Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2013), *27*, 291–299.

doi:10.1017/S0890060413000292

RESPONSE PAPER

Identifying requirements for physics-based reasoning on function structure graphs

CHIRADEEP SEN[1] AND JOSHUA D. SUMMERS[2]

[1]School of Mechanical, Industrial, and Manufacturing Engineering, Oregon State University, Corvallis, Oregon, USA
[2]Department of Mechanical Engineering, Clemson University, Clemson, South Carolina, USA

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

Abstract

Function-based design and modeling have been taught, studied, and practiced in various forms for several years with efforts centered on using function modeling to help designers understand problems or to facilitate idea generation. Only limited focus has been placed on potential use for qualitative and quantitative reasoning and analysis of the design concept. This potential for early stage analysis has not been fully explored partly because computational reasoning tools have not been developed for this express purpose. This paper presents a set of requirements and their justification to realize this design enabling tool. The requirements include coverage, consistency, validity against physics laws, domain neutrality, physics-based definitions, normative and descriptive modeling, and qualitative and quantitative modeling and reasoning. Each requirement is defined in concrete terms and illustrated with examples and logic. With the requirements for function-based reasoning and representation clearly identified, future research toward formalizing of function-based design will be more focused and objective validation of proposed representations against these requirements would be possible.

Keywords: Concept Analysis; Conceptual Design; Formal Representation; Function Modeling; Reasoning

1. INTRODUCTION

Research in the representation of mechanical functions has been conducted from different viewpoints historically (Goel, 2013). In the *artificial intelligence (AI) view*, models exist mainly to support device description, cause-and-effect explanation, and design synthesis. The second viewpoint, the *engineering design view*, uses a popular representation, the graph-based function structure, to support different design reasoning. These views have approached the definition and understanding of function as behavior, user interaction, capabilities, goals, and structure with different design activity objectives (Vermaas, 2013). These different views and approaches to functional reasoning are developed to ultimately support engineering design activities in industry and education (Eckert, 2013). Ultimately, the capabilities of the existing tools limit the views of how function *might* be explored in engineering design. We envision a new direction of supporting physics-based reasoning through functional modeling based on conservation principles that can only be realized with formalized representations. The goal of this paper is to present a detailed enumeration of requirements that must be satisfied to fully support the envisioned reasoning support system without being in conflict with the form-independent view of function modeling. The proposed requirements supports one possible role of function modeling and design space exploration through feasibility reasoning on proffered solution concepts. These requirements can be used for both assessment and comparative benchmarking of current modeling and reasoning approaches while simultaneously providing research motivation and guidance for the community at large.

Reprint requests to: Joshua D. Summers, Department of Mechanical Engineering, Clemson University, 250 Fluor Daniel Building, Clemson, SC 29634-0921, USA. E-mail: joshua.summers@ces.clemson.edu

2. REVIEW OF FUNCTION-BASED DESIGN

The ultimate goal of the reasoning that can result from satisfying these requirements is to more fully support engineering design by providing tools that are more well received in industry (Eckert, 2013). Function-based reasoning about mechanical systems has been a topic of early interest in AI and cognition research (Eastman, 1969; Freeman & Newell, 1971; Simon, 1998), with design reasoning focused primarily on supporting artifact synthesis as opposed to explaining the workings of an existing device (Simon, 1998). Consequently, design reasoning requires a representation that captures intent.

The AI models of function, several of which briefly discussed in Vermaas (2013), are inspired by the complex interaction between multiple entities and are primarily descriptive (Umeda et al., 1996, Chandrasekaran & Josephson, 2000; Gero & Kannengiesser, 2002; Goel & Bhatta, 2004, Chandrasekaran, 2005; van Eck et al., 2007). This view recognizes that "how a device works" (behavior; Gero & Kannengiesser, 2002) is less dependent on the observer's viewpoints than on "what a device is for" or "what does a device do for human needs" (function; Umeda et al., 1996; Gero & Kannengiesser, 2002; Goel & Bhatta, 2004). These are more holistic views of function and are subject to the difficulties of modeling intentionality of designers and device use (Kroes, 2010). By contrast, the engineering design view takes a simpler view of functions as transformations of material, energy, and signal.

In engineering design, function is often seen as transformative actions of input and output flows in a system (Rodenacker, 1971; Pahl et al., 2007; Fenves et al., 2008). Often, a graph-based representation (function structure) captures these transformations (Pahl et al., 2007; Ullman, 2010) through nodes as transformative actions (functions) and edges as objects of actions (flows: material, energy, or signal). Figure 1 illustrates the function structure model of a hair dryer. To help formalize this representation, controlled vocabularies of functions and flows are proposed, typically through empirical observations, where actions and flows within mechanical devices are identified and cataloged (Collins et al., 1976; Kirschman & Fadel, 1998; Hirtz et al., 2002; Bohm et al., 2005). Several function-based tools have been developed to support engineering design activities (McAdams & Wood, 2002; Vucovich et al., 2006; Kurtoglu et al., 2010). Although these tools are computational, they do not perform reasoning to draw inferences about physical behavior and validity against natural laws. At best, these tools are generative, not analytical (Linz, 2011). Although researchers and engineers have employed function structures both for describing existing solutions, such as through reverse engineering, and for forward engineering, such as through conceptual design, each view of the function is still a transformative point of view.

Overall, the two directions in function representation research differ in a few ways. First, the AI representations attempt to include the user's and designer's intent, although the graph-based model does not include that, though in some models in the Design Repository (http://repository.designengineeringlab.org/) the user's interaction (usage) with the device is captured through flows of human material or human energy (Hirtz et al., 2002). Second, although a function structure captures function as transformations of flows within the system, the AI models traditionally discard this view because transformation alone is inadequate to capture the entire essence of functions, specifically the user's intent and the artifact's effect on the environment (Umeda et al., 1996). Third, the AI models typically use a free, natural language, and several expanding ontologies of functions have been proposed (Sasajima et al., 1995; Kitamura et al., 2004; Cebrian-Tarrason et al., 2008). Fourth, the AI models have been extended to substantial degree of formalism, where modeling languages such as Causal Functional Representation Language and tools such as case-based reasoners have been implemented. In the present state of the art, formalization of the function and flow definitions in function structures is limited to informal definitions of function vocabularies that does not yet support automated reasoning.

Functions might be goal based, physical, or existential (Eckert, 2013). Here, we address the physical and existential views as captured in the engineering design focused function structures. Although the reasoning requirements developed are specifically developed for the transformative view of functions, these might be expanded to other views of function modeling (Goel, 2013).

3. THE NEED FOR COMPUTER MODELING AND REASONING ON FUNCTION STRUCTURES

The lack of computer support to early mechanical design, as compared to detailed design, is commonly recognized. Computer-aided design (CAD)/computer-aided engineering tools support modeling and analysis using form-based design information (Fenves et al., 2008). For example, reasoning on strength requirements of components and systems are enabled through formal representations of geometry, loads, and boundary conditions, and manufacturing process planning is supported through geometric reasoning on standard repre-

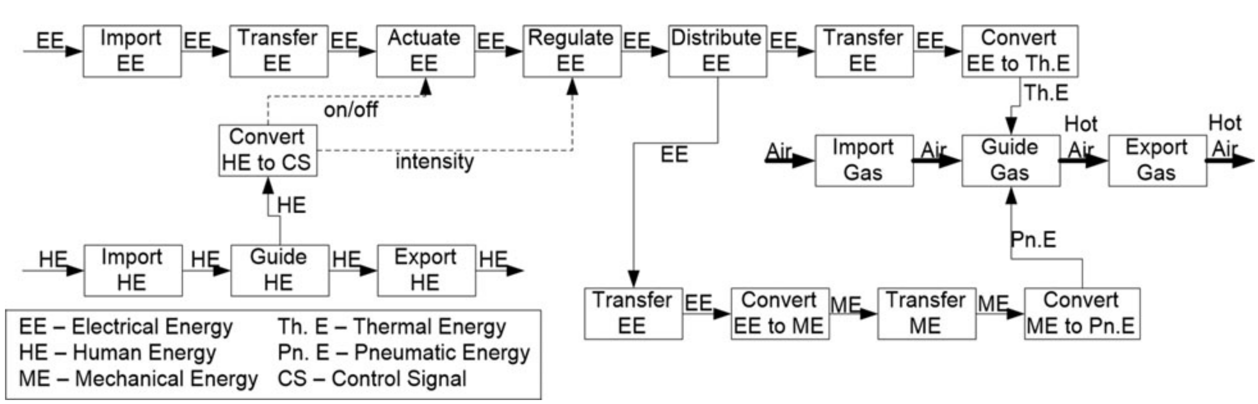

Fig. 1. The function structure of a hairdryer stored in the design repository.

https://doi.org/10.1017/S0890060413000292 Published online by Cambridge University Press

sentations. These types of reasoning are only possible with formal representations. In conceptual design, there is a need for similar tools that a designer could use to model a concept on a computer, interactively expand or explore design variants, and finally analyze the concept to get feedback suitable for concept-level decision making. Although function modeling might be useful for ideation, the potential for early stage analysis has not been adequately explored. In early design, designers often perform low-fidelity analyses, such as concept feasibility (is the concept realistic or does it violate physical laws?), theoretical completeness (have all flows and functions been considered or are some overlooked?), side effects (what emissions will the design produce and how much?), and energy efficiency (component-wise and overall efficiency, losses, power required, power produced). These analyses are not currently supported through automation for general-purpose mechanical design. Design insight could be gained early if these analyses were supported, thus enabling more informed decisions by front-loading the design process (Thomke & Fujimoto, 2000).

To support this type of reasoning, the representation underlying the models must be logically consistent and valid against the laws of physics. However, formal reasoning with the function structure representation has been realized with limited success. Recently, it was demonstrated that the lack of logical consistency and the lack of validity against the laws of physics is the root cause of much of this limitation (Sen et al., 2011*a*). To address this gap, a set of representation requirements is proposed.

3.1. Modeling gap

Function structure graphs are widely recommended in design texts as a means to model the intended functionality of new design concepts (Hubka & Eder, 1988; Otto & Wood, 2001; Pahl et al., 2007; Ullman, 2010). At present, limited work is reported toward constructing these graphs on the computer. Typically, concepts are modeled manually, on pencil and paper, whereas computer software, such as MS Visio^{TM}, might be used as diagramming tools. Although these digital models look like function structure graphs, these tools do not assist design in an intelligent manner; they merely replace the pencil with the mouse. They do not provide a controlled vocabulary of model entities (e.g., function modeling icons), do not prevent the designer from drawing absurd or unrealistic concepts, do not provide feedback if the model is in agreement with natural laws, and do not support any postmodeling reasoning on the concept itself. Their level of automation is analogous to the early two-dimensional drafting tools, where the designer was responsible for the correctness of the drawing: the tool would not raise an alarm if, say, a projection view was incorrect. Two initiatives of function structure modeling are significant. The first is the automatic generation of models using graph grammars (Sridharan & Campbell, 2005; Kurtoglu et al., 2010). This tool puts the computer in charge of synthesizing models, rather than the designer, and does thus not support designer-driven modeling and exploration outlined above. The second is function CAD (Nagel et al., 2009). The designer develops the concept using the functional basis vocabulary as icons or menu options. However, it cannot check for logical or physical inconsistency of the model; it permits inconsistent model topologies and violations of the laws of physics, and it does not support any postmodeling reasoning.

3.2. Reasoning gap

Model-based reasoning uses logic queries on a model to draw inferences or to reveal information about the modeled reality that is not explicitly captured in the model (Summers & Shah, 2004; Luger, 2005). For example, CAD kernels support computing the distance between vertices of a body, orthographic projection views, and mass properties, even if this data is not directly used to construct the model. In function-based design, consistent reasoning using function structure graphs has not been demonstrated adequately. Tools used in similarity detection (McAdams & Wood, 2002), concept generation (Vucovich et al., 2006), component layout (Kurtoglu et al., 2010), or failure modeling (Stone et al., 2005) use function-based information as the basis of reasoning but not from the graph form of the model. Rather, they use information in other representations, typically relational databases (Bohm et al., 2005). Graph grammar-based synthesis tools (Sridharan & Campbell, 2005; Kurtoglu et al., 2010) use the graphs as the basis of reasoning, but they are focused on generating concepts by pattern integration rather than inferencing. Other functional modeling tools exist (Vescovi et al., 1993; Umeda et al., 1996; Chakrabarti & Bligh, 2001; Deng, 2002; Goel & Bhatta, 2004; Albers et al., 2008; Wang et al., 2009; Srinivasan et al., 2012), but they do not use the notion of function as transformative action (Pahl et al., 2007) used in the function structure. This paper explores requirements for specifically the graph-based function structures, one of the most commonly used representations of functions.

Consistent and reliable reasoning requires the representation to be valid against the bodies of knowledge used in reasoning. For example, CAD tools can compute mass properties of a solid using geometric representations and algorithms to compute the volume or moment of inertia. The more generic the knowledge, the wider is the applicability of this reasoning. In mechanical design, external knowledge that applies universally regardless of function or form includes the balance and irreversibility laws. Irrespective of the specific functions or principles chosen in a concept, and even in the early stage of concept development, it is true that the design must satisfy the balance of mass and energy and is subject to irreversibility. The function reasoning tools mentioned above do not use such external knowledge as the basis of reasoning. By enabling physics-based reasoning of this type in function-based design, much insight could be gained early in the process, which would support more informed decisions.

4. REQUIREMENTS FOR THE REPRESENTATION

This section identifies requirements for a representation to support function modeling and physics-based reasoning in early design. Coverage, consistency, and validity are considered generic requirements of any formal representation (Summers & Shah, 2004; Luger, 2005). The other three are specific to this representation, based on gaps in function-based design (Sen et al., 2011*b*, 2013). The requirements are the following:

1. coverage
2. consistency
3. validity against physics laws
4. physics-based concreteness
5. normative and descriptive modeling
6. qualitative modeling and reasoning

4.1. Coverage

Coverage refers to the knowledge domains, the objects and phenomena from which can be modeled and reasoned by the representation (Baader, 1996; Summers, 2005*a*). For example, if representation A supports modeling electrical devices and representation B supports both electrical and acoustic domains, B has broader coverage than A.

Mechanical devices use a variety of physics principles for their operation. The principles ultimately used to solve a novel design problem are difficult to foresee in early design. In the interest of broad coverage, the representation must support modeling and reasoning a variety of physical phenomena and principles. Broader coverage usually comes at a cost of reasoning accuracy and efficiency (Summers, 2005*b*). At one extreme, expert systems such as rule-based tools (Chavali et al., 2008) use knowledge specific to the task at hand and perform fast and accurate reasoning (low coverage). However, they cannot perform a different task even with less speed or accuracy. The rule set used in these systems is the result of evolutionary refinement through generations of similar past designs, an opportunity that does not exist in novel designs. At the other end of this spectrum are the general-purpose CAD, computer-aided engineering, and control flow diagram tools that can model and analyze systems from virtually any domain (high coverage), but the designer must construct models for individual analyses. In an effort to bridge this gap, CAD companies either offer domain-specific tools such as NX Mold Wizard for injection mold design or enable user customization of CAD tools for specific tasks, such as DriveWorks. In respect to this spectrum, the requirement for the function representation is stated below.

> The representation should support modeling and reasoning with principles and devices commonly used/discussed in physics and mechanical engineering. Specifically, phenomena involving electrical, mechanical, and thermal energy and their interaction with various material forms must be supported.

4.2. Consistency

Consistency ensures that assertions within the representation or statements logically derived from them do not contradict mutually (Luger, 2005). Consistency is an internal property. It only prevents "mutual" conflict and does not require that the statements be valid against external knowledge. Sen et al. (2011*a*) illustrates typical inconsistencies found in function vocabularies. For example, the definition of branch in the functional basis (Hirtz et al., 2002), when taken literally as a formal statement, implies two contradictory assertions: the incoming flow must be branched into multiple flows, and the function must not produce multiple output flows. Although the functional basis is widely used for function modeling by human designers, this inconsistency is prohibitive of formal reasoning.

> The representation should be internally consistent. More formally, it should be impossible to derive two statements P and Q from the assertions within the representation such that $P = \neg Q$.

4.3. Validity against physics laws

Validity against an external knowledge ensures the impossibility to start from a premise that is true against that knowledge and draw inferences that are false against the same knowledge using reasoning supported by the representation (Bergmann et al., 1990). Unlike consistency, validity is an external property. It checks for agreement between assertions of the representation (or their logical derivatives) and external knowledge: here, the principles of conservation and irreversibility. To this end, the requirement is the following:

> The representation must be valid against the principles of conservation and irreversibility. If a model implies a violation of these principles, the reasoning algorithms to detect that violation must be supported.

4.4. Physics-based definitions

For supporting physics-based reasoning, the concepts in the representation must be defined in terms of physics-based actions. In function literature, the varying level of definitions has been recognized (Lind, 1994; Chandrasekaran, 2005). The need for the function representation is illustrated through the following example. The overall action of a hairdryer can be stated as the following:

1. "It allows the user to dry hair with a stream of hot air."
2. "It dries hair with a stream of hot air."
3. "It produces a stream of hot air for drying hair."
4. "It produces a stream of hot air."
5. "It transforms cold air at rest to hot air in motion."
6. "It converts electrical energy to heat and kinetic energy and adds them to air."

https://doi.org/10.1017/S0890060413000292 Published online by Cambridge University Press

7. "It converts electrical energy to heat using a heater and to the kinetic energy of a fan."
8. (All of 7) + "It rejects part of the input electrical energy as other forms."

Each statement is correct, but they are increasingly objective and definitively physical. The first sentence describes what a person could do with the device, similar to affordances (Maier & Fadel, 2009), without mentioning its physical actions. The last two sentences focus on the physical actions without describing what someone could do with it. The middle steps incrementally drop the user (2), the effect of drying hair (3), and the perceived purpose: "for drying hair" (4). Later statements are focused on the actions performed (5) and their physics (6). The last two sentences, (7) and (8), even drop the surrounding air. In this view, the function is limited to spinning the fan and producing heat; the effect of blowing hot air is contingent on the device being immersed in air, and thus is not a necessary part of its function. For the function representation, the objectivity of Sentence (6), plus the losses described in (8) seem appropriate, because (6) is the most objective level that does not include form descriptions, such as heater or fan. An effort to define the level of abstractions including qualitative and quantitative reasoning support is illustrated in Chandrasekaran and Josephson (2000).

Existing function vocabularies do not define the terms with this physics-based objectivity, because they were not meant to support physics-based computation. For example, the verb separate is defined in the functional basis as "To isolate a flow (material, energy, signal) into distinct components. The separated components are distinct from the flow before separation, as well as each other" (Hirtz et al., 2002). The verb is frequently used in the Design Repository (http://repository.designengineeringlab.org), two of which are the function of a can opener (separates the lid from the can) and the filter in a vacuum cleaner (separates dust from the air + dust mixture). Both applications are in agreement with the definition, because the incoming material flow is a conjoined form of the outgoing material and at exit the two flows are distinct from each other. However, as seen in Figure 2, the physics of these two actions are quite different. The can opener consumes external energy, and mechanical work is done on the material flow: **E added to M**. The filter does not consume external energy; it removes kinetic energy from the dust particles completely (stopping) and from the air partially (slowing down). Energy released to the surroundings (heat and sound) is extracted from the material flow itself: **E removed from M**. The definition of *separate* applies notionally to both functions, but it describes two different physical actions, which shows that the definition is not indicative of a specific physics-based action. Thus, physics-based objectivity is not realized in this definition. The recent literature proposes two new verbs, energize material and de-energize material, which describe the atomic actions of adding and removing energy or work from material flows (Sen et al., 2011*a*, 2013). These verbs could capture the distinction between the two cases. The need for physics-based objectivity is summarized below.

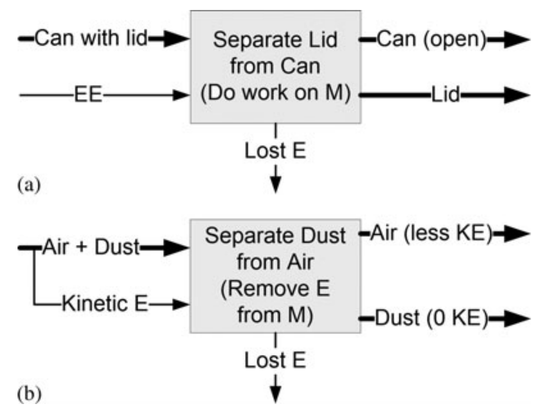

Fig. 2. An analysis of separate functions as applied in two models.

> The entities, relations, attributes, and grammar of the representation must be defined in terms of physical actions, specifically to support reasoning against the conservation and irreversibility principles.

4.5. Normative and descriptive modeling support

Normative means "what should be" and descriptive means "what is." Engineering begins with a need for a solution (normative) and ends with a solution (descriptive; Simon, 1998). A *normative function structure* describes the intended functions and flows in their ideal topologic arrangement. A *descriptive function structure* describes the actual actions, flows, and topologic arrangement that occur in a device during a mode of use. These definitions are applied to the different function modeling approaches of both the AI and design points of views (Table 1).

Although the terms normative and descriptive apply to models, they distinguish models by their intended use, rather than their content. The definitions do not assume completeness, correctness, consistency, validity, or feasibility. A normative model could, as such, violate laws of physics, possibly due to incomplete modeling, and still be a useful representation of the need. Similarly, a descriptive model does not guarantee that the designer successfully observed every functional detail. Physics-based analyses, such as detecting violations of balance laws or irreversibility, could be applied to both models. For normative models, they could evaluate the theoretical feasibility or efficiency of a normative model. For descriptive models, they could ensure that no flows, specifically losses, are omitted. Finally, a representation that supports both normative and descriptive models would allow for a systematic reformulation (Gero & Kannengiesser, 2002) by exposing the gap between the normative (wish) and descriptive (reality). In this manner, a representation that supports both normative and descriptive modeling can begin to span the two

https://doi.org/10.1017/S0890060413000292 Published online by Cambridge University Press

Table 1. *Characterization of existing function representations*

Modeling Approach	Normative	Descriptive	Comments
Function as effect (Chandrasekaran & Josephson, 2000)		X	Used for causal reasoning
Causal function representation language (Vescovi et al., 1993)		X	Used for causal reasoning
Function–behavior–state (Umeda et al., 1996)	X		Design new artifacts and analyze behavior
Functional basis (Hirtz et al., 2002)		X	Created by reverse engineering
Collins' vocabulary (Collins et al., 1976)		X	Created by reverse engineering
Function–behavior based tools (Bohm et al., 2005)		X	Morphological combinations of existing solution fragments described functionally
Function–behavior based tools (Kurtoglu et al., 2010).	X		Graph-grammar tools use prepopulated models as knowledge base
Function–behavior–structure model (Gero & Kannengiesser, 2002)	X	X	Provides explanation of how design solution evolves by comparing normative to descriptive states

different views of functions, both AI and design. The requirement is summarized below.

> The representation must support descriptive modeling of existing devices, concepts, or physical principles. It must also support normative modeling of new design concepts.

4.6. Qualitative and quantitative modeling and reasoning support

Qualitative physics is a technique of using physics for qualitative reasoning of confluences (De Kleer & Brown, 1984; Forbus, 1984), mainly to estimate how the increase or decrease of a parameter in a system causes a change in other parameters, without using quantitative values. For function structure models, *confluence-based* reasoning would be to infer from Figure 3a that increasing the power of the input electrical energy will cause an increase in the mechanical energy output or that Figure 3a violates *conservation*, because it does not produce a material output despite receiving one. Similarly, Figure 3b violates *irreversibility*, because it suggests that the input electrical energy is entirely converted into mechanical energy without loss. This reasoning can detect modeler-inflicted inconsistency. An example of quantitative reasoning is to infer that the input electrical energy in Figure 3b must be supplied at a rate of 4825.7 Watts, using the power of the output mechanical energy and the efficiency of the function. This quantitative reasoning requires quantitative information, which the representation must support. The capability of different representations to support quantitative and qualitative reasoning is illustrated in Table 2.

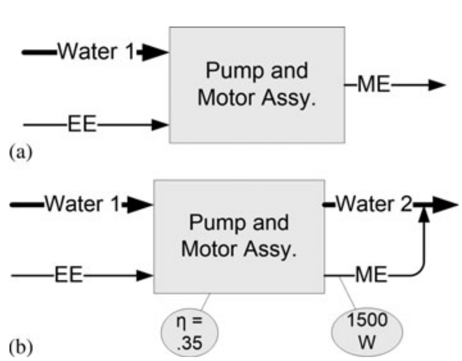

Fig. 3. The function model for qualitative and quantitative reasoning.

Qualitative reasoning will be more useful in early design, when quantitative information is not available. However, the designer should be allowed to evolve the model as a reflection of his thoughts, which is likely to progress from qualitative to quantitative details in later design stages. Thus, the requirements are summarized below.

> *Qualitative:* It must be possible to model concepts qualitatively, when quantitative information is not available. It must be possible to perform qualitative reasoning on the concepts using the conservation and irreversibility principles, and confluence.
>
> *Quantitative:* It must be possible to add quantitative data to qualitative models subsequently. Quantitative reasoning for calculating efficiency, power required and produced, mass flow, volume flow, and other physical parameters of interest should be supported on these models.

5. DISCUSSION

Six requirements are identified above for formalizing function structure graphs to support modeling and formal analysis of concepts. Coverage, consistency, and validity are generic requirements of formal logic: they are articulated here for function modeling. Three additional requirements are iden-

https://doi.org/10.1017/S0890060413000292 Published online by Cambridge University Press

Table 2. *Quantiative and Qualitative Reasoning Support*

Representation	Qualitative	Quantitative	Comments
Function structure of design repository (Bohm et al., 2005)	X	Inconclusive evidence	Some models include numerical information, but not uniformly
Function–behavior graph grammar (Sridharan & Campbell, 2005)	X		Primarily pattern matching
Functional representation (Chandrasekaran, 2005)	X		No static vocabulary
Causal function representation language (Chandrasekaran & Josephson, 2000)	X		User defined vocabulary can include nonnumeric parameters
Function–behavior–state model (Umeda et al., 1996)	X		
Structure–behavior–function model (Goel et al., 1997; Goel & Bhatta, 2004)	X		
Function ontologies (Sasajima et al., 1995; Kim & Duffy, 2003; Garbacz, 2005; Cebrian-Tarrason et al., 2008)	X		

tified based on research gaps. In each case, the gap is established with analysis of other representations.

It is anticipated that this representation will address the general lack of intelligent automation support in early design, specifically in concept-level analysis. The envisioned software can be used to model the normative functionality of a design and examine or explore design variants on the computer, while getting analytical feedback from it, in a much similar manner because geometric CAD today supports modeling and analysis (both real-time and postmodeling). Advances in this direction have been made in the past years. Specifically, Sen et al. (2011*a*) describes a protocol to derive formal definitions of existing function verbs, Sen et al. (2011*b*) illustrates the potential of automated reasoning using this representation, and Sen (2012) presents a formal representation of functions that partially meets these requirements. Finally, Sen (2012) illustrates a software tool, Concept Modeler, which addresses the requirements of coverage, consistency, validity, domain neutrality, qualitative reasoning, and quantitative reasoning. The tool currently supports descriptive modeling better than normative modeling, thus that requirement is not adequately addressed.

Although a set of requirements are offered, the relative importance of these requirements is not discussed. As with any development effort, trade-offs between requirements and the level of satisfaction of the requirements is necessary. These trade-offs should be driven from contextual and specific use cases developed for function reasoning. One must recognize that several different representations might meet the requirements, but with different levels of performance, and might be better suited for different types of reasoning activities. This paper does not try to impose a valuation on the importance of the requirements but to simply define and justify why each should be considered.

Ultimately, this paper is intended to provide a starting point for developing a consistent and canonical set of requirements for function reasoning, representation, and modeling. These requirements are necessary for clearly defining the potential capabilities of the proliferation of different approaches and views for different intents. It is strongly recommended that other researchers take these requirements as a starting point for self-critique on their developed approaches or use these proposed and justified requirements as motivation for refinement and future work. These are not intended to be the comprehensive set of requirements but a fundamental collection.

REFERENCES

Albers, A., Thau, S., & Alink, T. (2008). Support of design engineering activity through C & CM- temporal decomposition of design problems. *Proc. Tools and Methods for Competitive Engineering Conf.,* pp. 295–306, Izmir, Turkey.

Baader, F. (1996). A formal definition for the expressive power of terminological knowledge representation languages. *Journal of Logic and Computation 6(1)*, 33–54.

Bergmann, M., Moor, J., & Nelson, J. (1990). *The Logic Book*. New York: McGraw–Hill.

Bohm, M., Stone, R., & Szykman, S. (2005). Enhancing virtual product representations for advanced design repository systems. *ASME Transactions Journal of Computing and Information Science in Engineering 5(4)*, 360–372.

Cebrian-Tarrason, D., Lopez-Montero, J.A., & Vidal, R. (2008). OntoFaBeS: ontology design based in FBS framework. *Proc. CIRP Design Conference: Design Synthesis*. Enschede, Netherlands: CIRP.

Chakrabarti, A., & Bligh, T.P. (2001). A scheme for functional reasoning in conceptual design. *Design Studies 22(6)*, 493–517.

Chandrasekaran, B. (2005). Representing function: relating functional representation and functional modeling research streams. *Artificial Intelligence in Engineering Design, Analysis and Manufacturing 19*, 65–74.

Chandrasekaran, B., & Josephson, J. (2000). Function in device representation. *Engineering With Computers 16(3–4)*, 162–177.

Chavali, S.R.K., Sen, C., Mocko, G.M., & Summers, J.D. (2008). Using rule-based design in engineer-to-order industry: an SME case study. *Computer-Aided Design and Applications 5(1–4)*, 178–193.

Collins, J.A., Hagan, B.T., & Bratt, H.M. (1976). The failure-experience matrix—a useful design tool. *Journal of Engineering for Industry 98(3)*, 1074.

De Kleer, J., & Brown, J.S. (1984). A qualitative physics based on confluences. *Artificial Intelligence 24(1)*, 7–83.

Deng, Y.M. (2002). Function and behavior representation in conceptual mechanical design. *Artificial Intelligence in Engineering Design, Analysis and Manufacturing 16(5)*, 343–362.

Eastman, C.M. (1969). Cognitive processes and ill-defined problems: a case study from design. *Proc. Joint Int. Conf. Artificial Intelligence*, pp. 669–690. San Francisco, CA: Morgan Kaufmann.

Eckert, C. (2013). That which is not form: the practical challenges in using functional concepts in design. *Artificial Intelligence in Engineering Design, Analysis and Manufacturing 27(3)*, 217–232 [this issue].

Fenves, S.J., Foufou, S., Bock, C., & Sriram, R.D. (2008). CPM2: a core model for product data. *Journal of Computing and Information Science in Engineering 8*, 14501.

Forbus, K.D. (1984). Qualitative process theory. *Artificial Intelligence 24(1)*, 85–168.

Freeman, P., & Newell, A. (1971). A model for functional reasoning in design. *Proc. 2nd Int. Joint Conf. Artificial Intelligence*, pp. 621–640. San Francisco, CA: Morgan Kaufmann.

Garbacz, P. (2005). Towards a standard taxonomy of artifact functions. *Applied Ontology 1*, 221–236.

Gero, J., & Kannengiesser, U. (2002). The situated–function–behaviour–structure framework. In *Artificial Intelligence in Design* (Gero, J., Ed.), pp. 89–104. Norwell, MA: Kluwer Academic.

Goel, A.K. (2013). One 30-year case study and 15 principles: implications of an artificial intelligence methodology for functional modeling. *Artificial Intelligence in Engineering Design, Analysis and Manufacturing 27(3)*, 203–215 [this issue].

Goel, A.K., & Bhatta, S.R. (2004). Use of design patterns in analogy-based design. *Advanced Engineering Informatics 18*, 85–94.

Goel, A.K., Bhatta, S.R., & Stroulia, E. (1997). KRITIK: an early case-based design system. In *Issues and Applications of Case-Based Reasoning in Design* (Maher, M.L. & Pu, P., Eds.), pp. 87–132. Mahwah, NJ: Erlbaum.

Hirtz, J., Stone, R., McAdams, D., Szykman, S., & Wood, K. (2002). A functional basis for engineering design: reconciling and evolving previous efforts. *Research in Engineering Design 13*, 65–82.

Hubka, V., & Eder, W.E. (1988). *Theory of Technical Systems: A Total Concept Theory for Engineering Design*. Berlin: Springer–Verlag.

Kim, S., & Duffy, A.H.B. (2003). Towards an ontology of generic engineering design activities. *Research in Engineering Design 14*, 200–223.

Kirschman, C.F., & Fadel, G.M. (1998). Classifying functions for mechanical design. *Journal of Mechanical Design 120(3)*, 475–482.

Kitamura, Y., Kashiwase, M., Fuse, M., & Mizoguchi, R. (2004). Deployment of an ontological framework of functional design knowledge. *Advanced Engineering Informatics 18(2)*, 115–127.

Kroes, P. (2010). Formalization of technical functions: why is it so difficult. *Proc. Tools and Methods for Competitive Engineering Conf. (TMCE 2010)*, Ancona, Italy, April 12–16.

Kurtoglu, T., Swantner, A., & Campbell, M. (2010). Automating the conceptual design process: from black box to component selection. *Artificial Intelligence in Engineering Design, Analysis and Manufacturing 24(1)*, 49–62.

Lind, M. (1994). Modeling goals and functions of complex industrial plants. *Applied Artificial Intelligence an International Journal 8(2)*, 259–283.

Linz, P. (2011). *An Introduction to Formal Languages and Automata* (5th ed.). Sudberry, MA: Jones & Bartlett.

Luger, G.F. (2005). *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. Essex: Addison–Wesley Longman.

Maier, J.R.A., & Fadel, G.M. (2009). Affordance-based design methods for innovative design, redesign and reverse engineering. *Research in Engineering Design 20(4)*, 225–239.

McAdams, D., & Wood, K. (2002). A quantiative similarity metric for design-by-analogy. *ASME Transactions Journal of Mechanical Design 124(2)*, 173–182.

Nagel, R.L., Perry, K.L., Stone, R.B., & McAdams, D.A. (2009). FunctionCAD: a functional modeling application based on the function design framework. *ASME Int. Design Engineering Technical Conf. Computers and Information in Engineering Conf.*, San Diego, CA.

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., Wallace, K., & Blessing, L. (2007). *Engineering Design: A Systematic Approach* (3rd ed.). London: Springer–Verlag.

Rodenacker, W. (1971). *Methodisches Konstruieren*. Berlin: Springer–Verlag.

Sasajima, M., Kitamura, Y., Ikeda, M., & Mizoguchi, R. (1995). FBRL: a function and behavior representation language. *Proc. Int. Joint Conf. Artificial Intelligence*, pp. 1830–1836. San Francisco, CA: Morgan Kaufmann.

Sen, C. (2012). *A formal representation of mechanical functions to support physics-based computational reasoning in early mechanical design*. PhD Thesis. Clemson University.

Sen, C., Summers, J.D., & Mocko, G.M. (2011*a*). A protocol to formalise function verbs to support conservation-based model checking. *Journal of Engineering Design 22 (11–12)*, 765–788.

Sen, C., Summers, J.D., & Mocko, G.M. (2011*b*). Exploring potentials for conservational reasoning using topologic rules of function structure graphs. *Proc. 18th Int. Conf. Engineering Design (ICED11)*, Vol. 9, pp. 377–388. Copenhagen: Design Society.

Sen, C., Summers, J.D., & Mocko, G.M. (2013). A formal representation of function structure graphs for computer-directed modeling and conservation-based reasoning. *ASME Transactions Journal of Computing and Information Science in Engineering 13(1)*.

Simon, H. (1998). *The Sciences of the Artificial*. Cambridge, MA: MIT Press.

Sridharan, P., & Campbell, M. (2005). A study on the grammatical construction of function structures. *Artificial Intelligence in Engineering Design, Analysis and Manufacturing 19(3)*, 139–160.

Srinivasan, V., Chakrabarti, A., & Lindemann, U. (2012). A framework for describing functions in design. *Proc. Int. Design Conf.*, pp. 1111–1122. Dubrovnik: Design Society.

Stone, R.B., Tumer, I.Y., & Van Wie, M. (2005). The function-failure design method. *ASME Transactions Journal of Mechanical Design 127(3)*, 397–407.

Summers, J.D. (2005*a*). Expressiveness of the design exemplar. *Proc. Int. Design Engineering Technical Conf. Computers and Information in Engineering Conf.*, Paper No. CIE-85135. Long Beach, CA: ASME.

Summers, J.D. (2005*b*). Reasoning in engineering design. *Proc. Int. Design Engineering Technical Conf. Computers and Information in Engineering Conf.*, Paper No. DTM-85334. Long Beach, CA: ASME.

Summers, J.D., & Shah, J.J. (2004). Representation in engineering design: a framework for classification. *Proc. Int. Design Engineering Technical Conf. Computers and Information in Engineering Conf.*, Paper No. DTM-57514, Salt Lake City, UT.

Thomke, S., & Fujimoto, T. (2000). The effect of "front-loading" problem-solving on product development performance. *Journal of Product Innovation Management 17(2)*, 128–142.

Ullman, D.G. (2010). *The Mechanical Design Process*. New York: McGraw–Hill.

Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y., & Tomiyama, T. (1996). Supporting conceptual design based on the function–behavior–state modeler. *Artificial Intelligence in Engineering Design, Analysis and Manufacturing 10(4)*, 275–288.

Van Eck, D., McAdams, D.A., & Vermaas, P.E. (2007). Functional decomposition in engineering: a survey. *Proc. ASME Int. Design Engineering Technical Conf. Computers and Information in Engineering Conf.*, Las Vegas, NV.

Vermaas, P.E. (2013). The coexistence of engineering meanings of function: four responses and their methodological implications. *Artificial Intelligence in Engineering Design, Analysis and Manufacturing 27(3)*, 191–202 [this issue].

Vescovi, M., Iwasaki, Y., Fikes, R., & Chandrasekaran, B. (1993). CFRL: a language for specifying the causal functionality of engineered devices. *Proc. 11th National Conf. Artificial Intelligence*. Washington, DC: American Association for Artificial Intelligence.

Vucovich, J., Bhardwaj, N., Hoi-Hei, H., Ramakrishna, M., Thakur, M., & Stone, R. (2006). Concept generation algorithms for repository-based early design. *Proc. ASME Int. Design Engineering Technical Conf. Computers and Information in Engineering Conf.*, Paper No. DETC2006–99466, Philadelphia, PA.

Wang, C., Qing-liang, Z., & Albers, A. (2009). A unified product model based on C&CM for mechatronic product development. *Proc. Int. Conf. Mechatronics and Automation, ICMA 2009*, pp. 2173–2177, Changchun, Jilin, China, August 9–12.

Chiradeep Sen is a Senior Engineer at Dresser-Rand Co., NA, and a courtesy faculty member at the School of Mechanical, Industrial, and Manufacturing Engineering at Oregon State University. He earned his PhD and MS in mechanical engineering from the CEDAR Lab at Clemson University and his BE in mechanical engineering from Jadavpur University. Dr. Sen has worked in industry at various design and manufacturing positions for 12 years. His research interest is in formal representation and reasoning to support concep-

tual design, cognitive studies to understand design synthesis and analogy, knowledge-based engineering systems, and design informatics.

Joshua D. Summers is a Professor of mechanical engineering and IDEaS Professor at Clemson University, where he is the Codirector of the Clemson Engineering Design Applications and Research Group. He earned his PhD in mechanical engineering from Arizona State University (design automation) and his MS (submarine design) and BS (fluidized bed design) from the University of Missouri. Dr. Summers has worked at the Naval Research Laboratory (VR Lab and NCARAI). His research has been funded by government, large industry, and small- to medium-sized enterprises. His areas of interest include collaborative design, knowledge management, and design enabler development with the overall objective of improving design through collaboration and computation.