operation at a time. Given that complex multidisciplinary systems often have multiple interconnected modes of operation, there is a fundamental need for a functional framework to support the solution-independent representation and analysis across multiple modes of operation. This has been taken as a requirement for the development of the SSFD methodology.

Building on the critical review of the existing function modeling frameworks, a rigorous basis for SSFD functional modeling is developed. This addresses the key elements of the SSFD, that is, the conventions for states definition and representation, and the function definition and representation scheme. This work is presented in Section 3 of the paper.

In order to support the practical implementation of the SSFD functional modeling approach, research and analysis leading to the development of a set of heuristics to guide the practitioner in carrying out systematic function analysis and decomposition of complex systems is presented next. Similar to Otto and Wood (2001), SSFD heuristics are developed to address key operations on flows: identification and decomposition of the main flow; connecting flows; and branching flow. In addition, a conditional fork node heuristic is introduced in order to address the practical need to aggregate functional models for complex multidisciplinary systems that have multiple modes of operation connected through logical or parametric conditions.

The SSFD heuristics collectively define a consistent methodology and framework for conducting function analysis of technical systems. In order to validate this methodology, the glue gun benchmark problem set for the engineering design community by Eckert and Summers (2014) is analyzed. To provide a full validation of the SSFD heuristics framework, the analysis considered both the basic glue gun recommended as a benchmark, and a higher specification glue gun that offers low-temperature and high-temperature operation modes. This detailed analysis is presented in Section 4 of the paper.

To augment the empirical evidence basis for the validity of the SSFD as a design methodology based on the validation square framework of Pedersen et al. (2000), an industrial

case study of the application of the SSFD methodology to an electric vehicle powertrain (EVP) is presented in Section 5 of the paper.

Based on the evidence from the two case studies presented in the paper, a critical evaluation of the SSFD as a functional modeling methodology is presented in Section 6, underpinned by the function benchmarking framework established by Summers et al. (2013). This analysis, presented in Section 7 of the paper, considers all the dimensions of the benchmarking framework, that is, representation characteristics, modeling characteristics, cognitive dimensions, characteristics, and reasoning characteristics.

2. FUNCTION MODELING BACKGROUND

Erden et al. (2008) used *function modeling* to refer to the activity of developing models of systems based on their functionalities and the functionalities of their subcomponents. Several reviews of function modeling approaches have been presented in the literature over time, encompassing different viewpoints. King and Sivaloganathan (1998) classified existing function analysis methods and techniques into five areas of application: value analysis, failure analysis, concept analysis, artificial intelligence, and function classification. Chiang et al. (2001) presented a review of function modeling approaches with a strong focus on the way of reasoning and representing functions. Erden et al. (2008) have classified function modeling approaches by using 17 criteria organized under the six headings of ontology, semantic definition of function, function representation formalism, function–context relation, decomposition and verification, and implementation in a programming environment and application. The review of Srinivasan et al. (2012) focused on the chronology of the development of function definitions and function representations, and deduced four views of function: level of abstraction, requirement–solution, system–environment, and intended–unintended. Summers et al. (2013) proposed three dimensions of function modeling approaches: representation characteristics, supported cognitive dimension characteristics, and enabled reasoning activities. Eisenbart (2014) emphasized the disciplinary differences in function modeling approaches, that is, mechanical engineering, electrical engineering, software development, service development, mechatronic system development, product–service systems design, and systems engineering.

Modern engineering systems present a high degree of complexity, and rely on the fulfillment of a number of interconnected function chains achieving different system functions. This emphasizes the practical importance of identification and characterization of function chains in a structured way. From the point of view of the approach to function decomposition into function chains, function modeling approaches can be classified into four perspectives:

1. task oriented,
2. flow oriented,

3. function–structure oriented, and
4. FBS oriented.

The following subsections outline the analysis of function modeling approaches based on this framework.

2.1. Task-oriented approaches

Task-oriented approaches represent the flows of functions through the system with respect to causality in the chain sequence, focusing on structure-independent description of what needs to be achieved. The functional flow block diagram (NASA, 2007; Chourey & Sharma, 2016) provides a network structure of functions of the system in a sequential relationship that leads to the achievement of overall system function. The functional flow block diagram is well supported by tools, for example, the SysML activity diagram (Friedenthal et al., 2008). The function analysis system technique (Kaufman & Woodhead, 2006), commonly used in industrial practice through brainstorming-based techniques, promotes the top-down decomposition of a main function into subfunctions, by iteratively applying how–why–when questions of the higher level function. The integrated function modeling (Eisenbart, 2014; Eisenbart et al., 2016) represents the functional model of a system through six connected views: use case view, state view, interaction view, actor view, effect view, and process flow view. The process flow view represents the sequence of processes required for the fulfillment of a use case, illustrating both sequential and parallel transformation processes.

Task-oriented approaches promote a hierarchical decomposition for system functions. However, they lack a structured approach to the development of the function chains describing the achievement of function at a certain level of the hierarchy, and the combination of chains within the overall functional structure of the system.

2.2. Flow-oriented approaches

Pahl et al. (2007) underpin the development of “function structures” by introducing a concise taxonomy for the flows through a system in terms of energy, material, and signal (information). The basis for function representation is the block diagram (i.e., black box) in which the function is articulated in verb–noun structure. The development of the function model starts by identifying the main flow, followed by the identification of the auxiliary flows with their subfunctions (Pahl et al., 2007). The basic principles of the function modeling approach proposed by Pahl et al. underpin the functional basis of Stone and Wood (2000), widely adopted in mechanical engineering, electrical engineering, mechatronic system development, and product service system design literature (see Eisenbart, 2014, for detail). Pahl et al. (2007) do not focus on the flow characteristics; therefore, multiple operation modes that depend on different parameter values of the flows cannot be captured in a single model (Schultz et al.,

2010). For the same reason, it is difficult to ensure that the functional model complies with the conservation laws of physics and therefore that it is “mathematically correct” (also questioned by Vermaas, 2008, 2009).

Bond graphs model physical systems through a network (McBride, 2005; Silva et al., 2014), whose vertices represent physical elements (analogous to design elements and referred to by the number of energy ports that they possess) and whose edges (known as power bonds or simply bonds) represent power flow. Each bond implicitly carries an effort variable (e.g., force or voltage) and a corresponding flow variable (e.g., velocity or current). There are also two special types of vertices called junctions that represent series and parallel connection. Bond graphs support the analysis of complex systems across disciplines (Summers et al., 2001). Triengo and Bos (1985) note that it is difficult to model mechanical systems, while Umeda et al. (1990) pointed out that the model focuses on structure and behavior of a system, but not on its functions. Recent work has focused on addressing this shortcoming, as evidenced by Lucero et al. (2017) and Mokhtarian et al. (2017), who both tabulate mappings of the five bond graph elements to the functional taxonomy of Stone and Wood (2000).

2.3. Function-structure oriented approaches

Function–structure oriented approaches attempt to address the “what” and the “how” concurrently. The integration definition for function modeling (IDEF0; Buede, 2009; Ally & Ning, 2015) also represents controls and mechanisms associated with a function box, in addition to the inputs and the outputs (of material, energy, and information). The top-level function is decomposed into subfunctions by preserving the bounding arrows (input, output, control, and mechanism) of the top-level function box (Buede, 2009). While the IDEF0 can represent the operation of a system in terms of a chain of functions in detail, function modeling using the IDEF0 can become complicated even in the analysis of a simple system (King & Sivaloganathan, 1998; Eisenbart, 2014).

Functions can also be represented in relation to state transitions. A state machine characterizes the behavior of a system in terms of a state at a particular time, and it is visually represented by a state-transition diagram (Harel, 1987). Harel (1987; see Jaelson et al., 2014) extended the state transition diagrams to statecharts by introducing the notions of clustering, concurrency, refinement, and zooming. However, as noted by Buede (2009), semantics and syntax of statecharts are limited in the modeling of complex systems. Statecharts are useful in the analysis of a known configuration, and they are used in the popular graphical modeling languages, Universal Modeling Language (UML) and SysML, that are commonly employed for the model-based representation of multidisciplinary systems (Friedenthal et al., 2008; Albers & Zingel, 2013). UML/SysML state machines introduce a strong formalism, but do not cover the way of defining, articulating, and decomposing functions (Eisenbart et al., 2015).

The SAPPPhIRE model (Srinivasan & Chakrabarti, 2009; Srinivasan et al., 2012) uses state transitions as one of its entities (i.e., state change, action, part, phenomenon, input, organ, effect) in the analysis of an engineering system. Srinivasan and Chakrabarti (2009) introduced a model of causality used to explain the causality of engineered systems through these entities.

Matthiesen and Ruckpaul (2012) represented modeling of systems with logical states in the contact and channel approach (C&C²-A). They introduce the “sequence model” in which a sequence consists of at least two states and it determines the operational mode of a system. A state can be part of different sequences and can address the fulfillment of several functions. A function includes at least two working surface pairs, connecting channel and support structure, and at least two connectors to embed the model into its environment. Design decomposition is carried out through describing the basic elements working surface pairs, connecting channel and support structure, and connectors at different levels of the system (Matthiesen & Ruckpaul, 2012; Albers & Wintergerst, 2014; Freund et al., 2015). The development of a functional model for a reasonable large system through C&C²-A may become difficult due to the use of multiple elements in the representation of a function.

2.4. FBS-oriented approaches

FBS approaches introduce the concept of “behavior” to establish a link between “what” and “how” in relation to function modeling. The FBS model of Umeda (Umeda et al., 1996; Umeda & Tomiyama, 2015) considers that a function cannot be represented independent of behavior so they represent a function as an association of “to do something” and “a set of behaviors” that exhibit this function. A function is expressed in verb–object–modifier format, while behavior is defined as sequential state transitions over time where a state consists of entities, attributes of entities, and relations between entities. Physical phenomena regulate the changes of entity attributes by relating them to physical laws (Alvarez Cabrera et al., 2009). The representation of a functional model using the FBS model can easily and quickly become complicated and bulky, as the model consists of too many nodes and edges (van Beek & Tomiyama, 2009).

The SBF model of Goel (Goel et al., 2009; Goel, 2013) uses the concept of behavior to map structural elements of a known device to functions. The behavior concept represents the change in structure in terms of a sequence of state transitions. Each state transition is annotated by the reasons for the transition, for example, physical laws. The SBF model provides a structured methodology in the decomposition of given input and output states into sequential state transitions. This also limits the model to the development of nonbranching state transitions.

Object–process methodology (OPM) of Dori (2002, 2016) describes function, structure, and behavior of a system in a single model through its basic elements, objects, and

processes. The OPM can represent a system at any level through its complexity management mechanisms (Dori, 2002).

3. FUNCTION MODELING BASIS FOR THE SSFD

In relation to the framework for function modeling approaches discussed in Section 2, the SSFD can be defined as a flow-oriented approach, which aims to capture the transformations that must take place on each flow such that the overall function of the system is achieved. The SSFD function modeling representation schema, generically illustrated in Figure 2, follows the general principles embodied in state diagrams (such as state transition diagrams or reliability state diagrams; see, for example, Rausand & Hoyland, 2003) and statecharts (Harel, 1987), which are the basis of the UML/SysML state machine diagrams (Weilkiens, 2006). The SSFD graphical convention is that boxes are used to represent the states of the objects, for example, the input and the output states, as illustrated in Figure 2, while the arrow denotes the function required to achieve the transfer from the initial (input) state to the desired final (output) state.

The following sections aim to establish a rigorous basis for the definition of states and a consistent graphical representation to ensure a concise capture of the required information; and the articulation of function coherent with the SSFD representation.

3.1. Conventions for state definitions and representations

Several function modeling tools, such as statecharts (Harel, 1987), FBS (Umeda et al., 1996), SBF (Goel et al., 2009), and OPM (Dori, 2002), use states for the function modeling and representation of a system. Figure 3 summarizes some of the key definitions and representations of states based on the cited authors' examples.

With the exception of FBS (Umeda et al., 1996), which uses multiple elements in the representation of a state, representations of states are based on a rectangular box. The state representation of Harel (1987) is prevalent in the software-based graphical representations of multidisciplinary systems, for example, the UML/SysML state machine diagram (Weilkiens, 2006), and it was therefore adopted for the SSFD representation.

In relation to the definition of state, it is apparent from Figure 3 that the reviewed approaches, while using different terms, are generally consistent with each other. The term *state* is associated with a specific situation of a given element (Harel, 1987), an entity (Umeda et al., 1996), substance/com-

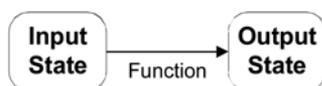


Fig. 2. System state flow diagram function modeling representation as state transition.

ponent (Goel et al., 2009), and an object (Dori, 2002). Umeda et al. (1996) and Dori (2002) point out that the situation of an entity and an object can be specified in terms of a set of “attributes” that have values, while Goel et al. (2009) use the term *parameter* instead. This requires a critical analysis to establish a consistent approach for both the definition of states in relation to objects and the description of states via attributes.

3.1.1. State definition via objects

The terms *entity*, *component*, and *object* are used interchangeably in the literature. Umeda et al. (1996) describe an entity as a component, while Alvarez Cabrera et al. (2009) suggest that an entity corresponds to an object. Goel et al. (2009) used the term *substance* to describe the structure of a system, with substances being contained in the components. Weilkiens (2006) suggested the term *element* in the UML/SysML state machine diagrams.

The term *object* is used in the definition and the articulation of a function by numerous function modeling approaches. For example, Stone and Wood (2000) articulate a subfunction in verb–object form and describe the object as “the recipient of a function’s operation.” Sickafus (1997) defines an object as a “tangible item.” For Dori (2002), an object is a thing that has the potential of physical or mental existence. The former pertains to tangible aspect of an object, which can be shown, touched, or experienced. Mental existence of an object is referred to as being intangible, which can be realized, depending on its form, through means such as being recorded on some tangible medium such as paper, the human brain, and so on (Dori, 2002).

Stone and Wood (2000) describe a flow as the object of a subfunction. The flow taxonomy of the functional basis provides a wide range of objects (i.e., flows) in the form of material, signal, and energy; however, some terms on this taxonomy are related to characteristics of an object rather than the object per se, for example, velocity. The object description of Sickafus (1997) pertains to tangible aspects of objects, although information and light are considered special cases, at the designer’s discretion.

The SSFD adopts the definition of an object based on Dori (2002), which is also the basis for the international standard for automation systems and integration (International Organization for Standardization, 2015).

3.1.2. Attributes in the definition of a state

The statecharts of Harel (1987) and the implicit state representation of Dori (2002) in Figure 3 directly show the value of a given element and an object, while Umeda et al. (1996) and the explicit state representation of Dori (2002) in the same figure describe the value of an entity and an object through attributes. Goel et al. (2009) use the term *parameter* with the same meaning.

Coherent with the attribute description of Tomiyama and Yoshikawa (1986), Umeda et al. (1990) suggested that an attribute should be measurable to be observed by scientific

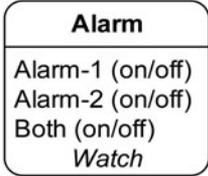
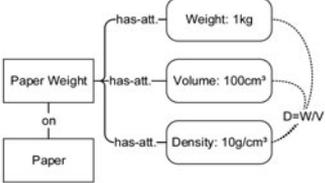
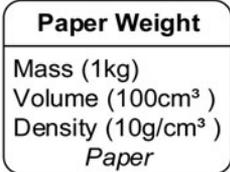
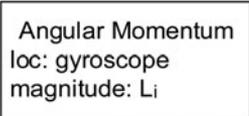
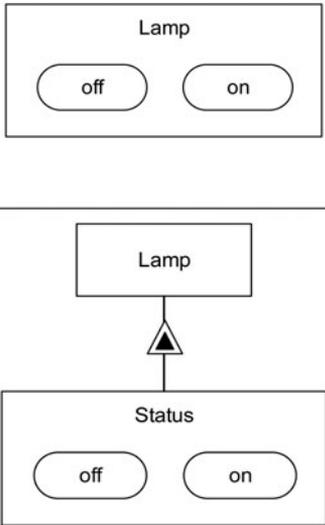
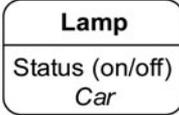
Approach	State Definition	State Representation	Example of states	SSFD State Model
Statecharts (Harel, 1987)	“the condition of a given element which can be specified in terms of a set of value combinations”	A state is represented by a rounded rectangle at any level. A sub-state is expressed using the notion of encapsulation, e.g. the state “alarm 1 beeps”.	Alarm function of a Watch 	
FBS model (Umeda et al., 1996)	The state of an entity is described in terms of a set of attributes that have values and relations among relevant entities	Illustrative example on the right represents an entity, an attribute and a relation as a rectangle, rounded rectangle and line, respectively.	Paper weight 	
SBF model (Goel et al., 2009)	the values of the parameters of substances and/or components	The state of the substance “angular momentum” is shown in a rectangle box along with its parameters (e.g. magnitude) and the values of these parameters (e.g. Li)	Gyroscope 	
OPM (Dori, 2002)	the situation of an object at a particular time	<p>Implicit representation: The object “lamp” is shown inside a rectangle box. , while the state of the lamp is represented by a rounded rectangle box within the object, e.g. “off”</p> <p>Explicit representation: The object “lamp” in the rectangle box exhibit the attribute “status”. The value “on” and “off” of “status” within rounded rectangle boxes are the lamp’s states.</p>	Lamp 	

Fig. 3. Summary of definitions and representations of states.

means. All approaches in Figure 3 use the term *value* in the articulation of a measurable object attribute. According to Dori (2002), a value is the concrete specification of an attribute. The SSFD uses the concepts of Umeda et al. (1990) and Dori (2002) in the description of “measurable” attributes of both physical and conceptual existence of an object. Sickafus (1997) suggested that a tangible object should possess at least the attributes of mass and volume, while for Dori (2002), a physical object should have mass and occupy coordinates in space and time. Some controversy exists in literature in relation to objects that may not possess mass and volume (e.g., light).

However, it could be hypothesized that space and time can be referred to as “global attributes” of an object, meaning that each object should possess these attributes. The design approach of Pahl et al. (2007) and the functional basis of Stone and Wood (2000) reflect the global attribute “time” by representing sub-functions of a system in respect of causality, while the FBS of Umeda et al. (1996) and the SBF of Goel et al. (2009) embed the time dimension into the function definition through behavior, which is described as sequential state transitions.

The SSFD incorporates “time” and “space” as “global attributes” of an object. The global attribute “time” is embedded into the state definition by representing transfer between states with respect to causality, that is, an input state of a state transition is the output state of another state transition, and vice versa. The global attribute “space” is referred to as the “location” of an object in which it is acted on and it is specified along with the object’s attributes (i.e., local attributes) in italic. Figure 4 summarizes the convention for the representation of a state in the SSFD.

The SSFD state box in Figure 4 consists of two parts; the upper part contains the name of the object in bold, while the lower part contains the object attributes including local measurable attribute(s) and global attribute “location.”

For illustration, the last column in Figure 3 shows the SSFD state representations for the examples considered based on the reviewed approaches. This shows that the SSFD state model provides a compact state definition and representation in a structured way by differentiating between local attributes (e.g., size) and global attributes (i.e., location and time) of an object. For example, in contrast to the OPM, the SSFD model represents the state of the lamp in a single representation.

3.2. Function modeling in the SSFD

There is no uniform function definition in the literature (Umeda et al., 1996); however, the meaning of function can be generalized based on the design method used (Vermaas,

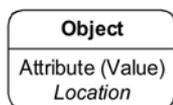


Fig. 4. State representation in the system state flow diagram.

2013). Coherent with the proposed SSFD state model, the OAF framework of Sickafus (1997) was adapted for the SSFD in the definition of a function. Thus, in the SSFD, a function is conceptually defined in terms of the triad of an input state of an object, an output state of an object, and an object (the design solution) that has to be provided in order to achieve the transfer between states, by addressing relevant attribute(s) of the input state. The associated graphical representation is illustrated in Figure 5.

The use of an arrow in the representation of a function is common practice in state-based approaches, for example, the statecharts (Harel, 1987). Coherent with the principles of Harel (1987), an open arrow with the function text below is used to denote a function in the SSFD, as shown in Figure 5. The head of the arrow is located on the output state. The function text indicates that the attribute(s) of the output object is modified through combining relevant attribute(s) of the design solution with the attribute(s) of the input object. The design solution is represented in a gray box in Figure 5.

A function is commonly articulated in verb–noun format in respect of the flows of material, energy, and information. For the SSFD, this articulation is related to the transfer between states. Considering the SSFD state model, the articulation of a function in verb–noun format is coherent the OAF framework of Sickafus (1997), that is, based on the rule that the verb corresponds to the operation on the object attribute(s), whereas the noun relates to the object or the object attribute.

Figure 6 summarizes the comparison of the SSFD function model with the approaches previously considered in Figure 3. Comparative representations in Figure 6 shows that the SSFD supports a compact function representation. The SSFD function model divorces the consideration of function from the consideration of the design solution by including the design solution conceptually in function modeling. Therefore, only the flow of state transitions is represented, initially without an explicit reference to a design solution. For example, in the second row of Figure 6, the SSFD models the FBS “paper weight” example (a) in terms of the input state and the output state of the paper instead of the paper weight; the function “stabilize paper” is articulated based on the specification of the paper location. If a specific design solution is chosen (e.g., a paper weight), coherent with the SSFD function model, the paper weight can fulfill this function as long as its relevant attribute (i.e., mass) combines with the relevant

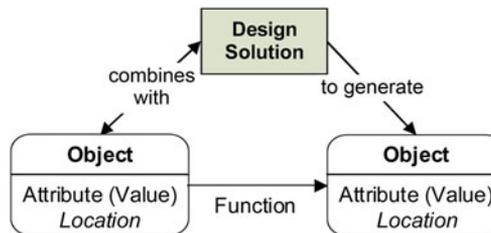


Fig. 5. Comparison of the system state flow diagram function model with the approaches in Figure 1.

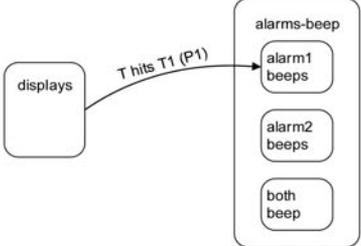
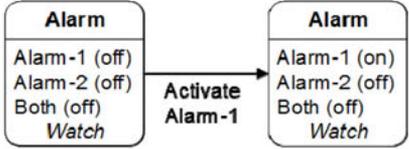
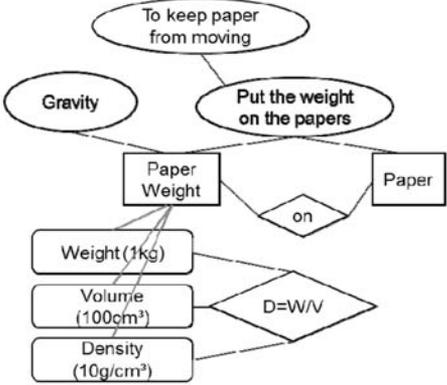
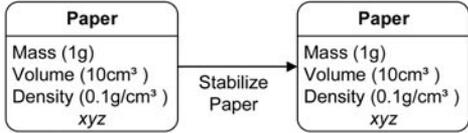
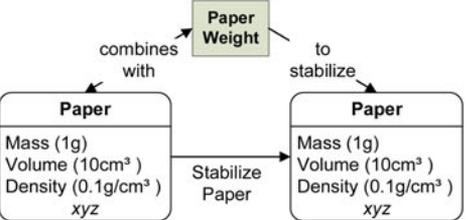
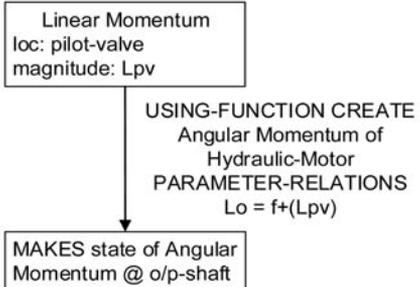
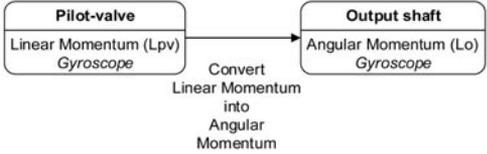
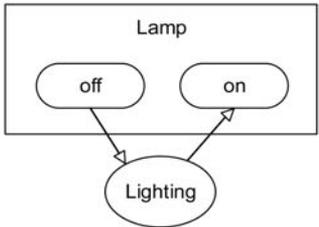
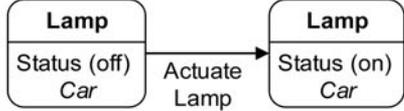
Approach	Example of function and representation	SSFD state transition representation
<p>Statecharts (Harel, 1987)</p>	<p>Alarm function of a Watch Function (event): T hits T1</p> 	
<p>FBS model (Umeda et al., 1996)</p>	<p>Paper weight Function: to keep paper from moving</p> 	<p>(a)</p>  <p>(b)</p> 
<p>SBF model (Goel et al., 2009)</p>	<p>A sub-function of a gyroscope Function: Create angular momentum of hydraulic motor</p> 	
<p>OPM (Dori, 2002)</p>	<p>Lamp Function: Lighting and Lamp function to make an object or area easy to see</p> 	

Fig. 6. System state flow diagram function model.

attribute of the paper, leading to a specific output state. The SSFD function definition triad is used to illustrate this, see the SSFD Model (b) in the second row of Figure 6. Different design solutions can be used to achieve the same objective, and can be illustrated in a similar way. The main advantages of the SSFD over the FBS are the representation of the state in a single box and the clear separation between function and design solution.

An SSFD functional model of a system can be developed by defining all functions of the system on the basis of the triad defined in Figure 5. Coherent with Suh's (2005) complexity definition, it can be suggested that the more flows of material, energy, and information a system addresses, the more complicated the functional model of the system becomes. This shows the need for a set of steps for the establishment of functional model of a system using the SSFD function model.

4. DEVELOPMENT OF SSFD HEURISTICS

Practical implementation of function modeling approaches requires extensive subjective judgement of the practitioner, which often conflicts with the desired "structured" development of functional models of systems. The effect of this subjectivity on function modeling can be mitigated by providing explicit, prescriptive guidelines for the practitioner, which have been termed *heuristics* by Maier and Rechtin (2000), described as "trusted, time-tested guidelines for serious problem solving." Otto and Wood (2001) have introduced "module heuristics" to support the practical implementation of the "functional basis" approach (Stone & Wood, 2000). This section describes the development of heuristics for the SSFD function modeling methodology, following a similar logic and approach with that of Otto and Wood (2001), by defining a main flow heuristic, a connecting flow heuristic, and a branching flow heuristic. In addition, to reflect the requirements of function modeling for contemporary multidisciplinary systems, which have multiple distinct modes of operation connected by logical conditions, a conditional fork node heuristic is also introduced.

The deployment of the SSFD heuristics is exemplified with a case study on the glue gun device, designated as a function modeling benchmarking case study by the engineering design community (Eckert & Summers, 2014).

4.1. SSFD main flow heuristic

Pahl et al. (2007) have introduced the term *main flow* to represent the main functions of a system, that is, those that directly address the fulfillment of its overall function. In a similar vein, Otto and Wood (2001) have used the term *dominant flow*. Complex multidisciplinary systems often present difficulties with the identification of the main flow, as they have multiple interconnected flows. The identification of a main flow to start the analysis is therefore the first challenge that needs to be addressed by the analyst. The decomposition of the main flow into a functional chain is the next step. The following sections outline the SSFD methodology for completing these tasks.

4.1.1. Identification of the main flow

Chandrasekaran and Josephson (2000) have discussed that there are two views in function modeling of a system: device centric and environment centric. The effect of the device on the environment in which it operates (i.e., environment-centric view) is a result of the way of working of the device (i.e., device-centric view) related to the intended effect on the environment. The environment is the user or an object related to the intended effect on the user. Coherent with Deng's (2002) description of user-focused purpose of design, and based on the methodology of Chandrasekaran and Josephson (2000), the main flow heuristic in the SSFD is introduced to describe the purpose of a system by determining the flow related to the intended effect of the system on the user.

Figure 7 illustrates this concept for the glue gun example. In relation to the use of the glue gun, the main intent of the user is to join two or more elements to create craftwork or to close and seal a parcel or box. Figure 7 shows an environment-centric SSFD representation, capturing the input state "open box" and the output state "closed box" of the object of interest, associated with the main intent of the user. The SSFD "triad" representation includes the main function, for example, to "join box" (e.g., by joining the two surfaces of the box top flaps), and the "glue" as the design solution for the use case illustrated.

The fundamental working principle of the gun is that a quantity of melted glue is deposited on the desired location of the box flap; the two flap surfaces are then joined and held together until the glue is fully solid and thus the box is sealed. Thus, from a device-centric perspective, the main

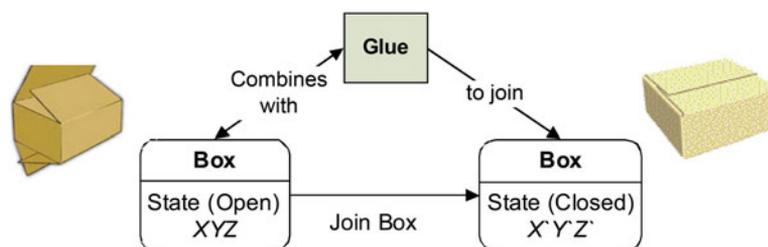


Fig. 7. Environment-centric system state flow diagram for the glue gun in relation to the main function "Join box."

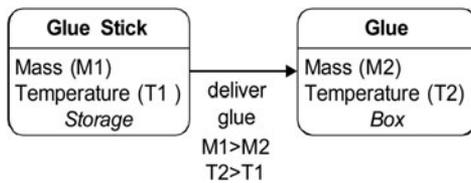


Fig. 8. Device-centric high-level system state flow diagram for the glue gun.

flow is a material flow, that is, the glue. The input state is the glue stick at room temperature (temperature T_1) located in storage, while the intended output state is a quantity M_2 of melted glue (temperature T_2) deposited on the surface of the box, in the desired location. The function of the glue gun is therefore to deliver a quantity of glue in melted state on the desired location on the box, such that the required environment-centric function “join box” can be achieved. The corresponding SSFD for the glue gun device is shown in Figure 8. Attribute values are denoted by symbols for the sake of simplicity of the analysis (e.g., M_1 for glue stick mass).

4.1.2. State-based decomposition of the main flow

Coherent with the conversion–transmission module heuristic of Otto and Wood (2001), operation types on a state transition can be categorized into conversion and transmission. The former addresses the change in the attributes of an object that the function is applied to, while the latter is about changing the location of an object through the applied function. The glue stick, which is the input material, undergoes multiple conversion and transmission operations to achieve the specified output state of “glue” described in Figure 8. These operations can be addressed by describing intermediate state transitions between the input and output states in Figure 8. The identification of these state transitions requires the description of the way of achievement (see Kitamura & Mizoguchi, 2003) of the state transition in Figure 8, that is, the physical phenomena that regulate the changes of the object attributes. As noted by Umeda et al. (1996), we can reason out intermediate states between the input state and the output state based on description of the physical phenomena. Like the input and the output states, an intermediate state is thought of as an object described by its measurable attributes. Once intermediate states are identified, the flow through these states is mapped between the input state and the output state with respect to causality. Finally, functions are iteratively articulated to achieve the sequence of state transitions.

Solution-neutral decomposition reasoning usually starts with identifying the key conversions that describe the “way of achievement.” For the glue flow, the main conversion required is to increase the temperature of the glue stick to the specified temperature (e.g., 120–180°C, depending on the material used) such that a physical state change (“melted” glue) is achieved. In order to deliver the device centric function discussed in Figure 8, a specified quantity (M_2) of melted glue needs to be extracted and deposited at the desired location; this can be achieved by subsequent conversions like applying a force to pressurize the melted glue, followed by extrusion of melted glue to atmospheric pressure. In order to apply these conversion operations, the glue stick must be “imported” into the device, which is a transmission operation.

Figure 9 shows the sequence of transmission and conversions on the main flow in an SSFD format, that is, representing each intermediate state described in terms of the values of the relevant attributes, linked to the previous state with an arrow that denotes the function (transmission or conversion) that needs to be achieved. The attributes involved are consistently indicated for all intermediate states, to support the subsequent specification of the functional requirements.

The device-centric SSFD for the main flow of the glue shown in Figure 9 can be aggregated into the environment-centric functional model showed as an SSFD state transition in Figure 7. Thus, the output state of “glue” in Figure 9 denotes the design solution, which combines with the input state of the box (open) in Figure 7, to achieve the output state of box (closed), fulfilling the user-required function to “join box.” Figure 7 showed this link by using two types of arrows (i.e., one and double headed) for the purpose of clarifying the triad, but in practice this link can be shown by one arrow, as illustrated in Figure 10. The arrow from the output state is pointed to the function text related to the conversion, showing extruded glue as the design solution in this function triad. This arrow is represented with a dashed line to distinguish it from the function arrow, as shown in Figure 10.

4.2. SSFD connecting flow heuristic

The main flow in Figure 10 shows the conversion and the transmission operations between the input state and the output state of the gun. In order to achieve the conversion operations, flows of additional resources need to be “connected” to the main flow to complete the function “triads,” showing the way of achievement of the function. The “connecting flow” heuristic relates to the systematic identification and analysis of the flows associated with the conceptual, solution-inde-

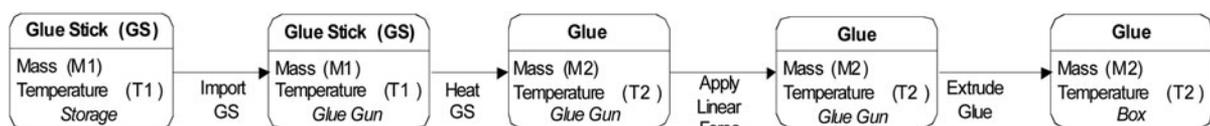


Fig. 9. The “main flow” through the glue gun device.

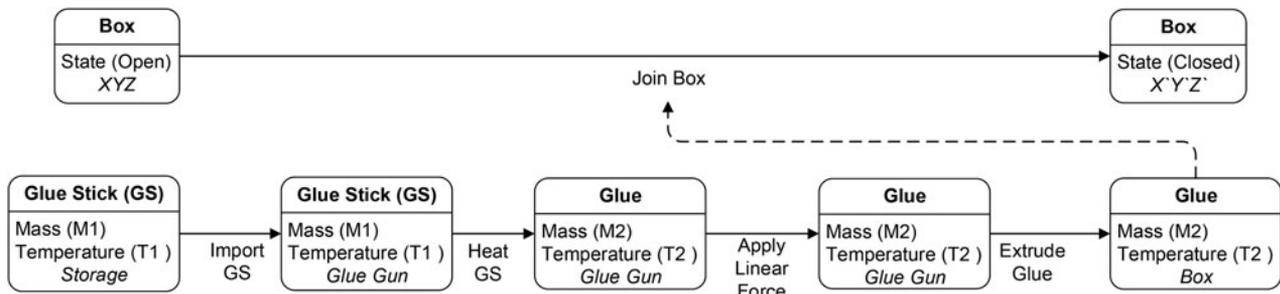


Fig. 10. System state flow diagram for the glue gun: the flow of glue.

pendent “design solution” that delivers the functions identified on the main flow.

The functions “heat GS” and “apply linear force” are conversion functions on the main flow of the glue gun in Figure 10, which require additional flows. The way of achievement in the process of melting glue is associated with an increase of the glue temperature. A common glue gun uses conduction to transmit the heat between the heat source (which can be an electric heat converter, e.g., a Nichrome wire) and the glue stick, via a conducting rigid body (the glue gun barrel), which also plays a role in the “apply linear force” conversion. Thus the flow of energy, from the electricity source to the heat transmitted through conduction to the glue stick, must be mapped using the SSFD and “connected” to the main flow of the glue gun device. Figure 11 illustrates the SSFD for this flow, based on the assumption that mains electrical energy is used as a source. Intermediate state transitions on this flow can be determined using the same SSFD principles followed in the decomposition of the main flow of the glue gun. The functions required, as shown in Figure 11, are to import electrical energy to the glue gun, and then to convert electric energy into heat, which is transmitted (through conduction) to the glue stick.

In terms of the function “apply linear force,” for the common glue gun, the user is the source of energy. The glue gun device converts energy from the user to linear force applied to the cold end of the glue stick; the rigid glue stick acts as a piston, transmitting force and converting it into pressure/fluid energy for the melted glue. Figure 12 illustrates this energy flow.

Figure 13 shows an updated SSFD that connects the electrical and user energy flows with the main flow through the device. This shows the design solutions for achieving the desired behaviors as functions on the main flow as follows:

- the heat energy converted from mains electrical energy combines with the glue stick to deliver an increase in the stick temperature to the point where part of the stick becomes fluid (melted); and
- the linear force (generated from the user energy) applied to the glue by the glue stick pressurizes the melted glue.

4.3. SSFD branching flow heuristic

Some technical systems require the flows to be split and allocated to different functions. For example, the hydraulic brake system for a vehicle distributes the fluid energy from a master cylinder to more than one wheel to simultaneously generate a retardation force for the vehicle. It is also the case that the real behavior of electromechanical systems cannot be achieved without loss; the flow of the resources that are diverted away from the intended function need to be mapped and accounted for, as these can cause issues elsewhere in the system or at the interface with other systems. For example, not all the energy of the heat source is transmitted to the glue stick; a part of it is transmitted to the environment via convection. The quantity of heat that is transmitted to the environment increases as the temperature of the glue increases, and thus the temperature gradient between the heat source and the glue decreases. The SSFD branching flow heuristic provides a mechanism to represent flows that branch out of the considered flow, and to ensure that the SSFD functional model is “mathematically correct,” that is, observes the conservation laws of physics. The correctness of the model in relation to branching flows can be checked via the balance of the attributes.

Figure 14 shows an updated SSFD, which illustrates the flow of heat branching out of the energy flow, with heat

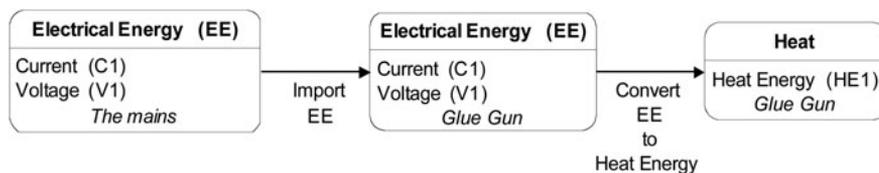


Fig. 11. System state flow diagram for the flow of electrical energy through the glue gun.

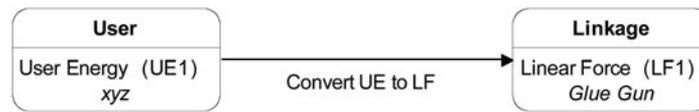


Fig. 12. System state flow diagram for the conversion of user energy in the glue gun.

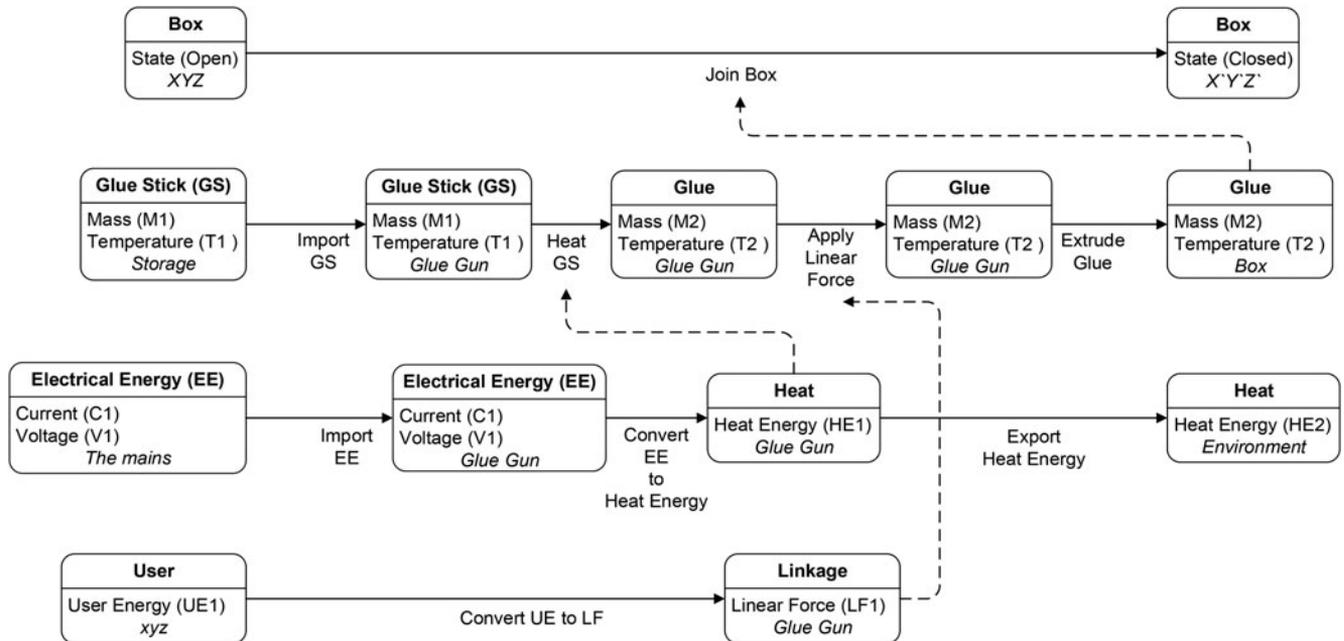


Fig. 13. System state flow diagram for the glue gun: the flow of electrical and user energy.

energy being exported to the environment from the glue gun through convection. The effect of this energy export to the environment needs to be investigated from an engineering point of view, to ensure that no detrimental effects are allowed and reduce system efficiency. For example, the effect of this heat energy on the environmental air temperature is very likely negligible; however, if we consider the impact on other environmental factors, such as the user, or another object like the box itself, the effect could be severe, leading to injury of the user and damage (or even fire) to the box.

The representation in Figure 14 also shows the “system boundary”; this is denoted by the rectangular box that contains the states and functions that must be achieved by the device, confirming the demarcation between the environment-centric view and the device-centric view.

4.4. SSFD conditional fork node heuristic

Complex multidisciplinary systems often have multiple modes of operation, corresponding to different use cases. While each operation of the system can be independently described by a functional chain, aggregating all functional chains into a global functional model for the system that captures the complex functionality of the system is a challenge, often not appropriately addressed by conventional functional

models. The aggregation of functional chains often requires the identification of logical or parametric conditions. A functional model that is capable of representing such a complex system needs to capture the logic of the system. In order to address this challenge, a special SSFD heuristic is introduced to indicate the flow operation (either connecting type or branching type) that is subject to a logical or parametric condition. Coherent with the statecharts (Harel, 1987) convention and the SysML state machine “fork node” (Friedenthal et al., 2008), this SSFD heuristic is called the “conditional fork node.” Figure 15 illustrates the representation of the fork node for two distinct cases:

1. there can be multiple possible output states distinguished by the attribute values involved, which depend, through the stated condition, on the attribute value of the input state and the attributes or parameters of the function; and
2. there can be multiple input states distinguished by attribute values, which depending on the operating conditions and the function attribute or parameter value, determine the output state attribute value.

In order to illustrate this for the glue gun case study, a glue gun that offers low-temperature and high-temperature opera-

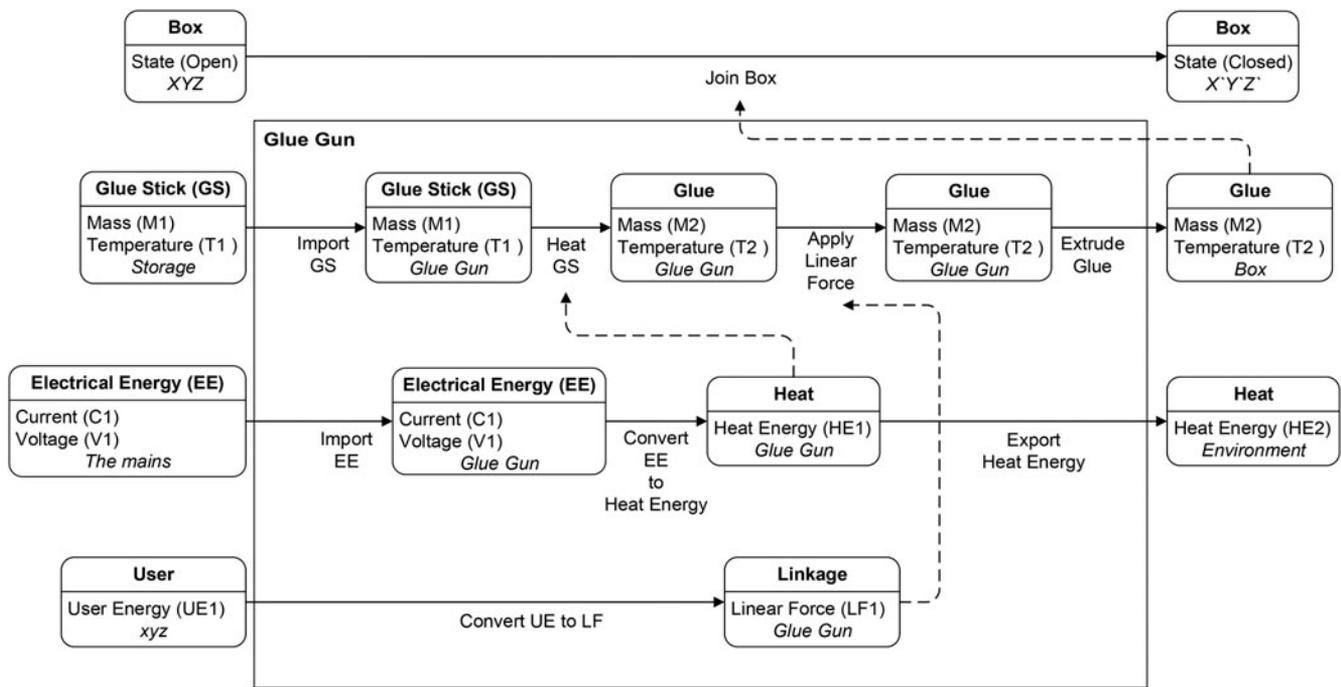


Fig. 14. Glue gun system state flow diagram: branching flow of thermal radiation.

tion modes to serve multiple use cases/user requirements in terms of the strength of the bond is considered. Such devices provide the option of low temperature (around 120°C, suitable for working with more delicate materials) or high temperature (operating at approximately 190–210°C, providing a stronger bond), which can be selected by the user via a switch positioned on the body of the device. Different glue stick materials are employed for the two modes of operation. For both operation modes, the device operates based on the same fundamental working principle, and with the same functional chain, although the specified attribute values (including the temperature of the glue at the output state) will be different depending on the conditions. Figure 16 illustrates the SSFD of the whole system, using several fork nodes to implement the operating logic of the system.

1. Fork node F1 indicates that the input state can be either low- or high-temperature glue stick, differentiated by the material composition and attributes (e.g., mass, de-

noted by M1 and M3, respectively). The geometric dimensions are similar, and the way in which the sticks are imported to the glue gun is the same, as shown by the fork F1. The value of the mass attribute corresponding to the output state of the function “import GS” is denoted by M and it can be either M1 or M3, depending on the input state.

2. The function “heat GS” is controlled by the attribute of the heat energy output state of the function “convert EE to heat energy”; in turn this is controlled by the current attribute value of the output state of the function “regulate current,” which is conditional on the heat gun control setting, as shown in fork node F2. The corresponding triad for this function consists of input state current (C), which combines with the control signal (if an electronic switch is used; otherwise, it could be a simple circuit breaker that modulates the resistance of the heater element by shorting out a section of it) to “regulate current” (assuming a constant supply voltage

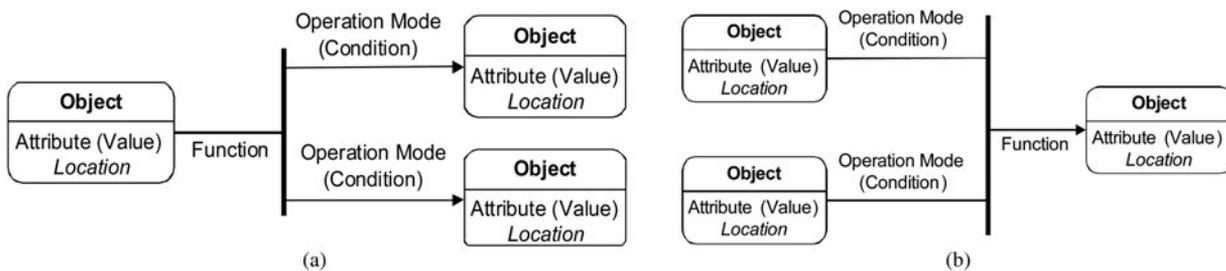


Fig. 15. System state flow diagram fork node.

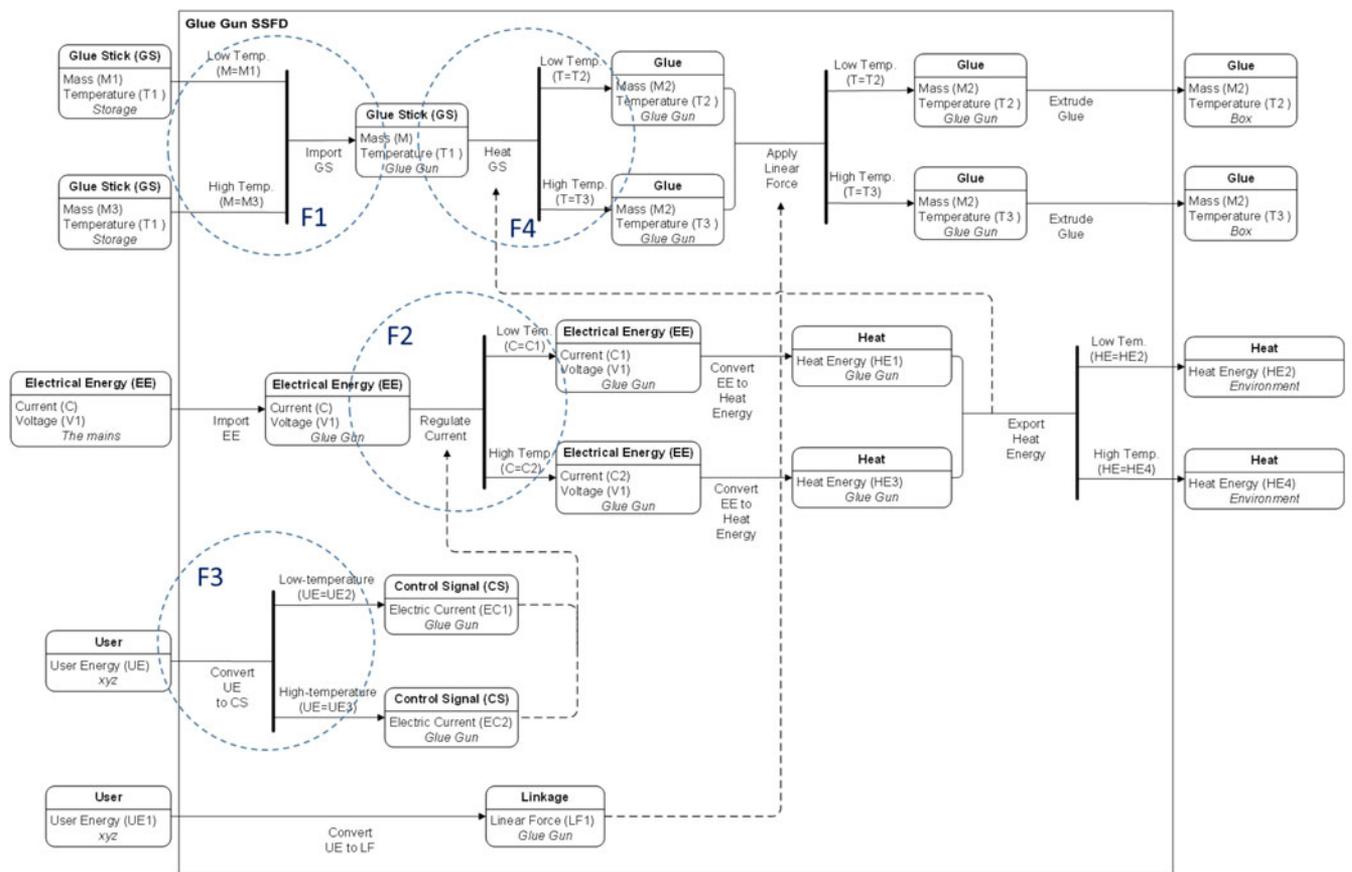


Fig. 16. Glue gun system state flow diagram for low- and high-temperature operation modes.

the current will be determined by the resistance) and deliver the output state attribute value, which is either C1 or C2.

3. Fork node F3 illustrates the SSFD for the temperature control selector (switch), which is activated by the user.
4. Fork node F4 illustrates that the output of the “heat GS” function depends on

- the attribute value of the input state, which can be either low-temperature or high-temperature glue stick; and
- the value of the attribute heat energy, which controls the function.

5. Assuming that the trigger force is the same in all cases, the successful output from the system will be the specified amount of melted glue at low or high temperature.
6. Heat energy exported to the environment from the glue gun will be different depending on the operation mode of the gun.

5. INDUSTRIAL CASE STUDY: EVP MODELING USING SSFD

To provide further empirical validation of the methodology, a case study of the application of the SSFD function modeling

to a real-world engineering problem was considered, for the analysis of an EVP for a full electric light commercial vehicle, based on an industrial case study analysis carried out by the authors, outlined in Campean et al. (2011).

An EVP has several modes of operation, as illustrated in the SysML use case diagram in Figure 17. For the purpose of this analysis, the use cases considered include “Charge EV” (UC1)

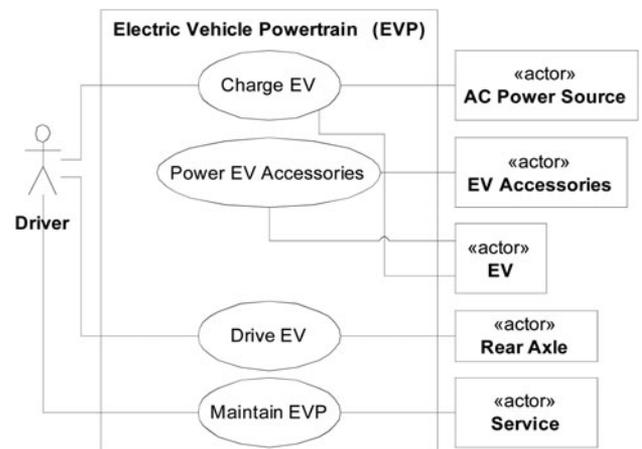


Fig. 17. Use case diagram for the electric vehicle powertrain.

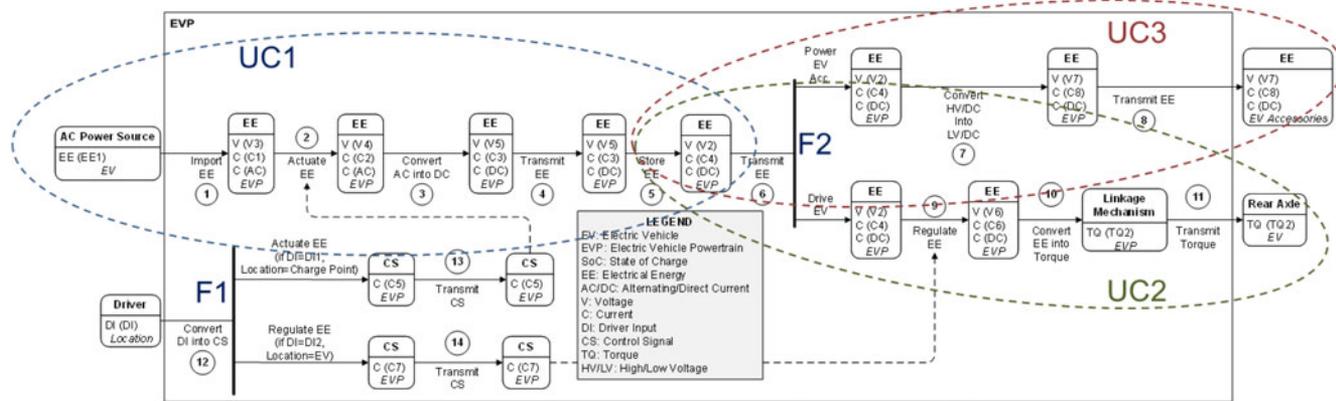


Fig. 18. System state flow diagram function model for the electric vehicle powertrain.

charge and store electrical energy; “Drive EV” (UC2) associated with the requirement to “provide controlled torque at the rear axle”; and “Power EV Accessories” (UC3) to provide power for low-voltage vehicle consumers (Campean et al., 2011).

An SSFD function model can be constructed for each individual use case by applying the SSFD main flow heuristic. The three SSFDs can be joined up in a single EVP function model, by using the connecting flow, branching flow, and fork node heuristics, ensuring that the appropriate logic conditions are fulfilled; for example, the “Drive EV” mode cannot be selected while the vehicle is being charged (i.e., in “Charge EV” mode). The aggregated SSFD function model for the EVP from a device-centric view is shown in Figure 18, with the use cases identified on the diagram.

The control logic of the system is based on the user input, which can be either DI1, to connect the vehicle to a charge point, or DI2, to drive the vehicle (fork node F1); in both cases, the user intent is converted to a control signal. The flows for UC2 and UC3 are connected to the output state of the UC1 flow (stored DC electric energy) via fork node F2; in this case, the fork node F2 can be treated as a simple branching flow as the stored EE is supplied to both UC2 and UC3 flows.

The SSFD functional analysis for the EVP shown in Figure 18 is entirely generic, that is, solution neutral. The required transfer functions, annotated on the SSFD diagram (1–14), can be written explicitly, enabling the transfer functions for the whole function chain to be derived. By doing so, the model can be exported to a physical modeling environment

like Simulink. The function architecture model of the EVP can be extracted from the SSFD and represented as a function tree, as shown in Figure 19.

Figure 20 shows the physical architecture of a specific EVP configuration, represented as a structural boundary diagram, corresponding to the solution implemented in the industrial case study. The allocation of functions to design elements can be annotated on the SSFD, by adjusting appropriately the “location” state attribute, as shown in the updated SSFD function model in Figure 21. It is important to note that developing the structural design diagram on the basis of the SSFD preserves the integrity of the flows. This is useful because the arrows in the boundary diagram denote the flows, which could be checked (e.g., through diagnostics) using the attributes identified in the SSFD for the respective state. This also facilitates transition to a simulation model, with clear diagnostics and model checking.

6. CRITICAL EVALUATION OF SSFD USING THE FUNCTION REPRESENTATION BENCHMARKING FRAMEWORK

This paper has introduced a rigorous and comprehensive function modeling framework based on the SSFD, and has used two case studies (one for a benchmark problem, the glue gun, and one for an industrial application) to illustrate application of the framework. This section aims to provide a critical reflection on the SSFD as a function modeling methodology based on the function representation bench-

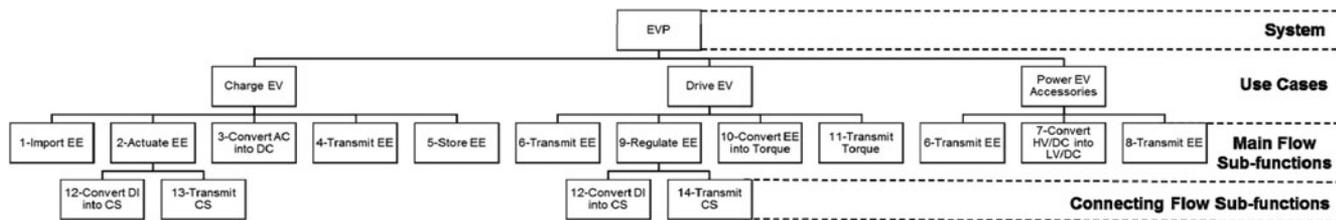


Fig. 19. Electric vehicle powertrain function architecture representation as function tree.

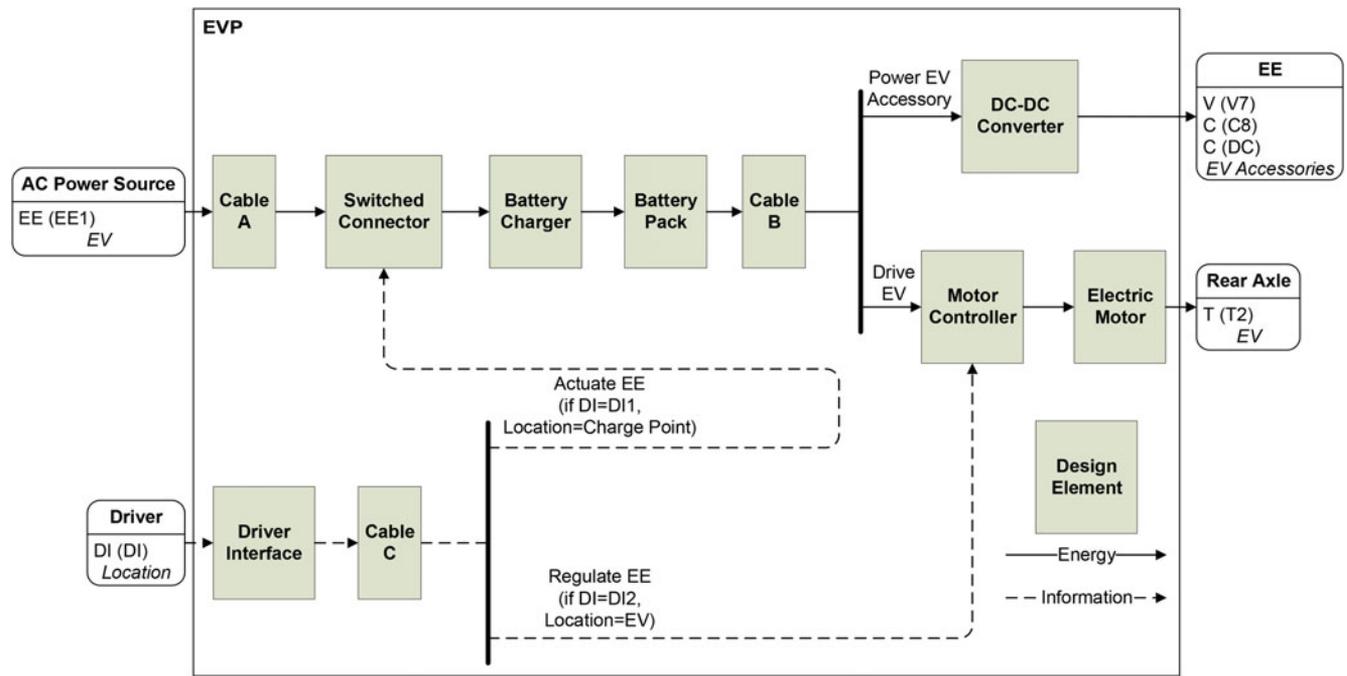


Fig. 20. Structural design boundary diagram for the electric vehicle powertrain, showing design components and flows through the system.

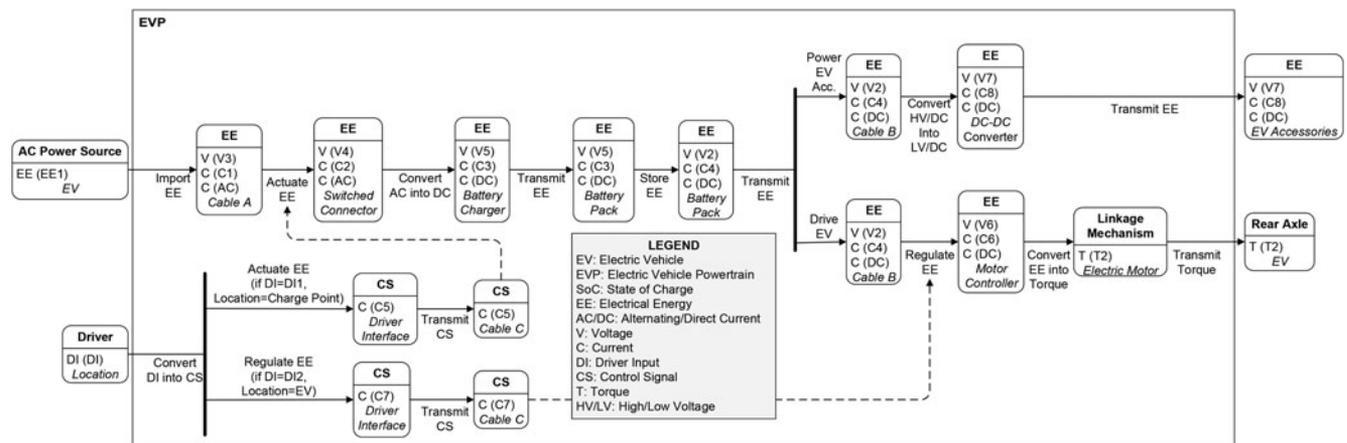


Fig. 21. System state flow diagram model for the electric vehicle powertrain system, updated with design solutions.

marking framework established by Summers et al. (2013), using evidence from the two case studies considered. The suggested benchmarking dimensions of representation characteristics, modeling characteristics, cognitive dimension characteristics, and reasoning characteristics will be discussed in turn in relation to the SSFD function modeling framework in the following sections.

6.1. Representation characteristics

6.1.1. Scope

As presented in this paper, SSFD is essentially a flow-oriented approach, because it aims to provide a description

of the functional architecture of a system in a solution-independent manner. This sets it aside from the function–structure and function–behavior–structure oriented approaches discussed in Section 2, which provide function models that combine function and structure. However, reference to the design solution, once known, can be added to the model as a global state attribute, as illustrated in Figure 21 for the EVP case study. Reference to the structure could also be added as an annotation in the SSFD, as shown in Figure 1, and illustrated in the industrial examples of Campean, Henshall, and Rutter (2013) and Henshall et al. (2015). In summary, the SSFD can be used for function representation for reverse engineering as well as for hierarchical top-down decomposition of complex systems.

6.1.2. Consistency

Both case studies presented in this paper illustrate that the SSFD provides a consistent framework for function modeling. Consistency is ensured by the state-based function representation, with the states being described in terms of (physically) measurable attributes. Furthermore, the SSFD heuristics ensure that other conditions (time/sequence dependency and parameter dependency) can be appropriately captured in the system functional model, for example, as illustrated by the glue gun analysis in Figure 16. The representation of conditional flows based on the fork node provides a better representation compared to other flow-based function modeling techniques, including the functional basis of Stone and Wood (2000).

6.1.3. Behavior

The SSFD does not explicitly describe behavior; however, in the SSFD, behavior is implicitly addressed through its consistent approach to compliance with the laws of physics and the explicit capture of conditions that represent the behavior of the system. To substantiate this point, consider Figures 13 and 14 from the glue gun analysis; in Figure 13, the intended behavior was represented through the connecting flow of heat applied to the glue stick to increase the temperature of the glue. However, in Figure 14, based on the assumption of a common glue gun heating solution (using nichrome resistor), the actual behavior is shown in terms of the branching flow of heat diverted via convection to the environment. The description of dynamics of this behavior would be in the form of a heat transfer function, captured in parametric form in the state description in Figure 14. Figures 16 for the glue gun and Figure 18 for the EVP illustrate the behavior in relation to conditional flows associated with specific use cases; behavior can be described by monitoring the transformation of the attributes and the description of transfer functions, as shown with more detail by Yildirim (2015) for the EVP case study.

6.1.4. Scalability

The SSFD can be used at different scales of the system; however, the methodology was principally designed for top-down analysis. The glue gun case study illustrated this by starting with an environmental-centric analysis (the highest level of abstraction), followed by a device-centric analysis, with increasing level of detail shown by the progression of model development from Figure 13 to 14 and to 15. The analysis could be further developed in parallel with the design architecture development, for example, by representing the way the user energy is converted to linear force, which is applied to the glue stick. The connecting flow, branching flow, and conditional fork node heuristics enable the development of the SSFD with an increasing level of resolution.

However, experience with practical application shows that the SSFD, like most other function modeling methodologies, is more suited for the higher levels of system. Applying SSFD

at a level where functions are essentially pure physics becomes more difficult (but possible) in comparison with physics-oriented modeling tools like the bond graphs (Silva et al., 2014; Mokhtarian et al., 2017) and the contact and channel functional modeling framework (Albers et al., 2014), which provide a stronger functional modeling approach for this level of abstraction.

For practical design management purpose in industry, it is preferable to keep functional models reasonably simple at each level of system analysis, to facilitate communication between teams. The common practice observed in industry is to use design structure matrix analysis (Eppinger & Browning, 2012) or interface management tools (Uddin et al., 2016) based on the structure of the system (e.g., the boundary diagram illustrated in Fig. 20) to capture the interface exchanges incurred by the actual behavior of the design components.

6.1.5. Indexing

From an engineering point of view, the SSFD provides an important opportunity to diagnose function models by measuring the internal states of the flow, for example, via the relevant attributes of the objects whose states are described in the SSFD. If functional modules are developed and modeled through the SSFD, these will have a built-in diagnostics structure that could be easily accessed by the analyst.

6.1.6. Flexibility and translationability

Being a generic function modeling approach, the SSFD can be applied to diverse problems both within and across the engineering domains. The glue gun and the EVP analysis (at the limit of resolution considered in the paper) mostly illustrate the SSFD deployment in the electromechanical domain. However, experience with industrial use of the methods has shown that the methodology can be applied to control and software systems (Campean, Henshall, & Rutter, 2013; Naidu et al., 2017) and even to process function modeling (Dobryden et al., 2017). The SSFD could be linked to other engineering models such as bond graphs or physical/ analytical simulation models (e.g., based on Simulink).

6.2. Modeling characteristics

The SSFD heuristics play an essential role in the way the analyst interacts with the model, which is a strength of the framework. The main flow heuristic is arguably the most important step. The approach suggested in the SSFD is to start with the environment-centric perspective, focusing on the goal or utility that is required from the device. The main flow through the device should always be aligned with the main goal or utility. In the case of the glue gun analysis, this has immediately pointed to “glue” as being the main flow through the device. However, with large complex systems, the main flows are often not immediately obvious.

The input and output states for the main flow are defined next. If the analysis is driven by the environment-centric anal-

ysis, then the output state would have to be defined first; in the case of the glue gun, the mass, location, pattern, and temperature of the glue were defined as attributes of the output state. Identifying the input state requires design analysis and judgment; the benchmark glue gun considered the use of glue sticks, although pellets could also have been used.

Decomposition of the main flow within an SSFD is based on the identification of intermediate states, which can be “measured.” This can be pursued either from the input state to the output state or from the output state backward to the input state; in both cases, the consistency of the attributes describing the states must be ensured. For innovative design, when no reference solution is available or considered, the strategy of Pahl et al. (2007) of starting with identifying the key transformative operation is the most robust strategy. In terms of SSFD, this would be equivalent to identifying a mid-state of the flow, associated with the main transformation, and then identifying state transitions required before and after the main transformative step. For material flows, the visualization technique commonly used in process mapping was suggested by Campean, Henshall, and Rutter (2013) as an efficient strategy: “if I am a molecule of glue at the tip of the glue stick, what transformations do I go through to end up deposited in melted state in the required location on the box?”

This approach to guiding the analyst to correctly identify the main flow is much stronger than other functional modeling approaches, including the functional basis of Stone and Wood (2000). The engineers tend to focus on mapping the energy flows first, as illustrated by the example of glue gun function analysis based on mapping the flows of energy material and information in Eckert and Summers (2014). While such a model might still be correct from an engineering point of view, it is not focused on the goal of the system. From a practical/industry application point of view, the focus on the goals is paramount. In order to address this, from an engineering point of view, we need to understand the use case goals, and to ensure that we have engineering measures consistent with the stakeholder measures of the delivered output, and that we focus the way of achievement on the delivery of the output.

The development of the SSFD model through the connecting, branching, and conditional fork node heuristics provides the analyst with the facility to develop models with increased levels of detail. In effect, this is equivalent to developing the model across successive levels of abstraction. The way this is managed in practice depends on the approach adopted for product development management. If the design development is carried out by one design unit, then the ability to develop the model with increased levels of detail and ultimately to have a detailed model of the whole system, which can be verified and simulated, would be of interest. However, if the design is carried out by a distributed organisation (e.g., involving suppliers), which is typically the case with many OEMs, then the analysis will be carried out at successive system levels based on the system architecture. In this case, a modular approach to the development of the SSFD can be taken.

6.3. Cognitive dimensions characteristics

Table 1 summarizes the evaluation of the SSFD function modeling framework against the key cognitive dimensions suggested by Summers et al. (2013). A noteworthy point in relation to the analysis presented in Table 1 is that the sensitivity of the SSFD function modeling to the expertise of the analyst is an area that still requires research and development. While the SSFD provides a crisp focus on identification of measurable attributes to describe the states (input and output states to begin with), many real-world systems have such complexity that would require a large number of attributes and metrics to fully characterize the state of the object. This can be illustrated even in the case of the glue gun: the analysis in Figure 10 suggests that the relevant attributes to describe the output state of glue are mass, temperature, and location (XYZ). This would be appropriate if the use case goal is to have a blob of glue in the desired location, but if a pattern is required, then the mass flow rate would be a more appropriate attribute. This problem could be addressed with further research in two ways: develop a taxonomy for description of states; and develop a library of generic models for generic flows, which could be imported and customised for specific engineering problems.

6.4. Reasoning characteristics

The original reason for developing the SSFD function modeling approach was to enhance the function failure reasoning (Campean et al., 2011; Campean, Henshall, & Rutter, 2013). The automotive industry uses a model-based approach for function failure analysis (AIAG, 2008) early in the design process, which requires a robust functional model to carry out the function failure reasoning. The evidence from industry practice (Campean et al., 2011) is that functions are often brainstormed, and coherent function models are often not developed. It has also been observed that there is poor linkage between system levels in terms of function traceability; engineering teams would start to develop a functional model for a module based on the understanding of how the module works rather than what function it is required to deliver within the bigger system. This leads to poor integration and traceability within the system design development. The SSFD has delivered a fundamental underpinning for the development of a framework for robust engineering design based on failure mode avoidance, which has been comprehensively deployed within the automotive industry (Campean, Henshall, & Rutter, 2013; Henshall et al., 2014, 2015; Dobryden et al., 2017; Naidu et al., 2017).

The subsequent development of the SSFD methodology, including the framework described in this paper, was driven by the need to develop consistent function models starting with the customer and logical layers of the system, and traceable across the levels of the systems, through robust functional requirements and function failure avoidance metrics. The

Table 1. Evaluation of SSFD in relation to key cognitive dimensions

Dimension	Question of Functional Modeling	Evaluation of SSFD Function Modeling
Abstraction gradient	What are the minimum and maximum levels of abstraction? Can partial models be created?	SSFD is a top-down function modeling framework; it can be applied at the highest level of abstraction (user/environment centric) and sequentially to lower levels.
Closeness of mapping	What modeling conventions need to be learned? How intuitive is the resulting model?	SSFD is an engineering focused framework, with strong focus on description of states via measurable attributes. The SSFD heuristics are not complex. The interpretation of the model focuses on engineering (i.e., based on following the flow of states) and verifying correctness through object attributes and transfer functions.
Error-proneness	Does the design of the notation induce “careless mistakes”?	Although the SSFD modeling notation is in principle accurate, it still relies on the expertise of the analyst to identify/define the appropriate attributes and metrics.
Hidden dependencies	Is every dependency overtly indicated in both directions? Is the indication perceptual or only symbolic?	Dependencies within the SSFD function models are captured in terms of the sequence through the directionality of the flow (from input to output) and in terms of condition through the conditional fork node heuristic. Dependencies can be articulated in terms of analytical transfer functions.
Premature commitment	Does the model require decisions before the information needed is available?	The SSFD is based on information from the higher level of the abstraction/hierarchy of the system. A generic functional model for the module might still be developed, provided the main flow can be correctly identified.
Secondary notation	Can the models be annotated or linked to other product representations?	The SSFD can be annotated with information about the design solution/architecture (i.e., components that deliver each function) or functional requirement articulation.
Viscosity	How much effort is required to perform a single change? How easy is it to adapt the model from a model of a similar product?	The SSFD is based on state based representations, so it has similar viscosity to state charts and state machines (both are structure based rather than flow based). However, there is no immediate tool that can be used to deploy SSFD, which could make model adjustments time consuming. SSFD models can be easily adopted (e.g., for functional modules) and/or adapted.
Visibility	How easy is it to see all aspects of the model? Can two models be compared?	The clear diagrammatic nature and the simplicity of capturing information in state definition boxes makes SSFD model visualization reasonably straightforward. Models can be compared using the diagrams directly or via the associated function trees of the analytical transfer functions.

SSFD provides a strong basis for exchange-based articulation of functional requirements and the associated requirements specification, underpinned by the state-based modeling and states description through attributes.

7. CONCLUSIONS

This paper has introduced a rigorous framework for function modeling and representation based on the SSFD, aimed at supporting the generic function modeling of complex multi-disciplinary systems.

A key contribution of the paper is that it has established a rigorous academic basis for the SSFD by defining the key elements of the function modeling framework, and the convention for the graphical representation of functional models, in the context of existing established functional modeling approaches.

The SSFD heuristics provide systematic guidance to the analyst in applying the framework to practical problems, facilitating the development of consistent SSFD function models. In particular, the main flow heuristic, which prompts the analyst to consider the highest level of abstraction and the environment-centric context of the system, facilitates the

identification of the device-centric function and main flow in relation to the main utility of the device. The consistent linkage between the environment-centric and the device-centric function reasoning is of significant importance to industry practice, as it focuses the attention in the function modeling on the value adding aspects from a user perspective.

The SSFD heuristics of connecting flow, branching flow, and conditional fork node, enable the development of scalable functional models that are consistent with the laws of physics, and with operating and behavior conditions of the system. This makes the SSFD applicable to complex multi-disciplinary systems that require function models that span across multiple use cases and modes of operation, and across multiple levels of abstraction of the system.

The SSFD provides a more compact representation compared to the FBS oriented approaches, while capturing compatible information in the function model. At the same time, the SSFD is more adept at representing generic yet consistent function models of the system at higher levels of system abstraction, compared to function modeling frameworks oriented on phenomenological physics, such as contact and channel and bond graphs.

The application of the SSFD framework to the glue gun benchmark problem, and the comprehensive evaluation of

the SSFD based on the benchmarking framework of Summers et al. (2013), provide a reference function model that can be used by other researchers to evaluate their own functional models.

The significance of this paper is that, having established a rigorous reference framework for the SSFD function representation, and a set of heuristics to guide the practitioner in its deployment, it paves the way for the consistent deployment of the SSFD in industrial practice for the analysis of complex multidisciplinary systems.

ACKNOWLEDGMENTS

The authors acknowledge the technical input and support from many engineers at Ford Motor Company and Jaguar Land Rover, who provided both motivation and valuable insight from the testing of the SSFD functional modeling framework on a large range of industrial problems. Special thanks are due to Ed Henshall, Royal Academy of Engineering Visiting Professor in integrated system design, and Brian Rutter of Ford Motor Company, who played an important role in the early development and validation of the SSFD. The views presented in this paper are those of the authors and do not represent those of the aforementioned companies.

REFERENCES

- AIAG. (2008). *Potential Failure Modes and Effects Analysis*, 4th ed. Southfield, MI: Author.
- Albers, A., Oerding, J., & Alink, T. (2011). Abstract objectives can become more tangible with the contact and channel model (C&CM). *Models and Methods for Variation Management in Global Product Development: Proc. 20th CIRP Design Conf.*, pp. 203–213, Ecole Centrale de Nantes, April 19–21, 2011. Berlin: Springer.
- Albers, A., & Wintergerst, E. (2014). The contact and channel approach (C&C²-A): relating a system's physical structure to its functionality. In *An Anthology of Theories and Models of Design* (Chakrabarti, A., & Blessing, L.T.M., Eds.), pp. 151–171. London: Springer-Verlag.
- Albers, A., & Zingel, C. (2013). Extending SysML for engineering designers by integration of the contact & channel-approach (C&C²-A) for function-based modelling of technical systems. *Procedia Computer Science* 16, 353–362.
- Ally, A.F., & Ning, Z. (2015). What the current system development trends tell us about systems development methodologies: toward explaining SSDAM, agile and IDEF0 methodologies. *Journal of Applied Computer Science & Mathematics* 19(9), 9–15.
- Alvarez Cabrera, A.A., Erden, M.S., & Tomiyama, T. (2009). On the potential of function-behaviour-state (FBS) methodology for the integration of modelling tools. *Competitive Design: Proc. 19th CIRP Design Conf.*, pp. 412–419, Cranfield, March 30–31.
- Buede, D.M. (2009). *The Engineering Design of Systems: Models and Methods*, 2nd ed. Hoboken, NJ: Wiley.
- Campean, I.F., Henshall, E., Brunson, D., Day, A., McLellan, R., & Hartley, J. (2011). A structured approach for function analysis of complex automotive systems. *SAE International Journal of Materials and Manufacturing* 4(1), 1255–1267.
- Campean, I.F., Henshall, E., & Rutter, B. (2013). Systems engineering excellence through design: an integrated approach based on failure mode avoidance. *SAE International Journal of Materials and Manufacturing* 6(3), 389–401.
- Campean, I.F., Henshall, E., Yildirim, U., Uddin, A., & Williams, H. (2013). A structured approach for function based decomposition of complex multi-disciplinary automotive systems. *Smart Product Engineering—Proc. 23th CIRP Design Conf.* (Abramovici, M., & Stark, R., Eds.), pp. 191–200. Berlin: Springer-Verlag.
- Chandrasekaran, B., & Josephson, J.R. (2000). Function in device representation. *Engineering With Computers* 16, 162–177.
- Chiang, W.-C., Pennathur, A., & Mital, A. (2001). Designing and manufacturing consumer products for functionality: a literature review of current function definitions and design support tools. *Integrated Manufacturing Systems* 12(6), 430–448.
- Chourey, V., & Sharma, M. (2016). Functional flow diagram (FFD): semantics for evolving software. *Proc. Int. Conf. Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2193–2199, Jaipur, India, September 21–24.
- D'Amelio, V., & Tomiyama, T. (2007). Predicting the unpredictable problems in mechatronics design. *Proc. Int. Conf. Engineering Design*, Paris, August 28–31.
- Deng, Y.M. (2002). Function and behaviour representation in conceptual mechanical design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 16, 343–362.
- Dobryden, A., Rutter, B., Hartl, D., & Bramson, E. (2017). Failure mode avoidance approach for hybrid electric vehicle systems. *SAE International Journal of Engines* 10(2). doi:10.4271/2017-01-0298
- Dori, D. (2002). *Object-Process Methodology: A Holistic Systems Paradigm*. Berlin: Springer.
- Dori, D. (2016). *Model-Based Systems Engineering With OPM and SysML*. Berlin: Springer.
- Eckert, C., & Summers, J.D. (2014). *Workshop 7: Benchmarking Functional Models. Proc. ASME IDETC/CIE*, New York, August 17–20.
- Eisenbart, B. (2014). *Supporting interdisciplinary system development through integrated function modelling*. PhD thesis. University of Luxembourg.
- Eisenbart, B., Gericke, K., Lucienne, T.M., Blessing, L.T.M., & Timothy, C.M. (2016). A DSM-based framework for integrated function modelling: concept, application and evaluation. *Research in Engineering Design* 28(1), 25–51.
- Eisenbart, B., Mandel, C., Gericke, K., & Blessing, L. (2015). Integrated function modelling: comparing the IFM framework with SysML. *Proc. 20th Int. Conf. Engineering Design (ICED15)*. Milan, Italy, July 27–30.
- Eppinger, S.D., & Browning, T.R. (2012). *Design Structure Matrix Methods and Applications*. Cambridge, MA: MIT Press.
- Erden, M.S., Komoto, H., van Beek, T.J., D'Amelio, V., Echavarría, E., & Tomiyama, T. (2008). A review of function modeling: approaches and applications. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 22(2), 147–169.
- Freund, T., Klobberdanz, H., Lotz, J., & Würtenberger, J. (2015). An approach to analysing interface uncertainty using the contact and channel model. *Proc. 20th Int. Conf. Engineering Design (ICED15)*, Milan, Italy, July 27–30.
- Friedenthal, S., Moore, A., & Steiner, R. (2008). *A Practical Guide to SysML: The Systems Modeling Language*. Burlington, VT: Kaufmann.
- Gericke, K., & Eckert, C. (2015). The long road to improvement in modelling and managing engineering design processes. *Proc. Int. Conf. Engineering Design (ICED)*, Milan, Italy, July 27–30.
- Goel, A.K. (2013). A 30-year case study and 15 principles: implications of an artificial intelligence methodology for functional modeling. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 27(3), 203–215.
- Goel, A., Rugaber, S., & Vattam, S. (2009). Structure, behaviour, and function of complex systems: the structure, behaviour, and function modelling language. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 23(1), 23–35.
- Harel, D. (1987). Statecharts: a visual formalism for complex systems. *Science of Computer Programming* 8, 231–274.
- Hehenberger, P., Vogel-Heuser, B., Bradley, D., Eynard, B., Tomiyama, T., & Achiche, S. (2016). Design, modelling, simulation and integration of cyber physical systems: methods and applications. *Computers in Industry* 82, 273–289.
- Henshall, E., Campean, I., & Rutter, B. (2014). A systems approach to the development and use of FMEA in complex automotive applications. *SAE International Journal of Materials and Manufacturing* 7(2), 280–290.
- Henshall, E., Campean, I., & Rutter, B. (2017). A systems approach to the development of enhanced learning for engineering systems design analysis. *Procedia CIRP* 60, 530–535.
- Henshall, E., Rutter, B., & Souch, D. (2015). Extending the role of interface analysis within a systems engineering approach to the design of robust and reliable automotive product. *SAE International Journal of Materials and Manufacturing* 8(2), 322–335.
- Hoffmann, H.-P. (2011). *Systems Engineering Best Practices With the Rational Solution for Systems and Software Engineering*. Somers, NY: IBM Corporation.

- International Organization for Standardization. (2015). *ISO/PAS 19450: 2015(en): Automation Systems and Integration—Object-Process Methodology*. Geneva: Author.
- Jaelson, J.P., Mylopoulos, J., & Silva Souza, V.E. (2014). From requirements to statecharts via design refinement. *Proc. 29th Annual ACM Symp. Applied Computing*, pp. 995–1000, Gyeongju, Korea, March 24–28.
- Kaufman, J.J., & Woodhead, R. (2006). *Stimulating Innovation in Products and Services With Function Analysis and Mapping*. Hoboken, NJ: Wiley.
- King, A.M., & Sivaloganathan (1998). Development of a methodology for using function analysis in flexible design strategies. *Journal of Engineering Manufacture* 212(3), 215–230.
- Kitamura, Y., & Mizoguchi, R. (2003). Ontology-based description of functional design knowledge and its use in a functional way server. *Expert Systems With Applications* 24, 153–166.
- Komoto, H., & Tomiyama, T. (2011). A theory of decomposition in system architecting. *Proc. Int. Conf. Engineering Design*. Technical University of Denmark, August 15–18.
- Lindemann, U., Maurer, M., & Braun, T. (2009) *Structural Complexity Management: An Approach for the Field of Product Design*. Berlin: Springer-Verlag.
- Liu, C., Hildre, H.P., Zhang, H., & Rolvag, T. (2015). Conceptual design of multi-modal products. *Research in Engineering Design* 26, 219–234.
- Lu, S.C.-Y., & Suh, N.P. (2009). Complexity in design of technical systems. *CIRP Annals Manufacturing Technology* 58, 157–160. doi:10.1016/j.cirp.2009.03.067
- Lucero, B., Adams, M., & Turner, C. (2017). Introduction to quantitative engineering design methods via controls engineering. *Artificial Intelligence for Engineering, Design Analysis and Manufacturing* 31(4), 458–475.
- Maier, M.W., & Rechtin, E. (2000). *The Art of Systems Architecting*, 2nd ed. Boca Raton, FL: CRC Press.
- Matthiesen, S., & Ruckpaul, A. (2012). New insights on the Contact & Channel-Approach—modelling of systems with several logical states. *Proc. Int. Design Conf. (DESIGN)*, pp. 1019–1028, Dubrovnik, Croatia.
- McBride, R.T. (2005). *System analysis through bond graph modelling*. PhD Thesis. University of Arizona.
- Mital, A., Desai, A., Subramania, A., & Mital, A. (2008) *Product Development: A Structured Approach to Consumer Product Development, Design and Manufacture*. Oxford: Butterworth-Heinemann.
- Mokhtarian, H., Coatanea, E., & Paris, H. (2017). Function modeling combined with physics-based reasoning for assessing design options and supporting innovative ideation. *Artificial Intelligence for Engineering, Design Analysis and Manufacturing* 31(4), 476–500.
- Naidu, A., Brittle, P., Ma, X., & Rutter, B. (2017). *Integrated Systems Engineering Approach for Incremental 48 Volt Hybrid Technology Introduction*. SAE Technical Paper 2017-01-1603. doi:10.4271/2017-01-1603
- NASA. (2007). *Systems Engineering Handbook*, rev. Washington, DC: Author.
- Otto, K., & Wood, K. (2001). *Product Design: Techniques in Reverse Engineering and New Product Development*. Upper Saddle River, NJ: Prentice Hall.
- Pahl, G., Beitz, W., Feldhusen, J., & Grote, K.H. (2007). *Engineering Design: A Systematic Approach*, 3rd ed. London: Springer.
- Pedersen, K., Emblemvag, J., Bailey, R., Allen, J.K., & Mistree, F. (2000). Validating design methods and research: the validation square. *Proc. ASME Design Engineering Technical Conf.*, Paper No. ASME DETC 2000/DTM-14579, Baltimore, MD, September 10–14.
- Rausand, M., & Hoyland, A. (2003). *System Reliability Theory: Models, Statistical Methods, and Applications*, 2nd ed. New York: Wiley.
- Schultz, J., Sen, C., Caldwell, B., Mathieson, J., Summers, J.D., & Mocko, G.M. (2010). Limitations to function structures: a case study in morphing airfoil design. *ASME IDETC/CIE*, pp. 405–417, Montreal, August 15–18.
- Sickafus, E.N. (1997). *Unified Structured Inventive Thinking: How to Invent*. Grosse Ile, MI: Ntellect.
- Silva, L.I., Bouscayrol, A., De Angelo, C.H., & Lemaire-Semail, B. (2014). Coupling bond graph and energetic macroscopic representation for electric vehicle simulation. *Mechatronics* 24, 906–913.
- Srinivasan, V., & Chakrabarti, A. (2009). SAPPiRE: an approach to analysis and synthesis. *Proc. Int. Conf. Engineering Design (ICED)*, pp. 417–428, Stanford, CA.
- Srinivasan, V., Chakrabarti, A., & Lindemann, U. (2012). A framework for describing functions in design. *Design Methods: Proc. 12th Int. Design Conf.*, pp. 1111–1122, Dubrovnik, Croatia, May 21–24.
- Stone, R.B., & Wood, K.L. (2000). Development of a functional basis for design. *Journal of Mechanical Design* 122, 359–370.
- Suh, N.P. (2005). Complexity in engineering. *CIRP Annals Manufacturing Technology* 54(2), 46–63.
- Summers, J.D., Eckert, C., & Goel, A. (2013). Function in engineering: benchmarking representations and models. *Proc. Int. Conf. Engineering Design*, Sungkyunkwan University, Seoul, August 19–22.
- Summers, J.D., Hernandez-Vargas, N., Zhao, Z., Shah, J.J., & Lacroix, Z. (2001). Comparative study of representation structures for modelling function and behaviour of mechanical devices. *Proc. DETC*, Pittsburgh, PA, September 9–12.
- Tomiyama, T. (2016). Function allocation theory for creative design. *Procedia CIRP* 50, 210–215.
- Tomiyama, T., D'Amelio, V., Urbanic, J., & ElMaraghy, W. (2007). Complexity of multi-disciplinary design. *Annals of the CIRP* 56(1), 185–188.
- Tomiyama, T., Gu, P., Jin, Y., Lutters, D., Kind, Ch., & Kimura, F. (2009). Design methodologies: industrial and educational applications. *CIRP Annals Manufacturing Technology* 58, 543–565.
- Tomiyama, T., van Beek, T., Alvarez Cabrera, A., Komoto, H., & D'Amelio, V. (2013). Making function modeling practically usable. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 27(3), 301–309.
- Tomiyama, T., & Yoshikawa, H. (1986). Extended general design theory. *Proc. IFIP WG5.2 Working Conf.*, pp. 95–130. Amsterdam: North-Holland.
- Triengo, M., & Bos, A. (1985). Modeling the dynamics and kinematics of mechanical systems with multibond graphs. *Journal of Franklin Institute* 319, 37–50.
- Uddin, A., Campean, F., & Khan, M.K. (2016). Application of the interface analysis template for deriving system requirements. *Proc. Int. Design Conf. (DESIGN)*, pp. 543–552, Dubrovnik, Croatia, May 16–19.
- Ullman, D.G. (2010). *The Mechanical Design Process*, 4th ed. New York: McGraw-Hill.
- Ulrich, K.T., & Eppinger, S.D. (2003). *Product Design and Development*, 3rd ed. New York: McGraw-Hill/Irwin.
- Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y., & Tomiyama, T. (1996). Supporting conceptual design based on the function-behavior-state modeller. *Artificial Intelligence for Engineering, Design Analysis and Manufacturing* 10(4), 275–288.
- Umeda, Y., Takeda, H., Tomiyama, T., & Yoshikawa, H. (1990). Function, behaviour, and structure. In *Applications of Artificial Intelligence in Engineering* (Gero, J.S., Ed.), Vol. 5, pp. 177–193. Berlin: Springer-Verlag.
- Umeda, Y., & Tomiyama, T. (2015). Development of function modeling and its application to self-maintenance machine. In *Impact of Design Research on Industrial Practice* (Chkrabarti, A., & Lindemann, U., Eds.), pp. 311–319. Berlin: Springer-Verlag.
- van Beek, T.J., & Tomiyama, T. (2009). Integrating conventional system views with function-behaviour-state modelling. *Proc. CIRP Design Conf.*, Cranfield University, March 30–31.
- Vermaas, P.E. (2008). On engineering meanings and representations of technical functions. *Proc. ASME IDETC/CIE Conf.*, New York, August 3–6.
- Vermaas, P.E. (2009). The flexible meaning of function in engineering. *Proc. Int. Conf. Engineering Design*, pp. 113–124, Stanford, CA, August 24–27.
- Vermaas, P.E. (2013). The coexistence of engineering meanings of function: four responses and their methodological implications. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 27(4), 191–202.
- Weilkiens, T. (2006). *Systems Engineering With SysML/UML: Modelling, Analysis, Design*. Burlington, VT: Kaufmann.
- Yildirim, U. (2015). *Function modelling of complex multidisciplinary systems: development of a system state flow diagram methodology for function decomposition of complex multidisciplinary systems*. PhD Thesis. University of Bradford.

Unal Yildirim is a Postdoctoral Research Assistant at the University of Bradford, where he received his MS and PhD degrees. Dr. Yildirim's research interests include automotive systems engineering design and product development, in particular functional modeling of complex multidisciplinary systems and engineering change management.

Felician Campean is a Professor of automotive reliability engineering, Director of the Automotive Research Centre, and

Associate Dean for Research & Knowledge Transfer in the Faculty of Engineering and Informatics at the University of Bradford. His main research interests include automotive systems engineering design and product development, and model-based design and optimization, in particular multi-disciplinary and multiobjective optimization applied to large-scale/high-dimensionality problems. Dr. Campean has an extensive track record of collaborative research with the global automotive industry.

Huw Williams is an Independent Consultant and a Senior Research Associate of the University of Bradford. From 1986 till 2009, he worked in various positions in the Advanced Engineering Department of Jaguar Cars Ltd. Huw attained his PhD in physical applied mathematics at the University of East Anglia. Dr. Williams' current interests are mathematical modeling of engineering systems, failure mode avoidance, and teaching engineers how to apply mathematical and statistical methods.