
POSITION PAPER

A 30-year case study and 15 principles: Implications of an artificial intelligence methodology for functional modeling

ASHOK K. GOEL

Design & Intelligence Laboratory, School of Interactive Computing, Georgia Institute of Technology, Atlanta, Georgia, USA

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

Abstract

Research on design and analysis of complex systems has led to many functional representations with several meanings of function. This work on conceptual design uses a family of representations called structure–behavior–function (SBF) models. The SBF family ranges from behavior–function models of abstract design patterns to drawing–shape–SBF models that couple SBF models with visuospatial knowledge of technological systems. Development of SBF modeling is an instance of cognitively oriented artificial intelligence research that seeks to understand human cognition and build intelligent agents for addressing complex tasks such as design. This paper first traces the development of SBF modeling as our perspective on design evolved from that of problem solving to that of memory and learning. Next, the development of SBF modeling as a case study is used to abstract some of the core principles of an artificial intelligence methodology for functional modeling. Finally, some implications of the artificial intelligence methodology for different meanings of function are examined.

Keywords: Conceptual Design; Functional Modeling; Functional Reasoning; Functional Representation; Structure–Behavior–Function Models; Systems Thinking

1. INTRODUCTION

Research on design and analysis of complex systems has led to many functional representations (FRs) with several meanings of function (e.g., Hubka & Eder, 1988; Andreasen, 1991; Chittaro et al., 1993; Bracewell & Sharpe, 1996; Pahl & Beitz, 1996; Stone & Wood, 2000; Chakrabarti & Bligh, 2001; Hirtz et al., 2002; Carrara et al., 2011; Eckert et al., 2011). Erden et al. (2008) and Garbacz et al. (2011) present two recent reviews of what is common among some of the FRs, and what is different, why, and how. In two companion position articles in this Special Issue of *AI EDAM*, my colleagues Pieter E. Vermaas (2013) and Claudia Eckert (2013) examine several meanings of function in the design literature and design practice, respectively.

We posit that some differences in the meanings of function may arise due to different goals and methodologies of the various FRs. Thus, a comparison of different FRs requires a clear and precise articulation of their goals and methodologies. In this article, first we conduct a reflective study of a long line of artificial intelligence (AI) research on FR. We want to under-

stand what aspects of the representation remained stable over time, and why; and what representational elements changed, how and why? Next, we use this analysis as a case study to abstract some of the core principles of one specific AI methodology for functional modeling. Finally, we examine some implications of the AI methodology for different meanings of function.

We start with the FR scheme that Chandrasekaran and colleagues at Ohio State University's Laboratory for Artificial Intelligence Research developed in the early to mid-1980s (Sembugamoorthy & Chandrasekaran, 1986; Chandrasekaran et al., 1993; Chandrasekaran, 1994a, 1994b). Initially, the domain of interest was medicine, the task was diagnosis, and the context was automated reasoning in intelligent agents. Later, Chandrasekaran and his students, including me, used the FR scheme for supporting automated reasoning for a variety of tasks such as design, diagnosis, and explanation, in domains ranging from engineering to medicine (Goel & Chandrasekaran, 1988, 1989; Sticklen & Chandrasekaran, 1989; Keuneke, 1991; Chandrasekaran, 1994a). The cognitive perspective on FRs in this early work was that of problem solving.

In the late 1980s, we developed an intelligent agent for automated design in which retrieval and adaptation of past designs was grounded in FRs of the designs (Goel & Chandra-

Reprint requests to: Ashok K. Goel, Design & Intelligence Laboratory, School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA 30308. E-mail: goel@cc.gatech.edu

sekar, 1988; Goel, 1992; Goel & Chandrasekaran, 1992). This helped change our perspective on FRs from that of problem solving to that of memory and learning, led to new ontological commitments, and resulted in structure–behavior–function (SBF) models. In the mid-1990s, Sambasiva Bhatta in our Design & Intelligence Laboratory at Georgia Institute of Technology developed a technique called *model-based analogy* that uses SBF models of designs to abstract and transfer behavior–function (BF) models of design patterns from one design domain to another (Bhatta & Goel, 1994, 1996a, 1996b, 1997). In the middle of the last decade, Patrick Yaner at Georgia Tech’s Design & Intelligence Laboratory developed a technique called *compositional analogy* for understanding design drawings by analogical mapping and transfer at multiple levels of abstraction in drawing–shape–SBF (DSSBF) models (Yaner & Goel, 2007a, 2007b, 2008).

Research on the FR scheme and SBF models is an instance of cognitively oriented AI research on functional modeling. The twin goals of this kind of AI research are to understand human cognition and to build cognitively inspired intelligent agents (Goel & Davies, 2011; Langley, 2012). The general methodology is to observe human behavior as they address complex tasks such as diagnosis and design, build knowledge-based intelligent agents for addressing the tasks, use the knowledge-based methods as hypotheses for explaining human reasoning on the tasks, and build interactive tools that use the representations and knowledge-based methods for aiding humans in performing the tasks. Unlike cognitive psychology that often focuses on the microstructure of human cognition and behavior, cognitively oriented AI typically emphasizes the macrostructure: the content and representation of knowledge, the inferences enabled by the knowledge, and the tasks addressed by the inferences.

In cognitively oriented AI, knowledge content and representation are closely related to the reasoning task as noted above. This leads us to the first AI principle for functional modeling:

Principle 1: Functional models are constructed to support functional reasoning. Thus, any theory of functional modeling needs to be situated in a theory of functional reasoning: the whys, whats and hows of functional reasoning for various tasks are good starting points for answering the whys, whats and hows of functional models.

AI is a wide and open field, and different AI researchers make different assumptions and pursue different hypotheses. Thus, several researchers have developed similar cognitively oriented FRs with similar names, goals, and foci. FRs that share some aspects or elements of our SBF models include Freeman and Newell’s (1971) representations of software systems, Reiger and Grinberg’s (1978) representations of physical mechanisms, Rasmussen’s (1985) structure–behavior–function representations of large-scale industrial systems, Gero’s function–behavior–structure models (Gero, 1990; Gero et al., 1992; Gero & Kannengiesser, 2004), Umeda

and Tomiyama’s function–behavior–state models (Umeda et al., 1990, 1996; Umeda & Tomiyama, 1997), Govindaraj’s qualitative approximations of complex systems (Vasandani & Govindaraj, 1995), and Kitamura and Mizoguchi’s function–behavior representations (Sasajima et al., 1996; Kitamura et al., 2002, 2004). However, in this article we focus exclusively on the evolution of our SBF models over nearly 30 years.

2. SYSTEMS AND MODELS

A complex system manifests a large number of heterogeneous interacting components, nonlinear and cyclic processes, feedback, stochasticity, and/or emergence (Forrester, 1994). From a cognitive perspective, it is useful to think of complexity as a metaphor for the kinds of systems that are challenging for humans to fully understand. A system may be cognitively challenging to understand not only because of the properties mentioned above but also because the causal processes in the system may be invisible and occurring at multiple levels of organization.

One common class of complex systems exhibits hierarchical organization (Simon, 1962, 1996, 1999). Hierarchically organized systems are nearly decomposable: although causal processes at one abstraction level in these systems emerge out of interactions among components and processes at lower levels, the interactions among subsystems at any level are contained and can be organized hierarchically. Functions are mental abstractions that enable a hierarchical decomposition of a system into subsystems (Simon, 1962, 1996, 1999). This is an important cognitive feature because it implies that analyzing such systems entails decomposing them into appropriate subsystems at different levels of functional abstraction and understanding interactions among the subsystems at a given level of functional abstraction. This leads us to our second principle for functional modeling.

Principle 2: Functions are mental abstractions that enable a hierarchical decomposition of a complex system into subsystems.

We use the term *model* (not *representation*) in referring to SBF models. Although the word *model* has been applied to a variety of representations, we use the term in the sense of a scientific model: a scientific model is a scientist’s interpretation of a phenomenon (e.g., Nersessian, 1999; Machamer et al., 2000; Nersessian, 2008). We note two important cognitive features of such models. First, models are abstractions of reality: the abstractions constrain systems thinking by both simplifying the systems and suggesting a course of analysis (Darden, 1998). Second, scientists use models to generate explanations: modern philosophers of science regard models as essential to explanation (e.g., Bechtel & Richardson, 2010).

Technological systems are designed, monitored, measured, explained, diagnosed, repaired, and redesigned to achieve the desired functions. Analysis of technological systems is facili-

tated by knowledge of causal mechanisms that result in the system's functions. It follows that functional models should explain how a system's causal mechanisms accomplish its functions. This leads to our third principle for functional modeling.

Principle 3: Functional models specify causal mechanisms that explain how system functions are accomplished.

3. FR SCHEME FOR DIAGNOSIS AND EXPLANATION

In the early to the mid-1980s, Sembugamoorthy and Chandrasekaran (1986) developed the FR scheme for describing the functioning of physical devices to support automated diagnosis. Let us consider the example of the simple gyroscope follow-up illustrated in Figure 1. Briefly, the gyrocompass, with its ability to track true north as compared to the unreliable magnetic north, is an instrument for navigation and piloting aboard many ships. A gyroscope is an assembly with a very rapidly spinning top. A gyroscope follow-up automatically tracks and amplifies the movement of a spinning gyro. The follow-up servo can drive any number of gyrocompasses located anywhere on a ship, each of which replicates the reading of the central gyro. Figure 1 illustrates the schematic of a simple gyroscope follow-up with no feedback control. This example actually is taken from the Ideal system described in Section 5. However, we use it here to explain the FR scheme and to illustrate the evolution of SBF modeling.

Figure 2a and b illustrate the representation of the function and the behavior of the gyroscope follow-up in the FR scheme. The description of the structure of the gyroscope follow-up in the FR scheme (not shown here) would include specifications of the components of the device as well as the connections among them. The description of a function (Fig. 2a) specifies the behavior that accomplishes it; thus, functions in the FR scheme index behaviors responsible for them, and the behaviors are organized by the functions they achieve. The description of a behavior (Fig. 2b) specifies a causal mechanism consisting of states and transitions among them. The specification of a device function may also include

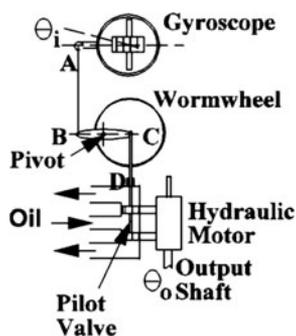


Fig. 1. A schematic diagram of the gyroscope follow-up.

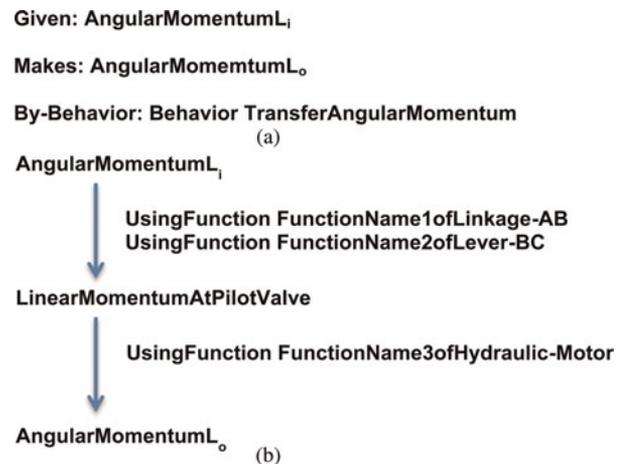


Fig. 2. Representations of (a) a function in the functional representation (FR) scheme and (b) behavior in the FR scheme. [A color version of this figure can be viewed online at <http://journals.cambridge.org/aie>]

an **ExternalStimulus** clause that specifies an external trigger and a **Provided** clause that specifies the conditions under which the behavior accomplishes the function. A transition in a behavior may be annotated by explanations for it, for example, **UsingFunction** of some component of the device (Fig. 2b). Annotations on the transitions may include **By-Behavior** (for a more detailed description of the transition), **UnderStructuralConditions** (for specifying the structural conditions for the transition), and **DomainPrinciple** (for specifying a general principle that explains the transition).

Note that a function of a system in the FR scheme is an intended output behavior of the system as well as an abstraction of an internal causal behavior. Note also that the annotation **UsingFunction** on the behavioral transitions provides a decomposition of the function of the system into functions of its components. However, instead of a function–subfunction decomposition, in the FR scheme, internal causal behaviors are an intermediate level of abstraction between function and structure. Thus, the abstraction hierarchy takes the form of $F \rightarrow B \rightarrow F \rightarrow B \dots \rightarrow F(S)$, where the decomposition can go to as many levels as needed. Sticklen and Chandrasekaran (1989) describe MDX2, an intelligent agent that uses the FR scheme for automated diagnosis. Keuneke (1991) describes the use of the FR scheme for automated explanation of the functioning of complex systems. The FR scheme leads to three additional principles for functional modeling.

Principle 4: Behavior is an intermediate abstraction between structure and function. Internal causal behaviors of a system specify the composition of a system function from the functions of its subsystems and components.

Principle 5: Organization of functional models is important for localizing reasoning. Functions may act as indices into behaviors; behaviors may be organized around functions.

Principle 6: Functional models need to be evaluated through experimentation. Construction of knowledge systems for

automated reasoning is one experimental method for testing, refining, and revising functional models.

4. CASE-BASED DESIGN: SBF MODELS OF DESIGNS

My 1989 PhD dissertation at Ohio State University under Chandrasekaran's supervision integrated case-based and model-based reasoning for adaptive design (Goel, 1989). Much of practical design is adaptive in that new designs are generated by adapting known designs. In the paradigm of case-based reasoning (Riesbeck & Schank, 1989; Kolodner, 1993), the intelligent agent is endowed with a memory populated with past experiences (called cases). Given a new problem, the agent retrieves a similar case from memory and, if needed, modifies it to suit the given problem. The agent evaluates the candidate solution and, if needed, repairs it. When the agent encounters a new case, it indexes and stores it in memory for potential reuse. While case-based reasoning provides a computational process for reasoning, it does not provide a content account of cases (i.e., the case-based reasoning does not specify what knowledge should be contained in a case). The knowledge contained in the cases should support the tasks of indexing, retrieval, modification, evaluation, and storage.

My dissertation described the use of functional models of known designs as the contents of design cases and the functions of the designs as indices to the design cases (Goel, 1992). The core of the task of conceptual design is an $F \rightarrow S$ mapping: it takes a specification of a set of desired functions as input and has the goal of giving as output a specification of the structure of a design that can achieve the desired functions. For example, a target design problem may specify the desired function of propagating angular momentum with only small fluctuations. Hence, the proposal to index design cases by the functions delivered by the stored designs so that the knowledge system can retrieve the design case most similar to the desired function. Further, if each design case contains an $S \rightarrow B \rightarrow F$ map, where the behavior B specifies how the structure S of the design achieved its function F , then the map may guide the modification of the design to achieve the desired function of the new problem. Hence, the proposal to include $S \rightarrow B \rightarrow F$ maps (or models) as the contents of the design cases, which explains the origin of the name of SBF modeling.

When I tried to use the FR scheme to specify the contents and indices of design cases, I encountered two problems. First, the FR scheme of the 1980s did not provide a vocabulary for specifying functions so that two functions could be compared to measure similarity between them. This is because the **Given** and **Makes** states in the function specification (e.g., **AngularMomentumL_i**) are simply strings of characters with little semantics. Thus, the FR scheme of the time could not support case retrieval. Second, the FR scheme of the 1980s did not provide a vocabulary for specifying functions so that one function could be substituted by another function during design case adaptation. This is because the

functional annotations on the behavioral state transitions (e.g., **FunctionName1ofLinkageAB**) are simply character strings with little semantics. Thus, the FR scheme of the time could not support case adaptation either.

To support design case retrieval and adaptation, we needed an ontology of states and functions. I adapted this ontology from a technique of compositional modeling called consolidation (Bylander & Chandrasekaran, 1985; Bylander, 1991). Consolidation is a technique for incrementally composing the output behaviors of a physical device from the output behaviors of its structural components. It subscribes to an ontology of components and substances, where a component can contain a substance and substances can flow between components. Components can have modes (such as *open* and *closed*). A substance can be any "stuff" that is conserved (unless explicitly *created* or *destroyed* by some component). Thus, a substance, for example, may be material, energy, or information; a substance may be physical or abstract (e.g., angular momentum).

This basic ontology leads to a small number of primitive output behaviors of components and devices (Bylander, 1991): *Allow*, *Move*, *Pump*, *Expel*, *Create*, *Destroy*, and *Change_Mode*. The technique of consolidation used a small set of domain-independent rules to compose potential output behaviors of a device from the output behaviors of its components. Note that the technique of consolidation focuses on the output behaviors of components and systems, not their internal causal behaviors as in the FR scheme. The functions of a system in the sense of the FR scheme are a desired subset of the set of all output behaviors. For example, a flashlight electrical circuit may have the output behaviors of producing light and generating a small amount of heat, where the desired function may only be to produce light. Although the FR scheme represented the functions of a system (e.g., production of light in case of the flashlight), SBF models represent all output behaviors of the system (e.g., production of light as well as generation of heat for the flashlight; Goel & Stroulia, 1996).

I used this ontology of components and substances, of output behaviors and functions, to develop a vocabulary for specifying the states in a system as well as functional annotations on the behavioral state transitions in the FR scheme. Figure 3a and b illustrate the representations of the gyroscope of Figure 1 in the new language. The representation of function in SBF models (Fig. 3a) may be compared with the representation of function in the FR scheme (Fig. 2a). Note that the representation of function in SBF models is no longer a string of characters. Instead, it has a semantics that enables comparison of two functions and measurement of similarity between them. This enabled retrieval of cases similar to the given problem based on the similarity between their respective functional specifications.

Similarly, the representation of behavior in SBF models (Fig. 3b) may be compared with the representation of behavior in the FR scheme (Fig. 2b). The functional annotations on the state transitions in the SBF models have a semantics that enables reasoning about the modifications needed to the re-

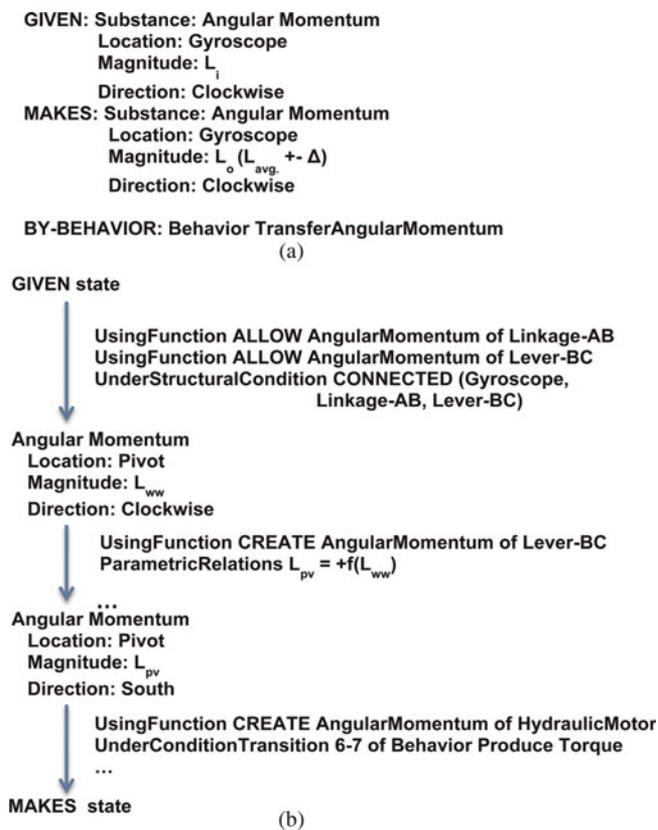


Fig. 3. Representations of (a) a function in structure–behavior–function (SBF) models and (b) behavior in SBF models. [A color version of this figure can be viewed online at <http://journals.cambridge.org/aie>]

tried design in order to achieve the desired function specified in the target problem. (The representation of behaviors illustrated here is only partial. For example, the last transition in the behavior **TransferAngularMomentum** depicted in Fig. 3b points to a state transition in another behavior **ProduceTorque** that is not shown in the figure.)

An intelligent agent called Kritik implemented the case-based method for adaptive design, grounding the case-based process in SBF models (Goel & Chandrasekaran, 1988; Goel, 1992; Goel & Chandrasekaran, 1992; Goel et al., 1997). The SBF models provided the vocabulary for indexing the design cases, as well as methods for retrieving, adapting, evaluating, and storing designs. Sycara et al. (1991) used another model-based scheme for indexing and retrieving design cases. Murdock et al. (1997) and Szykman et al. (2000) describe attempts at building large-scale libraries of design cases in engineering domains. The development of SBF models resulted in two more principles for functional modeling:

Principle 7: Functional modeling should support memory tasks in addition to problem-solving tasks. Memory tasks include case indexing, retrieval, and storage.

Principle 8: Functional modeling needs to be grounded in an ontology of systems. An ontology provides a disciplined vocabulary that in turn enables principled inferences.

5. ANALOGICAL DESIGN: BF MODELS OF DESIGN PATTERNS

The above theory of grounding the case-based process in functional models of design cases addressed conceptual design problems in which the desired function was almost identical to the function delivered by a design case so that known design needed to be modified only locally to obtain the desired design. The SBF models proved sufficient to support local modifications such as adjusting the parameters of some component or substance, or substituting some component in the design with another component at the same location as the first component. What additions to functional models might support more creative conceptual design, for example, problems requiring insertion of new components into the structure of the design case?

Let us consider as an example a design case containing the simple flashlight electrical circuit illustrated in Figure 4. (Briefly, the circuit contains a switch, a battery, and a bulb. When the switch is pressed, the circuit is closed, and the bulb produces light of an intensity that depends in part on the voltage of the battery.) Now suppose that the new design problem specifies the desired function to be several times the amount of light created by this system. One possible modification to the electrical system depicted in Figure 4 might be to replace the battery of 1.5 V in the circuit with a battery of higher voltage. However, let us further suppose that a battery of higher voltage is not available. What functional knowledge may enable an intelligent agent to adapt the design of this electrical circuit to achieve the desired function of generating light of a higher intensity?

For his PhD dissertation at Georgia Tech, Bhatta (1995) developed the technique of model-based analogy for innovative design adaptations based on the notion of design patterns. A design pattern in architecture specifies an abstract design goal, a generic design strategy for achieving the design goal in the form of an abstract spatial arrangement of structural components, and applicability conditions for instantiating the abstract design strategy (Alexander, 1977). Bhatta adapted this notion of design patterns for conceptual design of engineering systems, once again introducing behavior as an intermediate abstraction between function and structure in the specification of design patterns. Figure 5a illustrates the design goal of the cascade (or repetition) design pattern in terms of the functional descriptions of a desired function and of the function delivered by a known design case. It specifies that the output of the desired function is many times the output

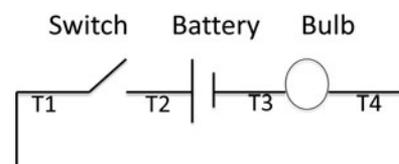


Fig. 4. A flashlight electrical circuit.

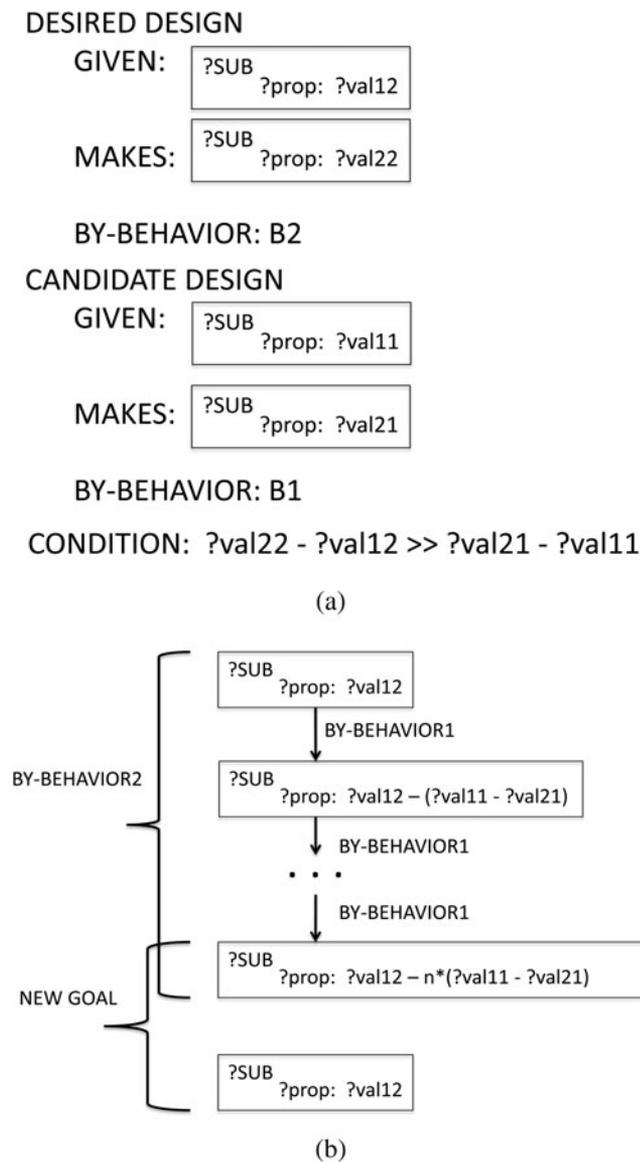


Fig. 5. (a) A functional description of the cascade design pattern and (b) a behavioral description of the cascade design pattern.

of the design case. **Figure 5b** illustrates the design strategy for the cascade design pattern in terms of description of how to compose the behaviors of the design case into a behavior for achieving the desired function, namely, repetition of the behavior of the design case. Applying this pattern to the problem of the electrical circuit above results in a new circuit in which the behavior of the 1.5 V battery is repeated as many times as needed to achieve the desired intensity of light: the new electrical circuit contains a cascade of 1.5 V batteries in series. Note that in model-based analogy, function is an abstraction of a design pattern; a function corresponds to the goal of the design strategy captured in the pattern.

Theories of analogies in general are of two types. The first kind of theory proposes general mechanisms for analogical mapping and transfer (e.g., Falkenhainer et al., 1989; Ho-

lyoak & Thagard, 1989; Thagard et al., 1990; Forbus, 1995). The second kind of theory of analogy describes contents of knowledge that drive the analogical mapping and transfer (e.g., Winston, 1979; Clement, 1988; Nersessian, 1999; Clement, 2008; Nersessian, 2008). Model-based analogy is a content theory of analogy. Although model-based analogy could potentially be implemented in some general mechanism, its power comes from the identification and representation of knowledge content driving the analogies (e.g., SBF models of design cases and BF models of design patterns).

An intelligent agent called Ideal implemented and evaluated the theory of model-based analogy for innovative conceptual design (Bhatta & Goel, 1994, 1996a, 1997). Ideal contained two libraries: a library of design cases and a library of design patterns. Given a new design problem, Ideal first operated like the Kritik agent, retrieving similar design cases and using their SBF models to tweak the design cases to address the new problem. However, if this tweaking of design cases was infeasible or insufficient, then Ideal retrieved the BF models of appropriate design patterns and instantiated them to modify the design case to address the given problem. Balazs and Brown (2002) developed a scheme for using SBF models to support analogical simplification of design problems.

Altshuller's (1984) TRIZ method often is viewed as a fundamental process of creative design in engineering. We conjectured that Ideal's method of model-based analogy might be a complementary process of creative design (Goel, 1997). Ideal learns BF models of design patterns from design cases such as the patterns of various kinds of feedback and feedforward. It uses these patterns to address cross-domain analogies in which a pattern learned in one domain is used to address a design problem in a different domain (Goel & Bhatta, 2004). As Ideal thus generates new designs, it encapsulates them as new design cases and stores them in its case library for potential reuse. It uses the SBF models of the design cases to learn the indices to the new cases (Bhatta & Goel, 1996b). This work resulted in two additional principles for functional modeling.

Principle 9: Functional modeling should cover not only complex systems but also design- and domain-independent abstract processes, patterns, and principles that systems embody.

Principle 10: Functional modeling should support learning tasks in addition to memory and problem-solving tasks. Learning tasks include learning of design cases and case indices, learning of abstract design patterns, and learning of abstraction hierarchies.

6. ENVIRONMENTALLY BOUND SBF (ESBF) MODELS OF SYSTEM-ENVIRONMENT INTERACTIONS

Chandrasekaran and Josephson (2000) describe device-centric and environment-centric views of functions of complex

systems. In their revised FR scheme, while function in the device-centric view is an abstraction of the internal causal behaviors of the system, in the environment-centric view, a function is an effect of the system on its external environment.

In parallel, Sattiraju Prabhakar developed ESBF models that capture an environment-centric view of complex systems (Prabhakar & Goel, 1996, 1998). Prabhakar noted that earlier notions of functions were limited to systems whose external environments play only a limited and passive role in the functioning of the system. For such systems, it was possible to abstract the system–environment interactions into **Given** and **Makes** states of the system and stimuli from the environment to the system. This is a system-centric view of function. However, the external environment plays an important and active role in the functioning of many systems. Consider, for example, an air-conditioning system for a room that can be mounted in a window. The function of the air conditioner is to lower the temperature in a room. Note that this function is the result of both internal causal behaviors of the air conditioner and behaviors in its external environment (e.g., heat sources and air currents in the room). This is an environment-centric view of function of a system: function as an abstraction of a state change in the system’s external environment. Thus, we extended SBF models into ESBF models (Prabhakar & Goel, 1996, 1998). This results in another principle:

Principle 11: Functional models should support multiple perspectives on a system, such as system-centric and environment-centric perspectives.

7. DSSBF MODELS OF DESIGN DRAWINGS

Let us consider the vector-graphics drawing illustrated in Figure 6a. Designers make drawings like this using computer-aided design (CAD) tools. If CAD programs could understand such drawings, then they could interactively aid the construction of such drawings, check if the design depicted in the

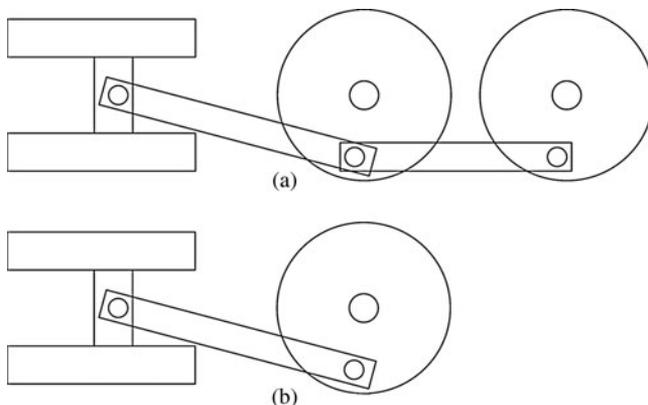


Fig. 6. (a) A target design drawing requiring interpretation and (b) a source drawing in a case library for which a structure–behavior–function model is known.

drawing is functionally and behaviorally correct, and acquire functional and behavioral knowledge from design drawings.

For his PhD dissertation at Georgia Tech, Yaner (2007) developed the technique of compositional analogy for understanding design drawings. An intelligent agent called Archytas implements and evaluates the method of compositional analogy (Yaner & Goel, 2007a, 2007b, 2008). Archytas constructs SBF models of the design depicted in a drawing such as the one in Figure 6a in analogy to known SBF models of similar designs stored in a case library. For example, if Archytas’s case library contains an SBF model of the design illustrated in Figure 6b, then it can transfer and adapt the SBF of the known design to construct an SBF model of the new design illustrated in Figure 6a. (Briefly, the function of the piston and the crankshaft system illustrated in Fig. 6b is to convert translational motion into rotational motion: as the piston moves linearly inside the cylinder depicted on the left of the figure, it generates angular motion in the crankshaft depicted on the right of the figure. The function of the new piston and crankshaft system illustrated in Fig. 6a is to generate rotational motion in two crankshafts.)

To support analogical transfer of the SBF model of a source drawing to a target drawing, Yaner extended SBF models to include drawings and shapes, resulting in DSSBF models. Figure 7 illustrates the DSSBF scheme. Note that just as behavior is an intermediate level of abstraction between structure and function, shape and spatial relations are an intermediate abstraction between drawings and structure, and further, structure now is an intermediate abstraction between shapes and behavior. Thus, DSSBF models form an abstraction hierarchy from drawings to function. Therefore, DSSBF models explicitly link functional knowledge to visuospatial knowledge. Anthony et al. (2001) describe another scheme for coupling SBF models and visuospatial knowledge of engineering systems.

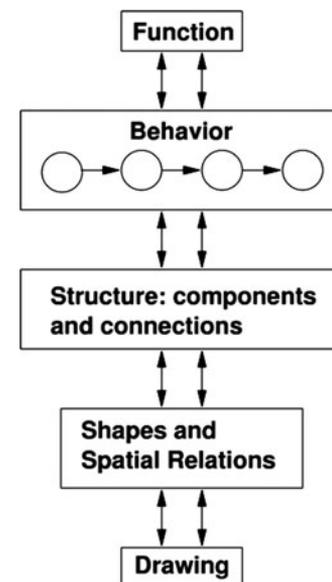


Fig. 7. The abstraction hierarchy of drawing–shape–structure–behavior–function models.

The method of compositional analogy first gathers individual lines, arcs, and intersection points in the target drawing into shapes and then finds mappings between the source drawing and the target drawing at this level of intersections. Then it groups these mappings and transfers shapes from the source drawing to the target drawing. Next, it finds a mapping between the source and target drawings at the level of the shapes and transfers the structural model from the source drawing to the target drawing. Thus, compositional analogy finds semantic labels for the components and connections in the unlabeled target drawing. Next, it finds a mapping between the source and target drawings at the structure level and transfers the behavioral model from the source to the target, and so on, all the way up to the level of function. Analogical mapping and transfer thus are iterative, working up to each level of the abstraction hierarchy of the DSSBF model, hence the name *compositional* analogy. This work leads to three more principles for functional modeling.

Principle 12: Functional reasoning should be coupled with visuospatial reasoning. Functional knowledge should be grounded in visuospatial knowledge.

Principle 13: Functional, behavioral, structural, and visuospatial knowledge form an abstraction hierarchy. Just as behaviors are an intermediate level of abstraction between structure and function, structure is an intermediate abstraction level between causal behaviors and spatial relations among shapes.

Principle 14: Functional models should support analogical mapping and transfer. In particular, mappings at one level of abstraction in the abstraction hierarchy of a functional model can support transfer at a higher abstraction level.

8. FORMALIZATION OF THE SBF MODELING LANGUAGE

One difficulty in comparing various FRs is that often they are specified only informally. My Georgia Tech colleague Spencer Rugaber used Backus–Naur form to formally specify the syntax and structural semantics of the SBF modeling language (Goel, 2009). Figure 8 illustrates a partial specification of function in the Backus–Naur notation.

This leads to another principle.

Principle 15: For clarity and precision, functional models should be specified in a formal language.

9. DISCUSSION AND CONCLUSIONS

In this article, we have tried to articulate the core principles of a specific cognitively oriented AI methodology for functional modeling by analyzing what remained stable in the evolution of SBF modeling over nearly 30 years, and what changed, how, and why. Table 1 summarizes the changes in the notion of functions of complex systems from SBF models (described in Section 4), to BF models (Section 5), to ESBF models (Section 6), to DSSBF models (Section 7).

There are two primary reasons for these changes. First, our notion of function changed from SBF models to ESBF models because of the expansion in the class of complex systems under consideration. Initially, we analyzed systems in which the external environment played only a limited and passive role. For such systems, system–environment interactions could be abstracted as states of the system and stimuli from the environment to the system. As we analyzed the functioning of systems in which behaviors in the environment played

```

FunctionModel := {Function}+

Function := INTEGER // Function

Name // Function name
[STRING] // Description

FunctionType
[INTEGER] // PrimaryArgument (ElementId)
INTEGER // Behavior Id
STRING // Provided
INTEGER // Prerequisite Stateld
{INTEGER}+ // Resultant Statelds
{INTEGER}* // StimulusIds

FunctionType := 'achievement' | 'maintenance' | 'prevention' | 'negation'
Primitive Functions of FunctionType Achievement := 'create' | 'destroy' | 'expel' | 'allow' | 'pump' | 'move' | 'change_mode'

```

Fig. 8. Specification of function in a structure–behavior–function (SBF) model.

Table 1. Summary of evolution in our notion of function

Name of Model	Primary References	Notions of Function
Structure–behavior–function model	Goel (1992), Goel & Stroulia (1996), Goel, Bhatta, & Stroulia (1997)	An intended output behavior of a system; an abstraction of an internal causal behavior
Behavior-Function Model	Bhatta & Goel (1994, 1996a, 1996b, 1997)	An abstraction of a design pattern; a design adaptation goal
Environmentally bound structure–behavior–function model	Prabhakar & Goel (1996, 1998)	An abstraction of a state change in the external environment of a system; an effect of the system on the environment
Drawing–shape–structure–behavior-function model	Yaner & Goel (2006, 2007a, 2007b)	An intended output behavior of a system; an abstraction of visuospatial knowledge through an abstraction hierarchy

a significant role, we developed an environment-centric view of system functioning, with system function as an abstraction of a state change in the environment.

Second, the other main reason for changes in our notion of function from SBF models to BF models to DSSBF models is the expansion in the range of tasks under consideration. Initially, we addressed the task of adaptive design (i.e., generation of conceptual designs by retrieving and adapting similar design cases from the same domain). To address this task, we used an ontology of components, substances, states, and functions, to expand the FR scheme into SBF models. Then, we addressed the task of analogical design (i.e., generation of conceptual designs by transferring design patterns across domains). To address this task, we abstracted BF models from SBF models to capture design- and domain-independent design patterns. Next, we addressed the task of understanding design drawings by constructing SBF models of the design depicted in a drawing. To address this task, we extended SBF models of systems into DSSBF models that coupled SBF models with visuospatial knowledge of the system.

9.1. Methodological principles

Table 2 summarizes the 15 principles for an AI methodology for functional modeling enumerated above. Although I have not mentioned it explicitly, a functional model of a system should be correct and accurate. Because this principle is common to all modeling methodologies, we may call it the *Zeroth Principle* of functional modeling.

We now briefly elaborate on six of these principles, which are six basic, core principles not always followed in functional modeling.

Principle 1: Functional models are constructed to support functional reasoning. Any theory of functional modeling needs to be situated in a theory of functional reasoning: the whys, whats and hows of functional reasoning are good starting points for answering the whys, whats and hows of functional models.

Functional models of complex systems and functional reasoning about the systems are closely intertwined. Functional

models, and meanings of functions, are constructed to support functional reasoning (and not the other way around). However, in the literature, functional models often are compared relative to various notions of function or classes of systems. Although this coverage of systems is important, the usefulness of a functional model, or of a meaning of *function*, for reasoning tasks is even more important. We hope that the analysis presented in this paper exposes this constraint clearly. Thus, as the task requirements on functional reasoning in our work evolved from adaptive design to analogical design, so did the FRs from SBF models of design cases to BF models of design patterns. Similarly, as the task requirements on functional reasoning evolved to understanding the functioning of systems depicted in design drawings, so did the FR from SBF models to DSSBF models. It follows that the functional models we have described here likely will con-

Table 2. Fifteen methodological principle for functional modeling

Issues	Principles
Basics of functional models	Principle 1 (functional reasoning)
	Principle 2 (functional decomposition)
	Principle 3 (functional explanation)
Basics of modeling methodology	Principle 6 (experimental evaluation)
	Principle 15 (formalization)
Abstractions	Principle 4 (behavior)
	Principle 9 (patterns)
	Principle 13 (abstraction hierarchy)
Modality of reasoning	Principle 11 (functional and visuospatial reasoning)
Ontology	Principle 8 (ontology of states and functions)
Knowledge and memory organization	Principle 9 (ontology of patterns)
	Principle 5 (functions as indices to behaviors)
Analogy and learning	Principle 7 (functions as indices to design cases)
	Principle 10 (learning cases, indices, patterns, abstractions)
	Principle 14 (analogical mapping and transfer in an abstraction hierarchy)
Perspectives	Principle 11 (system and environment centric views)

tinue to evolve as we address increasingly complex tasks entailing functional reasoning.

Principle 6: Theories of functional modeling need to be evaluated through experimentation. Construction of knowledge systems for automated reasoning is one experimental method for testing, refining, and revising functional models.

Theories of functional modeling should be empirically grounded. In particular, theories of functional modeling must be experimentally evaluated. Empirical studies can cover a wide spectrum, ranging from immersive cultural studies to controlled psychological experiments to neuroimaging to interactive CAD tools to automated reasoning in knowledge systems. Irrespective of the experimental method chosen, it is critical to evaluate theories of functional modeling. Nevertheless, many functional models in the literature remain empirically unevaluated.

Principle 7: Functional modeling should support memory tasks in addition to problem-solving and decision-making tasks. Memory tasks include case indexing, retrieval, and storage.

Reasoning about complex systems often is characterized in terms of problem solving, decision making, constraint satisfaction, and so on. Conceptual design, for example, is often characterized as a kind of problem solving or a kind of decision making. From a cognitive perspective, these views appear limited as well as limiting: design also engages memory and learning, for example. When we view design as decision making, we tend to develop representations that support the decision making and not necessarily memory. In our work, functional models play multiple roles in memory organization. For example, in adaptive design (the Kritik system) and analogical design (the Ideal system), functions acted as indices to design cases stored in a case library so that the right cases could be retrieved to achieve a desired function. Further, in analogical design, functions acted as indices to design patterns so that the right design pattern could be retrieved for analogical transfer.

Principle 10: Functional modeling should support learning tasks in addition to memory and problem-solving tasks. Learning tasks include learning of design cases and case indices, learning of abstract design patterns, and learning of abstraction hierarchies.

Our argument for use of functional models for learning tasks is similar to the one for their use in memory tasks. While the literature on functional models for supporting problem solving and decision making is very rich, the literature on functional models for supporting learning is quite sparse. Nevertheless, each episode of conceptual design also engages learning, both during and after the episode. In our work, in

both within-domain case-based design and cross-domain analogical design, functional models of design cases enable the learning of indices to the cases for storing them in memory. Further, in our work on cross-domain analogical design, functional models of design cases enable the learning of design patterns. Like memory tasks, learning tasks impose important constraints on functional modeling.

Principle 12: Functional reasoning should be coupled with visuospatial reasoning. Functional knowledge should be grounded in visuospatial knowledge.

Reasoning about complex systems typically engages not only functional and causal reasoning but also visuospatial reasoning (e.g., Ferguson, 1992). For example, visuospatial reasoning often plays an early and important role in adaptive design as well as redesign, yet the literature on functional reasoning in general does not directly connect functional models with visuospatial knowledge. On the contrary, knowledge of functions in most functional models is viewed as amodal (i.e., separated from any perceptual modality). In our work on DSSBF models, functional knowledge is grounded in visuospatial knowledge and functional reasoning is coupled with visuospatial reasoning.

Principle 15: For clarity and precision, functional models should be specified in a formal language.

Comparison and evaluation of functional models requires precise specification of the models. Symbolic logic, graph theory, and set theory are among many mathematical formalisms available for specifying functional models, yet many functional models are specified only informally in the literature.

9.2. Implications of the AI methodology

We posit two sets of implications of the above cognitively oriented AI methodology for functional modeling. First, it helps clarify the relationship between our SBF models and functional models developed using other methodologies. Let us consider, for example, Pahl and Beitz's (1996) functional models of technological systems. We see many deep commonalities and broad similarities between Pahl and Beitz's functional models and our SBF models, for example, explicit representation of functions, functional decomposition, and flow of substances in SBF models and flow of material, energy, and information in Pahl and Beitz's models. Of course, there are also some differences that arise in part due to different goals and methodologies. Pahl and Beitz's theory of systematic design is prescriptive, and their functional models are intended to support the prescribed design methods. In contrast, our theories of case-based design and analogical design are descriptive. Further, Pahl and Beitz's functional models, like the technological systems themselves, are external to the human mind. In contrast, in our research, the SBF

models typically are inside the “mind” of intelligent agent and, at least hypothetically, inside the human mind as well: that is, the SBF models are also hypotheses about mental models.

These differences between the goals and methodologies lead to different modeling constraints and affordances. For example, one of the key constraints in Pahl and Beitz’s (1996) methodology is accuracy and scalability of their functional models. Thus, they emphasize the importance of system ontology for constructing high-fidelity external representations of large-scale systems. Further, they tend to evaluate a model’s representational vocabulary by its coverage of different classes of technological systems. In contrast, one of the key constraints in our work is explanation and tractability of functional reasoning. Thus, our work on functional models emphasizes the importance of knowledge abstraction (e.g., BF models of design patterns), organization (e.g., DSSBF models), and localization of reasoning (e.g., functions acting as indices to behaviors responsible for accomplishing the functions and annotations on transitions acting as indices to functions of subsystems). Furthermore, it tends to evaluate functional reasoning by its coverage of functional reasoning, including problem-solving, memory, and learning tasks. Although we made the above comparisons in the context of Pahl and Beitz’s functional models, they are equally applicable to other functional modeling schemes that share Pahl and Beitz’s goals and methodologies, for example, Hubka and Eder (1988).

We should state explicitly and specifically that we are *not* arguing that our methodology for functional modeling is “right” or that our functional models are “better” than those developed using other methodologies. Instead, we view the two sets of goals and methodologies as *complementary* to each other. Chandrasekaran (2005) suggests one kind of complementarity between them: Pahl and Beitz’s (1996) functional models, for example, may provide domain-specific libraries of technological functions and structures, and our methods of functional reasoning may provide computational methods for automated reasoning with the libraries. We conjecture that construction of interactive tools for aiding humans in complex tasks such as conceptual design will require *both* high-fidelity modeling of large-scale complex systems and a deep understanding of human reasoning about functioning of complex systems. A challenging question is how to translate these complementarities into interactive tools for supporting design practice. This is a central issue in our current work on biologically inspired design (Goel et al., 2012).

The second set of implications of our AI methodology for functional modeling pertains to various meanings of function. We begin by noting that the evolution of SBF modeling illustrates three different *kinds* of meaning of function. First, *the meaning of function may lie in the objects, relations, and processes in the real world to which the function refers* (i.e., for which the function stands in the functional model). For example, the representation of function in the SBF model of the gy-

roscope follow-up illustrated in Figure 3a refers to, and stands for, specific objects, relations, and states in the actual gyroscope follow-up. This meaning of function is the meaning commonly used in functional modeling, including in the companion targets articles of Vermaas and Eckert. Second, *the meaning of function may lie in the invocation of related concepts*. For example, the representation of function in the SBF model of the gyroscope follow-up illustrated in Figure 3a indexes and invokes in memory the behavior that accomplishes the function. Similarly, the functional annotations on the behavior illustrated in Figure 3b index and invoke in memory specific functions of specific components of the gyroscope follow-up. Third, *the meaning of function may lie in the inferences it enables*. For example, given a specific design scenario of finely regulating the angular momentum of the gyroscope follow-up, the representation of the goal of the design pattern of cascading in Figure 4a enables inferences about whether or not the pattern is applicable to the scenario. Similarly, the representation of the design strategy in Figure 4b enables inferences about the application of the strategy to the given design scenario.

As Table 1 indicates, our work on SBF modeling subscribes not only to all three kinds of meanings of functions but also to several different meanings of function of the three kinds. Although we arrived at the different meanings at different points in the evolution of SBF modeling, the latter meanings augmented the earlier meanings instead of replacing them. We see no contradiction in the coexistence of different meanings of function. According to our Principle 1 of functional modeling, functional models, and meanings of functions, are constructed to support functional reasoning. Functional reasoning, whether in human cognition or in intelligent agents, is complex and makes many demands on problem solving and decision making, attention and perception, memory and learning. As and when feasible, functional reasoning makes opportunistic use of the different meanings of function in the available knowledge; as and when needed, it strategically re-represents knowledge and constructs new meanings of function. Thus, we see different meanings of function not as an obstacle to functional modeling but as a critical source of the power of functional reasoning.

ACKNOWLEDGMENTS

I am grateful to B. Chandrasekaran, my PhD adviser at Ohio State University in the 1980s and a mentor since then. My research on SBF modeling over the years has been supported by grants from several US agencies, including the National Science Foundation, the Defense Advanced Research Projects Agency, the Office of Naval Research, and the Institute of Education Sciences. My current work on SBF modeling is supported by the National Science Foundation (CreativeIT Grant 0855916). I am grateful to Pieter Vermaas and Claudia Eckert for inviting me to write this article. I thank David C. Brown, Spencer Rugaber, an anonymous reviewer, and Pieter and Claudia for their constructive critiques of earlier drafts that helped to significantly improve this article.

REFERENCES

- Alexander, C., Ishikawa, S., & Silverstein, M. (1977). *A Pattern Language—Towns, Buildings, Construction*. New York: Oxford University Press.
- Altshuller, G. (1984). *Creativity as an Exact Science: The Theory of the Solution of Inventive Problems*. (Williams, A., Trans.). New York: Gordon & Breach.
- Andreasen, M. (1991). Design methodology. *Journal of Engineering Design* 2(4), 321–335.
- Anthony, L., Regli, W., John, J., & Lombeyda, S. (2001). An approach to capturing structure, behavior and function of artifacts in CAD. *ASME Journal of Computing and Information Science in Engineering* 1(2), 186–192.
- Balazs, M., & Brown, D. (2002). Design simplification by analogical reasoning. In *From Knowledge Intensive CAD to Knowledge Intensive Engineering* (Cugini, U., & Wozny, M., Eds.), pp. 29–44. Amsterdam: Kluwer Academic.
- Bechtel, W., & Richardson, R. (2010). *Discovering Complexity* (2nd ed.). Cambridge, MA: MIT Press.
- Bhatta, S. (1995) *Model-based analogy in innovative device design*. PhD Thesis. Georgia Institute of Technology.
- Bhatta, S., & Goel, A. (1994). Model-based discovery of physical principles from design experiences. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 8(2), 113–123.
- Bhatta, S., & Goel, A. (1996a) From design cases to generic mechanisms. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 10, 131–136.
- Bhatta, S., & Goel, A. (1996b). Model-based indexing and index learning in engineering design. *Engineering Applications of AI* 9(6), 601–610.
- Bhatta, S., & Goel, A. (1997) Learning generic mechanisms for innovative strategies in adaptive design. *Journal of the Learning Sciences* 6(4), 367–396.
- Bracewell, R., & Sharpe, J. (1996). Functional descriptions used in computer support for qualitative scheme generation: Schemebuilder. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 10, 333–346.
- Bylander, T. (1991). A theory of consolidation for reasoning about devices. *Man-Machine Studies* 35, 467–489.
- Bylander, T., & Chandrasekaran, B. (1985). Understanding behavior using consolidation. *Proc. Int. Joint Conf. AI*, Vol. 1, pp. 450–454. San Francisco, CA: Morgan Kaufmann.
- Carrara, M., Garbacz, P., & Vermaas, P. (2011). If engineering function is a family resemblance concept: assessing three formalization strategies. *Applied Ontology* 6, 141–163.
- Chakrabarti, A., & Blich, T. (2001). A scheme for functional reasoning in design. *Design Studies* 22(6), 493–517.
- Chandrasekaran, B. (1994a). Functional representation: a brief historical perspective. *Applied Artificial Intelligence* 8(2), 173–197.
- Chandrasekaran, B. (1994b). Functional representations and causal processes. In *Advances in Computers* (Yovits, M., Ed.), pp. 73–143. San Diego, CA: Academic Press.
- Chandrasekaran, B. (2005). Representing function: relating functional representation and functional modeling research streams. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 19, 65–74.
- Chandrasekaran, B., Goel, A., & Iwasaki, Y. (1993). Functional representation as design rationale. *IEEE Computer* 26, 48–56.
- Chandrasekaran, B., & Josephson, J. (2000). Function in device representation. *Engineering With Computers* 16, 162–177.
- Chittaro, L., Guida, G., Tasso, C., & Toppano, E. (1993). Functional and teleological knowledge in the multimodeling approach for reasoning about physical systems: a case study in diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics* 23(6), 1718–1751.
- Clement, J. (1988). Observed methods for generating analogies in scientific problem solving. *Cognitive Science* 12, 563–586.
- Clement, J. (2008). *Creative Model Construction in Scientists and Students: The Role of Imagery, Analogy, and Mental Simulation*. Dordrecht: Springer.
- Darden, L. (1998). Anomaly-driven theory redesign: computational philosophy of science experiments. In *The Digital Phoenix: How Computers Are Changing Philosophy* (Bynum, T., & Moor, J., Eds.), pp. 62–78. New York: Blackwell.
- Eckert, C. (2013). That which is not form: the practical challenges in using functional concepts in design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 27(3), 217–232 [this issue].
- Eckert, C., Alink, T., Ruckpaul, A., & Albers, A. (2011). Different notions of function: results from an experiment on the analysis of an existing product. *Journal of Engineering Design* 22, 811–832.
- Erden, M., Komoto, H., van Beek, T., D'Amelio, V., Echavarria, E., & Tomiyama, T. (2008). A review of function modeling: approaches and applications. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 22(2), 147–169.
- Falkenhainer, B., Forbus, K., & Gentner, D. (1989). The structure-mapping engine: algorithm and examples. *Artificial Intelligence* 41, 1–63.
- Ferguson, E. (1992). *Engineering and the Mind's Eye*. Cambridge, MA: MIT Press.
- Forbus, K., Gentner, D., & Law, K. (1995). MAC/FAC: a model of similarity-based retrieval. *Cognitive Science* 19, 141–205.
- Forrester, J. (1994). *Learning through system dynamics as preparation for the 21st century*. Keynote address presented at the 1994 Systems Thinking and Dynamic Modeling Conference for K–12 Education, Concord, MA. Accessed at <http://clexchange.org/curriculum/>
- Freeman, P., & Newell, A. (1971). A model for functional reasoning in design. *Proc. 2nd Int. Joint Conf. Artificial Intelligence*, pp. 621–633, London, September 1–3.
- Garbacz, P., Borgo, S., Carrara, M., & Vermaas, P. (2011). Two ontology-driven formalizations of function and their comparison. *Journal of Engineering Design* 22, 733–764.
- Gero, J. (1990). Design prototypes: a knowledge representation schema for design. *AI Magazine* 11(4), 26.
- Gero, J., & Kannengiesser, U. (2004). The situated function–behavior–structure framework. *Design Studies* 25, 373–391.
- Gero, J., Tham, K., & Lee, H. (1992). Behavior: a link between function and structure in design. In *Intelligent Computer Aided Design* (Brown, D., Waldron, M., & Yoshikawa, H., Eds.), pp. 193–225. Amsterdam: North-Holland.
- Goel, A. (1989). *Integration of case-based reasoning and model-based reasoning for adaptive design problem solving*. PhD Thesis. Ohio State University.
- Goel, A. (1992). Representation of design functions in experience-based design. In *Intelligent Computer Aided Design* (Brown, D., Waldron, M., & Yoshikawa, H., Eds.), pp. 283–308. Amsterdam: North-Holland.
- Goel, A. (1997). Design, analogy, and creativity. *IEEE Expert* 12(3), 62–70.
- Goel, A., & Bhatta, S. (2004). Use of design patterns in analogy-based design. *Advanced Engineering Informatics* 18, 85–94.
- Goel, A., Bhatta, S., & Stroulia, E. (1997). Kritik: an early case-based design system. In *Issues and Applications of Case-Based Reasoning in Design* (Maher, M., & Pu, P., Eds.), pp. 87–132. Mahwah, NJ: Erlbaum.
- Goel, A., & Chandrasekaran, B. (1988). Integrating model-based reasoning with case-based reasoning for design problem solving. In *Proc. AAAI-88 Workshop on AI and Design*. St. Paul, MN, August 21–26, 1988.
- Goel, A., & Chandrasekaran, B. (1989). Functional representation of design and redesign problem solving. *Proc. 11th Int. Joint Conf. AI (IJCAI-89)*, pp. 1388–1394. San Francisco, CA: Morgan Kaufmann.
- Goel, A., & Chandrasekaran, B. (1992). Case-based design: a task analysis. In *Artificial Intelligence Approaches to Engineering Design: Volume II. Innovative Design* (Tong, C., & Sriram, D., Eds.), pp. 165–184. San Diego, CA: Academic Press.
- Goel, A., & Davies, J. (2011). Artificial intelligence. In *Cambridge Handbook of Intelligence* (Sternberg, R., & Kauffman, S., Eds.). Cambridge: Cambridge University Press.
- Goel, A., Rugaber, S., & Vattam, S. (2009). Structure, behavior & function of complex systems: The structure–behavior–function modeling language. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 23, 23–35.
- Goel, A., & Stroulia, E. (1996). Functional device models and model-based diagnosis in adaptive design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 10, 355–370.
- Goel, A., Vattam, S., Wiltgen, B., & Helms, M. (2012). Cognitive, collaborative, conceptual and creative—four characteristics of the next generation of knowledge-based CAD systems: a study in biologically inspired design. *Computer-Aided Design* 44(10), 879–900.
- 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(2), 65–82.
- Holyoak, K.J., & Thagard, P. (1989). Analogical mapping by constraint satisfaction. *Cognitive Science* 13(3), 295–335.
- Hubka, V., & Eder, W. (1988). *Theory of Technical Systems*. Berlin: Springer-Verlag.

- Keuneke, A. (1991). Device representation: the significance of functional knowledge. *IEEE Expert* 6(2), 22–25.
- Kitamura, Y., Kashiwase, M., Fuse, M., & Mizoguchi, R. (2004). Deployment of an ontological framework for functional design knowledge. *Advanced Engineering Informatics* 18(2), 115–127.
- Kitamura, Y., Sano, T., Namba, K., & Mizoguchi, R. (2002). A functional concept ontology and its application to automatic recognition of functional structures. *Advanced Engineering Informatics* 16(2), 145–163.
- Kolodner, J. (1993). *Case-Based Reasoning*. San Francisco, CA: Morgan Kaufmann.
- Langley, P. (2012). The cognitive systems paradigm. *Advances in Cognitive Systems* 1, 3–13.
- Machamer, P., Darden, L., & Craver, C. (2000). Thinking about mechanisms. *Philosophy of Science* 67, 1–25.
- Murdock, J., Szykman, S., & Sriram, R. (1997). An information modeling framework to support design databases and repositories. *Proc. 1997 ASME DET Conference on Design Theory and Methods*, Paper No. DETC97/DFM4373, Sacramento, CA.
- Nersessian, N. (1999). Model-based reasoning in conceptual change. In *Model-Based Reasoning in Scientific Discovery* (Magnani, L., Nersessian, N., & Thagard, P., Eds.). New York: Kluwer Academic/Plenum Press.
- Nersessian, N. (2008). *Creating Scientific Concepts*. Cambridge, MA: MIT Press.
- Pahl, G., & Beitz, W. (1996). *Engineering Design: A Systematic Approach* (Wallace, K., Blessing, L., & Bauert, F., Trans., Wallace, K., Ed., 2nd ed.). New York: Springer-Verlag.
- Prabhakar, S., & Goel, A. (1996). Learning about novel operating environments: designing by adaptive modeling. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 10, 151–156.
- Prabhakar, S., & Goel, A. (1998). Functional modeling for enabling adaptive design of devices for new environments. *Artificial Intelligence in Engineering* 12, 417–444.
- Rasmussen, J. (1985). The role of hierarchical knowledge representation in decision making and system management. *IEEE Transactions on Systems, Man, and Cybernetics* 15(2), 234–243.
- Rieger, C., & Grinberg, M. (1978). A system of cause–effect representation and simulation for computer-aided design. In *Artificial Intelligence and Pattern Recognition in Computer-Aided Design* (Latombe, J., Ed.), pp. 299–333. Amsterdam: North-Holland.
- Riesbeck, C., & Schank, R. (1989). *Inside Case-Based Reasoning*. Mahwah, NJ: Erlbaum.
- Sasajima, M., Kitamura, Y., Mitsuru, I., & Mizoguchi, R. (1996). A representation language for behavior and function: FBRL. *Expert Systems with Applications* 10(3/4), 471–479.
- Sembugamoorthy, V., & Chandrasekaran, B. (1986). Functional representation of devices and compilation of diagnostic problem-solving systems. In *Experience, Memory, and Learning* (Kolodner, J., & C. Riesbeck, C., Eds.), pp. 47–73. Mahwah, NJ: Erlbaum.
- Simon, H. (1962). The architecture of complexity. *Proceedings of the American Philosophical Society* 106(6), 467–482.
- Simon, H. (1996). *Sciences of the Artificial* (3rd ed.). Cambridge, MA: MIT Press.
- Simon, H. (1999). Can there be a science of complex systems? In *Unifying Themes in Complex Systems* (Bar-Yam, Y., Ed.), pp. 3–14. Cambridge, MA: Perseus.
- Sticklen, J., & Chandrasekaran, B. (1989). Integrating classification-based compiled level reasoning with function-based deep level reasoning. *Applied Artificial Intelligence* 3(2–3), 275–304.
- Stone, R., & Wood, K. (2000). Development of a functional basis for design. *ASME Journal of Mechanical Design* 122(4), 359–370.
- Sycara, K., Guttal, R., Koning, J., Narasimhan, S., & Navinchandra, D. (1991). CADET: a case-based synthesis tool for engineering design. *Expert Systems* 4(2), 157–188.
- Szykman, S., Sriram, R., Bochenek, C., Racz, J., & Senfaute, J. (2000). Design repositories: engineering design's new knowledge base. *IEEE Intelligent Systems* 15(3), 48–55.
- Thagard, P., Holyoak, K.J., Nelson, G., & Gochfeld, D. (1990). Analog retrieval by constraint satisfaction. *Artificial Intelligence* 46, 259–310.
- Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y., & Tomiyama, T. (1996). Supporting conceptual design based on the function–behavior–state modeler. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 10, 275–288.
- Umeda, Y., Takeda, H., Tomiyama, T., & Yoshikawa, H. (1990). Function, behavior, and structure. *AIENG '90 Applications of AI in Engineering*, pp. 177–193. Southampton: Computational Mechanics.
- Umeda, Y., & Tomiyama, T. (1997). Functional reasoning in design. *IEEE Expert* 12(2), 42–48.
- Vasandani, V., & Govindaraj, T. (1995). Knowledge organization in intelligent tutoring systems for diagnostic problem solving in complex dynamic domains. *IEEE Transactions on Systems, Man and Cybernetics* 25(7), 1076–1096.
- Vermaas, P. (2013). The coexistence of engineering meanings of function: four responses and their methodological implications. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 27(3), 191–202 [this issue].
- Winston, P. (1979). Learning & reasoning by analogy. *Communications of ACM* 23(12), 689–703.
- Yaner, P. (2007). *From shape to function: acquisition of teleological models from design drawings by compositional analogy*. PhD Thesis. College of Computing, Georgia Institute of Technology.
- Yaner, P., & Goel, A. (2007a). Understanding drawings by compositional analogy. *Proc. 20th Int. Joint Conf. Artificial Intelligence (IJCAI-2007)*, pp. 1131–1137, Hyderabad, India.
- Yaner, P., & Goel, A. (2007b). Visual analogies at multiple levels of abstraction. *Proc. 29th Annual Meeting of the Cognitive Science Society*, pp. 1671–1676, Nashville, TN, August 1–4.
- Yaner, P., & Goel, A. (2008). Analogical recognition of shape and structure in design drawings. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 22(2), 117–128.

Ashok K. Goel is a Professor of computer and cognitive science in the School of Interactive Computing at Georgia Institute of Technology. He is Director of the school's Design & Intelligence Laboratory and Codirector of the Institute's Center for Biologically Inspired Design. Dr. Goel serves on the boards of the Biomimicry 3.8 Institute and *BioInspired!* digital magazine. His 2012 TEDx talk (<http://www.youtube.com/watch?v=wiRDQ4hr9i8>) summarizes some of his research on biologically inspired design. An interactive tool for supporting some aspects of biologically inspired design developed by his laboratory is available at <http://dilab.cc.gatech.edu/dane/>