
Representing function: Relating functional representation and functional modeling research streams

B. CHANDRASEKARAN

Department of Computer Science and Engineering, 395 Dreese Laboratories, 2015 Neill Avenue, Ohio State University, Columbus, Ohio 43210-1277, USA

(RECEIVED August 7, 2004; ACCEPTED January 4, 2005)

Abstract

This paper is an informal description of some recent insights about what a device function is, how it arises in response to needs, and how function arises from the structure of a device and the functions of its components. These results formalize and clarify a set of contending intuitions about function that researchers have had. The paper relates the approaches, results, and goals of this stream of research, called functional representation (FR), with the functional modeling (FM) stream in engineering. Despite the occurrence of the term function in the two streams, often the results and techniques in the two streams appear not to have much to do with each other. I argue that, in fact, the two streams are performing research that is mutually complementary. FR research provides the basic layer for device ontology in a formal framework that helps to clarify the meanings of terms such as function and structure, and also to support representation of device knowledge for automated reasoning. FM research provides another layer in device ontology, by attempting to identify behavior primitives that are applicable to subsets of devices, with the hope that functions can be described in those domains with an economy of terms. This can lead to useful catalogs of functions and devices in specific areas of engineering. With increased attention to formalization, the work in FM can provide domain-specific terms for FR research in knowledge representation and automated reasoning.

Keywords: Device Ontology; Device Structure; Functional Modeling; Functional Representation; How Things Work

1. INTRODUCTION

Along with a number of colleagues, I have been working for two decades on a set of related issues about how to represent knowledge about devices so that automated systems can reason about the devices. This work is generally under the rubric of “artificial intelligence” (AI) or “AI in engineering,” because of its emphasis on knowledge representation, inference, and problem solving. A particular concern for my group has been the notion of device *function*, and a body of work of significant size and scope, generally dubbed functional representation (FR) research, has been created in representation of function and related notions of *structure* and *causal process*, and application of these representations for diagnosis and design. Sembugamoorthy and

Chandrasekaran (1986) proposed an account of how to represent an agent’s understanding of how a device works, that is, how a device’s functions arise from its structure and the functions of its components. From the representation, a diagnostic system for the device so represented could be automatically compiled. The work then expanded in many directions: simulation of the device from its FR, characterization of different types of functions, using functional indexing to choose devices with similar functions and modifying them for the current requirements, and so forth. This body of work, to which many people contributed, was reviewed in (Chandrasekaran, 1994a, 1994b). The concept of function that was implicit in this body of research was still informal, and additional formalization was required to clarify the concept. My work on developing a satisfactory concept of function culminated in Chandrasekaran and Josephson (2000), in which a formal framework is presented within which the meanings associated with the term function are explored. A major goal of the present paper is to give an informal account of the definition of the two

Reprint requests to: B. Chandrasekaran, Department of Computer Science and Engineering, 395 Dreese Laboratories, 2015 Neill Avenue, Ohio State University, Columbus, OH 43210-1277, USA. E-mail: Chandra@cse.ohio-state.edu

different senses of function that is developed in Chandrasekaran and Josephson (2000).

In the engineering community, with a heavy representation from systems, mechanical, and process engineers, there has been a parallel effort that is often called functional modeling (FM). As I understand this research, the emphasis has been on proposing *functional primitives*, a small set of terms that are posited to be fundamental in some way and out of which more complex functions can be described. The work has generally been done in mechanical and process engineering disciplines.

Despite the occurrence of the word “function,” and related words, such as “behavior” and “structure,” in both streams, it is not clear at first glance how the two streams of research relate to each other. The specific issues addressed, the techniques, and the methods of argumentation seem different. Even when they demonstrate their techniques by applications in mechanical or process engineering, the theoretical emphasis of the FR work is not on anything specific to these domains. That is, at the theoretical level, the work so far in the FR stream does not claim to be saying anything about any specific domain of engineering, but to offer a representational framework in which concepts such as function may be clarified in a domain-independent way. Domain-specific knowledge, for example, knowledge about the structure and function of an electronic buzzer, can indeed be represented, but the FR theory does not make available any primitives in that domain.

In contrast, the work in FM in engineering wrestles with such concepts as *energy, force, momentum, move, turn, convert, store*, and so forth. FM research offers proposals for basic vocabularies in which to represent functions along with arguments for why one proposal may be better than another. These terms are what are called *content* terms or elements of *ontology* in AI. Although the practitioners in the FM stream might demur, I think that it is best to view the proposals as domain specific in the sense that the primitives are motivated by phenomena in specific branches of engineering. Many of the researchers might like to think that their proposals cover all conceivable devices, including abstract devices such as battle plans and computer programs. Indeed, some of the content terms proposed in these theories might apply, but it is not necessary for these proposals to be universal for them to be useful and interesting. The important point is not whether the proposals are universal or not, but that they offer what are often called *content theories* for behaviors.

Are these two streams of research just two ships that pass each other in the dead of night, except for a *pro forma* ahoy in Special Issues like this one? George Bernard Shaw said that the United States and Britain were two nations united by an ocean and divided by a language. Are these two streams of research similarly divided by similar-sounding terminology, and if so what is the ocean that unites them?

A secondary goal in this paper is to offer some thoughts on how these two streams relate to each other. In the shorter

term, the somewhat different visions of how to use the results of the research drive somewhat different research agendas. Each group has much work to do in any case, and coming together may take awhile. However, there are good reasons for each group to know what the other is doing, because that can improve the work in each of the areas. In particular, the primitives being developed in the FM stream can be used to develop domain-specific versions of the FR language; and the FR language, so enhanced, may be used in engineering for a larger variety of applications for which the FM work is currently being proposed.

A caveat is in order. What follows is intended to be neither a review of the related literature nor a tutorial of the relevant ideas. It is more in the line of a pointer to a way of understanding certain commonalities and differences. My attempt to bridge the gap may falter, or utterly fail, for some readers because of the grounding in logic formalism required for understanding the rather brisk summary of the ideas I provide below. Reading Chandrasekaran and Josephson (2000) is necessary for a reasonably complete understanding of the issues.

2. THE FR STREAM

There is a large body of work in AI on representation of causal systems and devices in general, and on representing and reasoning with device functions in particular. This section focuses on one slice of the work on function, specifically on our attempts to clarify the meanings of the term function as used in engineering. Doing this requires a formal representational framework for modeling causal objects and the structural and causal relations between them.

Why do we need formal precision? Many people, within engineering and outside, have had good intuitions about the meanings of various terms such as function, and statements of them give a general idea about what is meant, but it is hard to be sure. Consider some proposals:

The word function is . . . a description of the action or effect required by a design problem, or that supplied by a solution. (Chakrabarti & Bligh, 2001)

Function of a mechanical object is dependent on [and derived from modeling and simulation of] the way that motion and forces are transmitted through the contacts between parts. (Faltings, 1990)

Function is a source of knowledge that abstracts behavior. Function of a component can be defined as operational, i.e., a relation between the input and output in the component; or purposive, i.e., a relation between the goal of a human user and the behavior of the component. (Chit-taro & Kumar, 1998)

. . . function of an object is distinct from its behaviour in that it is intentional rather than actual or expected, and proposes that there are two related but distinct views of

function . . . In one view, it is at the same level of abstraction as behaviour (intended behaviour), while in the other it is at a higher level. (purpose). (Chakrabarti, 1998)

. . . function can be semantically classified into two types: purpose function and action function. Purpose function is a *description of the designer's intention or the purpose of a design*. It is thus abstract and subjective. It is teleological knowledge and is human oriented. Action function is *an abstraction of intended and useful behavior that an artifact exhibits*. (Deng, 2002)

The first point to note is that intelligent and reasonable things are being said in each of them. However, although I can guess at what the authors mean by the various terms in the definitions, I am not quite sure. In addition, I am not sure if what one author means by behavior or action is exactly the same as meant by another author. Is the sense of the word "action" in the first of the quotes above, "Action required by the design problem" the same as the sense in "action function" in a latter quote? How does the proposal, "Function is a source of knowledge that abstracts behavior" relate to, say, "Function is action or effect required by the design problem"? Because the basic terms are somewhat unclear, the definitions amplify the uncertainty.

My own earlier work is not exempt from the above problems. One way of interpreting the early work of my group, the work that launched the FR stream in AI (Sembugamoorthy & Chandrasekaran, 1986), is that it viewed function as what the device does in terms of input–output transformation, but all expressed in terms of the device. For example, the function of a buzzer was stated to be that "when the button is pressed, the clapper arm should hit the clapper and make noise." *How* it did it was then said to be the causal processes that take place in the buzzer to go from the button press to clapper arm hitting the clapper repeatedly. One might at first blush think of this distinction as the same as the distinctions in the various quotations above. For example, one might say that the user wants (desires) a device that, when its button is pressed, will cause something in the device to hit a bell, thus identifying the input–output characterization with the purposive sense of function. How this purposive function is achieved can then be associated with the causal process (the behavior) sense of function, in the above definitions. So it looks like there is a neat correspondence between the function–behavior distinction in our 1986 paper and the various purpose–behavior distinctions in some of the quotes above. But is it really the case? Maybe there is yet another distinction, one between *why* one wants a device, *what* he wants it do, and *how* he wants it to do it. How to sort it all out? This was my mindset, somewhere in 1994, when I decided that we were going to be talking past each other about these ideas unless we found a way of being *very* precise about the various concepts.

In this search for a framework to help us talk precisely about the various concepts, we still need to start somewhere with some terms that themselves would be undefined,

but would be much clearer than the concepts that are problematic.

As it happens, an approach to compositional device modeling (Falkenhainer & Forbus, 1991) had been proposed, and a representation language called Compositional Modeling Language (CML) evolved from it at Stanford Knowledge Systems Laboratory. We adapted this framework for our purposes. The central intuition that it formalizes is that there are *objects* that are composed into objects whose parts, or components, the original objects become. These objects are themselves modeled as a pair: a set of *variables* and *causal constraints* between them. This is the same ontology as much of physics, although this does leave out things such as *fields*, which may also be a bearer of causality. But we choose simplicity over completeness at this stage if we can capture the concepts in the simpler framework. The framework can be later extended.

The variables in the models may be static, as in the model of an arch, or they may be *state variables*, as in objects where causes at one instant of time create effects at another instant of time, and the variable values change over time. CML also allows different types of interobject relations, such as electrical connections between terminals and rigid flexible connection between joints, to be modeled as a set of causal constraints between the variables of the individual objects in the relation. Given a set of objects along with their causal models, and the models for their interconnections, the causal model of the composed object can be generated automatically by the simulation system associated with CML.

A few, possibly obvious, points regarding models might be of interest. The same physical object can have different models for different purposes, and there may be abstraction relations between two models for the same object. For example, one model of an electronic calculator may be in terms of voltages and currents, another model may be in terms of numeral representations, and abstraction relations may be defined that translate between numeral variables and say voltage variables. The abstraction relations also induce a mapping between the two sets of causal constraints, for example, if one knew the causal constraints between the electrical variables at the circuit level, and the abstraction relations between the electrical level and the numeral representation level, it will be possible to obtain a set of causal constraints between the numeral representations. In Chandrasekaran and Josephson (2000) we develop the framework and discuss various modeling issues in some detail as the basis for explicating a variety of concepts including structure, behavior, and function. For example, the structure can be precisely modeled in terms of objects in certain causal relations. The composed object may be remodeled in terms of variables in a new abstraction level, and relations between composed object-level variables and component–object level variables can be expressed precisely.

In this article, I focus solely on the concept of function to show how it permits a variety of different senses to be captured precisely.

2.1. Function as effect

Let W be a world, modeled in terms of variables and causal constraints as just mentioned. Let us say there is a human user who has a need, say N , defined as a (desired) constraint between the variables of W . That is, the user wishes that the values of certain variables in W satisfy a constraint N . (The human user may or may not be part of W .)

For example, N_1 might be a need of a user described as “to know what is written on a sheet of paper” (in this case the user is in W), and N_2 might be the need of a designer to transmit a force from an object to another, say, O_1 and O_2 , in his partial design. Without getting too technical, let me say that the transmission need can be formulated as a constraint between the force variables of the two objects. (In the case of the designer, the relevant W is the world he is constructing, a design, and he is outside W .)

Artifacts (devices) exist, or are created, to meet such needs.¹ That is, they are used to create certain effects in some world into which they are introduced, leading to the satisfaction of the constraint specifying the need in that world. Usually, there is a certain amount of translation needed to identify or create the artifact that can satisfy the need. Typically, such a translation N results in a pair of new constraints, F_W and U , such that if both constraints are satisfied, then N is satisfied. In the example N_1 above, a common F_W is “the light intensity variable in user’s room must be above a certain lumens value,” with a corresponding U , “the user looks at the paper.” One can see that if F_W and U are satisfied, the causal models corresponding to a normal human user will ensure that N_1 will be satisfied.

Usually, an F_W is chosen such that it is easy to identify an existing or designable entity D outside of W , which when suitably introduced into W , can help satisfy F_W . In the example N_1 , there are indeed artifacts, called “lamps,” that are available that are labeled or associated with the F_W that we are seeking. (For simplicity, we are assuming that the lamps are self-contained and battery operated.) However, identifying such an artifact is not sufficient. For example, acquiring such a lamp and storing it in the drawer will not help N_1 to be satisfied. The artifact has to be properly embedded in W and causally interacted with. In Chandrasekaran and Josephson (2000) we use the term *mode of deployment* (M) to denote how D is embedded in W and how D and W affect each other causally. Thus, what one searches for is a pair, $\{D, M\}$, such that when object D is embedded in W satisfying M , F_W is satisfied in W .

Let us review the sequence of ideas. The lamp is associated with the F_W : that is, it can be used to increase the lumens in a room in the mode of deployment, “place in room and turn switch on.” If the user satisfies constraint U ,

namely, “place paper in room and direct eyes to paper surface,” the user’s need, “know what is written on paper surface” is satisfied. In contrast, if the user has a need to look for keys, the user would need to satisfy a different constraint, say, U' : “scan the surface of the room until key is seen.” Note that the mode of deployment is the same in both cases. The mode of deployment is the general way of embedding the artifact in the environment for a variety of uses, and the constraint U is specific to the specific need.

To complete discussion of the examples introduced earlier, in the case of N_2 , no translation is required, and F_W is simply N_2 . A device called a *shaft* is one of the devices typically labeled in terms of torque transmission, and the mode of deployment is appropriately connecting the shaft to the objects O_1 and O_2 , and provided with appropriate support to the external world.

What was the need to translate from N to F_W in the first place? Why not just look for devices that are directly associated with N ? Lamps can help to achieve many specific needs: it can help someone read a book, someone else to recover a lost object, and a third one to avoid bumping into things. The same artifact is essentially able to meet each of the needs, under different U s. If the designer is able to translate N_1 to F_W , then the same artifact under the same mode of deployment can be used to satisfy a variety of needs. Because of the many-to-one mapping from needs to artifacts, the number of artifacts can be a lot smaller in number than the needs to be satisfied. The process of going from N to F_W s is partly individual, partly social.

It is also true that there is a many-to-one mapping from artifacts to needs. That is, several different artifacts with different associated F_W s and under different modes of deployment can satisfy the same need. An artifact with the function “sounds in the air corresponding to the text in paper,” and in the mode of deployment, “place paper with text in appropriate place in the device, and place device close to the user,” will satisfy the same need N_1 , and thus might be more suitable if the user with the need were a blind person.

The availability of objects labeled with F_W s provides directionality and a stopping point in the search process involved in going from N to F_W . Design opportunities arise when someone finds an F_W for which no device exists, and a one-to-many mapping is seen between F_W and user needs. Identification of such F_W s and design of corresponding D s is what inventors are noted for.

The labels F_W are often called “functions.” This sense of function can be called “function-as-effect,” because the artifact’s role is defined in terms of, and it is acquired for, the effects it causes outside of itself.

2.2. Function in device-centric terms

In contrast to function-as-effect, at times, a device’s function is specified in terms of the variables of the device itself. For example, a battery’s function is often specified as

¹It should be clear from the examples we do not intend to restrict the term “need” to the wants of a man on the street. Any agent who desires that certain constraints be satisfied in some world has a need in the sense in which we use the term.

providing a voltage of V volts between *its* terminals. This device-centric specification of the function arises because there is often an understanding between the user and the device maker about the mode of deployment, that is, how to embed the artifact in the world of interest and causally interact with it. This helps the engineer to convert the function-as-effect F_W , expressed in terms of the variables in W , into a desired constraint F_D , expressed as a constraint in terms of the variables of D .

In the case of the electric lamp, there is a tacit agreement between the user and designer that the lamp is to be used with the switch turned on and in the location where a higher illumination is desired. The designers can then conveniently think in terms of a device-centered functional specification, “Produce so many lumens of light in such and such a wavelength range.” Thus, electric bulbs often come labeled with this as their F_D , that is, ratings of lumens produced, not illumination in a room.

As another example, the designer of glass pane for windows might be focusing almost entirely on getting its refractive index, a variable in device terms, to have an appropriate value, because she and the user community have a shared understanding of how the pane is to be used. The function-as-effect characterization of the same glass window would be in terms of the amount of light that passes through. However, although F_D is a common way to specify the function of a device, it is always important to keep in mind that it is a convenience, and devices exist because someone hopes to use the device to make something happen *outside* the device. If the supplier of the device and the user of the device share an understanding about the use of the device, then F_D , the function in device-centric terms, and F_W , the function-as-effect, can be easily translated back and forth, as convenience requires.

The preceding discussion can be captured in a pair of mnemonically useful expressions²:

$$F_W + U \rightarrow N,$$

$$F_D + M \rightarrow F_W.$$

2.3. Understanding how a device works

The framework also helps explain how we *understand* how a device works. Intuitively, understanding how a device works is a story we tell ourselves in which a certain behavior of the device that we characterize as its function is seen to come about as a result of the functions and causal properties of the components and the way they are connected. For simplicity, let us consider a device D with two components, d_1 and d_2 , in structural relation, R , that is, $R(d_1, d_2)$ holds. Let C_1 and C_2 be the causal models (along with the variables) for the components d_1 and d_2 , and let C_R be the

causal model for R . Let F_1, F_2 , and F_D be the device-centric functions of the components and the device, respectively. First, we assure ourselves that $C_i \rightarrow F_i$, for $i = 1, 2$, that is, the functions for which the components were chosen are in fact supported by their causal models. Then, we try to show that $C_1 + C_2 + C_R + R(d_1, d_2) \rightarrow F_D$. That is, the causal properties of the components and the relation imply that the device satisfies its device-centric functional specification. If there are no side effects over the functions for which the components are chosen, we should be able to show that $F_1 + F_2 + C_R + R(d_1, d_2) \rightarrow F_D$. That is, we just use the specified device-centric functions of components instead of the complete causal model C_i . For example, a simple specification of a battery’s function is that it provides a voltage V between its terminals, and thus we can use this alone as its causal model. However, its internal resistance may affect this, so in some cases, the more complete causal model that includes the internal resistance may be needed instead of just the functional specification.

Consider the electric lamp, for example. For simplicity, we model it as having just three components, a bulb, a battery, and a switch in series. Let us consider the following device-centered functional specifications:

$F_W(\text{lamp})$: In mode of deployment $M(\text{lamp, room})$, “when switch turned on, and placed in room,” it illuminates the room.

F_{lamp} : When switch state is ON, it produces light at the rate of L lumens.

F_{bulb} : When its electrical terminals have voltage V between them, current flows between the terminals, and the bulb surface produces light at the rate of L lumens.

F_{battery} : It has voltage V between its terminals and the terminals are electrically continuous.

F_{switch} : If state is ON, terminals are connected; if state is OFF, terminals are disconnected.

R : The bulb, battery, and switch are connected in series.

First, the analyst considers F_{lamp} and $F_W(\text{lamp})$ in $M(\text{lamp, room})$. If a lamp produces light and it is placed in room, that the room will be illuminated requires access to the relevant causal model of spaces such as rooms. Depending on the sophistication of this causal model, the analyst may even be able to develop a sense of variation of illumination in the room as a function of the distance from the lamp. This step in the argument is useful only to see that the lamp will have the intended effect in the room.

To understand how the lamp works, he argues as follows. From the structure and the device-centric functions of the components, he sees that if the switch is ON, a voltage V will be applied between the terminals of the bulb, and as a result it would satisfy its specified F_{bulb} , which in turn, implies that F_{lamp} would be satisfied. He can see the functions, the roles, played by the various components: the bat-

²The notation is informal; $C_1 + C_2$, for example, is to be read as the union of assertions in models C_1 and C_2 .

tery provides the voltage needed for the bulb, the bulb produces light, and the switch helps to close the circuit.

The representation framework in the work by Chandrasekaran and Josephson (2000) supports automation of all of this reasoning. A designer might give the system a configuration of components along with their specified functions/causal models, and the system can decide if the resulting behavior would satisfy a given functional constraint.

3. RELATION TO SIMILAR INTUITIONS IN THE ENGINEERING LITERATURE

To show the usefulness of the framework, let us revisit the distinctions made in some of the quotations on function in an earlier section. Consider the distinction in (Chakrabarti, 1998), between function as “the same level of abstraction as behaviour (intended behaviour),” and as something that “is at a higher level (purpose);” and the distinction in (Deng, 2002) between purpose and action functions. We notice that “purpose” in the above proposals is ambiguous regarding whether it refers to the effect in the environment desired (the *why* of the device) and the purpose in the sense of what the designer wants it do in terms of its variables (the *what* in device terms). By the same token, the meaning of behavior or action is also ambiguous: it could refer to what I just called the *what*, that is, that when something is done to the device it results in a specific output variable having certain values. Or it could refer to the detailed causal chain, the *how*, that realizes the *what*.

Using the buzzer example, the *why* of the device is that we need some indication, some effect in the world outside the buzzer, say alerting of the person inside the house, when a visitor wishes to announce himself. One may legitimately treat this as belonging to the purpose category. The *what* of the device may be described as, “I want something in the device to hit a bell, when its bell is pressed.” With respect to the *why* description, this may be thought of as the behavior, or action-level, sense of function. In contrast, we may treat the *what* description as the purpose of the designer, that is, he wishes to make a device that will actually achieve this transformation. The *how* level description of behavior (the bell pressing causes the circuit to be opened, the current to stop, the magnetic field to die out, the clapper arm to be dropped, etc.) can then be thought of as the action or behavior that helps the designer achieve his purpose.

Thus, it is hard to say if the distinctions between the senses in the various quotations corresponds to the distinction between function-as-effect and function in device-centric terms. However, the important point is that the formalization brings out the ambiguities, which remain hidden because, until the formalization, we think we know what we mean by behavior-level and purpose, but we do not quite realize the opportunity for multiple senses. Once the framework is laid out, we can see, as pointed out by Chandrasekaran and Josephson (2000), that there are at least four different senses in which the term “behavior” can be

used and is used in the literature, with the authors, including Sembugamoorthy and Chandrasekaran (1986), being quite unaware of these multiple possible meanings and the consequent potential for ambiguity.

4. FM

The intention of this section is not to be a review of the entire area that goes under the name of FM, with a heavy representation from mechanical, systems, and process engineers. Instead, I briefly consider two example proposals in the field to highlight what I think is an important feature of the work in this area: their search for a set of what they call “functional primitives.” Then, I place the research in the context of FR work. The choice of just two is motivated by the desire to simplify the discussion, not to slight the other researchers in the area.

Modarres and Cheon (1999) consider domains in which energy and matter conservation laws apply, and they propose the following class of what they call functional primitives: *generate, destroy, maintain, control, transform, transport*. Each of these verbs, behavior terms, can take as argument mass, energy, or momentum.

Stone and Wood (2000) consider devices based on flow, materials, energy and signals, a wide class indeed. They, too, propose a vocabulary of function primitives, but organized into classes and subclasses. Because the vocabulary is extensive, I’ll give a few examples rather than reproduce the list. A class called “Branch” has elements “Separate,” “Refine,” and “Distribute.” They say that English language verbs, “Switch,” “Divide,” “Release,” “Detach,” “Disconnect,” “Disassemble,” and “Subtract,” are synonyms of this. Another class called “Convert” has just one element “convert,” with synonyms “Transform,” “Liquefy,” “Solidify,” “Evaporate,” “Condense,” “Integrate,” “Differentiate,” and “Process.”

My goal here is not to evaluate the proposals, why one should prefer one set of primitives over another, what the criteria for primitives should be, or even whether the notion of primitives is quite what is needed. (However, I do want to remark that the notion of reducing natural language verbs to a small set of primitive was an alluring subject in AI in the 1970s and 1980s. Useful insights were obtained, but the field never quite settled on one set of primitives, nor on criteria that would help choose one set over another.) My goal is narrower: it is to use the proposals to understand the objectives of this body of research and see how they relate to those of FR.

5. COMPARISON: FIRST ROUND

The FM literature usually emphasizes the verb as an essential aspect of their theories. Verbs effect state changes, so there is a mapping from verblike specifications and property change specifications. “Generate(X, L),” for generate X at location L , is a constraint on the location property of X .

Thus, to me, the significant difference between the FM and FR work is not that the former uses verbs for functions, whereas the latter uses constraints on variables.

Both streams are responsive to the widespread intuition that the function of a device is what the device *does*. For example, in both FM and FR, the vocabulary of a function is the same as that for behavior. However, the FR work makes an additional distinction, that not all of what a device does is appropriately called its function. Only some aspects of what the device does is intended or desired, and that subset is what we mean by the device's function, that is, the reason the device is designed or bought. For example, a lamp's behavior at any instant includes the current through its filament, its heat output, the power it consumes, and so forth, in addition to the light it emits. Although all the behaviors other than light emitted may play a causal role in the production of light, those behaviors are not the lamp's function. I think the phrase "behavioral modeling" is a better description of what FM is trying to do than "FM" as such. FM research does not say much about the concept of function as such, but tacitly uses the idea that describing functions of a large class of devices requires a behavioral (or state variable) vocabulary and proceeds to make proposals about that vocabulary.

Because of its interest in clarifying the notion of function, FR makes much of the distinction between device-centric function versus function-as-effect. Clarification of the concept of function is not a topic of interest for FM, which is thus not especially interested in this distinction made by FR.

The most important relation between the FR work and the FM work in my view is that the behavioral primitives proposed by FM should be seen as a *content theory* of aspects of the general framework in which FR operates. To use current terminology, FM refines some of the terms in the *ontology* proposed in FR. As such, the proposals made by FM theories can be incorporated into an enlarged FR framework.

What ontology of FR does FM refine? For those readers who already know what is meant by the terms content theory or ontology, the following explanation may be sufficient. Recall that the representational framework for FR posited objects with causal variables and causal constraints on them, and interobject relations, also characterized by causal constraints on the variables of the objects in the relations. The framework did not make any commitments to what variables exist in what domains. For example, relevant variables for electrical domains would be currents, voltages, and so forth, whereas for a mechanical domain, the variables might be force, displacement, velocity, power, and so forth. Because behaviors are equivalent to changes in the values of state variables, the FM theories can be thought of as making content theory proposals about the state variables. Content theories work by narrowing the focus somewhat, that is, by restricting the domain of interest. I will take a short detour to explain what a content theory is and then return to the comparison.

5.1. Ontologies and content theories

A representation language makes commitments to an ontology of the world, that is, a theory of what sort of things exist and what we can say about them. First-order predicate calculus (PC), which is the basis for all knowledge representation languages,³ makes minimal, but definitive, commitments: that there are objects (individuals), objects have properties, that objects may be in relations defined in terms of the values of their properties, and that there are functions each of which takes a specified number of objects as arguments and specifies an object as its value. PC makes no commitments to other things such as space or time, not to types of objects, types of relations, or types of functions. The lack of additional commitments does not mean that we cannot use PC to represent knowledge about domains involving time, or various types of objects. It is just that PC itself does not offer additional primitive terms to identify the types. It has no theory to offer about them. However, before current technological interest in knowledge representation, a branch of philosophy called ontology has focused on the basic building blocks of reality, so to speak. AI and computer science are continuing ontology studies, but this time with a technological emphasis, to support knowledge representation or some form of automated processing of information. In addition to identifying the conceptual structure of some domain and providing terms to represent the concepts, ontologies offer an additional benefit: they help map synonyms in the same language, or terms in different languages that mean the same concept, to the same term in ontology.

The CML language and the FR framework (CML/FR) can be thought of as providing a refinement of the ontology of PC or a variant of PC for representing causal knowledge about interacting objects. The language makes additional ontological commitments. A specific type of relationship is asserted as potentially existing between objects, *causal interaction*. Thus, representation of causal and structural relations is specifically supported in the language. A type of object called device is identified, which has a distinguished property called "function," along with a mode of deployment.

What advantage does this specialized language have over just the original PC for someone trying to represent knowledge about some class of engineering devices? One advantage is that as long as the knowledge that is being represented is about devices, their components, their compositions, and their functions, appropriate primitives are available to express knowledge. These primitives are both suggestive and enforcing. They are suggestive in the sense that the terms provide a direction for modeling devices in a domain: they suggest

³Contrary to general assumptions, even supposedly nonlogic-based representational frameworks such as frames and scripts are still in this overall framework: the knowledge represented in these frameworks can still be viewed as asserting the existence of certain objects, their properties and relations between them, except that the inferences are not deductive in the strict sense of traditional logic.

looking for devicelike objects, asking about causal variables and intercomponent relations, about which constraints should be identified as intended effects on the environment, and so forth. They also enforce meaningfulness in the sense that the terms cannot be arbitrarily combined. For example, there are only certain kinds of things that can be said about the relations between two components and only certain kinds of things can be said about the components themselves. Finally, specialized inference rules can be defined within this language. For example, given a set of component objects in a certain relationship, inference rules can be specified to infer the behaviors of the configuration from the causal models of the components and their relations.

We see that ontology development is hierarchical in the following sense. PC and its variants provided a basic ontology of objects, properties, relations, and so forth. CML/FR refined these for the domain of devices by identifying types of relations and objects that are applicable to this domain, but it does not make any commitments to any specific family or domain of devices. For example, that in audio engineering, *oscillate* and *amplify* are two important behavioral primitives is not something that CML/FR would provide a modeler. We can continue the hierarchical ontology development by focusing on classes and subclasses of devices. An ontology of properties/behaviors/functions and device types and subtypes (e.g., electronic) might be identified. At each level of the hierarchy, we can provide the model maker with a catalog or a vocabulary of terms to use in representing the systems and devices of interest in the class or subclass.

This is exactly what FM theories try to do, for at least the behavior dimension of the representation.

6. CHALLENGES IN DEVELOPING ONTOLOGIES

As mentioned in Chandrasekaran et al. (1999), arriving at the set of concepts in terms of which the domain facts can be stated is “carving reality at its joints,” a deep intuition about what the structure of reality *is*. Such an analysis calls for skill in bringing together three different aspects of knowledge: the structure of reality that is part of our commonsense view of the world, scientific discoveries about the structure of some aspect of reality, and finally, pragmatic needs that add a layer of concepts for specific purposes.

The structure of reality that filters all our understanding is independent of science. Logicians have tried to come up with the ontology of this, aspects of which remain controversial. However, there is much agreement, as Chandrasekaran et al. (1999) says:

. . . there is agreement that there are *objects* in the world; these objects have *properties* that can take *values*; the objects may exist in various *relations* with each other;

the properties and relations may change over *time*; there are *events* that occur at different *time instants*; there are *processes* in which objects participate and that occur over time; the world and its objects can be in different *states*; events may *cause* other events as *effects*; and objects may have *parts*. Further, perhaps not as basic facts of the world but as ways of organizing them, is the notion of *classes*, *instances*, and *subclasses*, where “classhood” is associated with shared properties. Thus, *Is-A* relations indicating subclass relations are fundamental for ontology representations.

None of the terms mentioned above come from science as such, but they reflect a sophisticated analysis of what our language tells us about the basic terms in which to describe the world. As mentioned before, PC makes commitments to a small part of the above list.

The second source of a content theory, especially in specific domains, is what scientific theories say about what sort of entities exist. In a sense, doing science is ontology making: when Newtonian physics came up with the concept of force and the related notions of energy, work, power, and so forth, science was coming up with the right ontology for a certain class of phenomena. Similarly, when the science of electricity identified concepts such as current, voltage, impedance, and so forth, an ontology was being created. These make use of the commonsense ontology just mentioned, but they go beyond it and are empirical in nature.

The third source of a content theory, especially for device representations, is a behavioral repertoire that is consistent with both the commonsense⁴ and scientific ontologies, but specifies additional terms with which to describe relevant behaviors. For example, in the audio engineering domain, two of the behaviors of interest in the domain are *oscillation* and *amplification*. These behavior terms are of interest because often a designer wishes for these behaviors with specified parameters. The science of the domain itself does not directly provide these terms. In contrast to terms such as current, voltage, impedance, and relations between them given by Ohm’s, Kirchoff’s, and Maxwell’s laws, behavioral terms such as those just mentioned arise in the context of device making with certain kinds of desired behaviors in mind. That is, although these behavioral terms can be defined in terms of the underlying scientific and commonsense ontology, they go beyond them in important ways.

The right ontology is *expressive*, that is, it enables one to represent the all relevant facts, and is at the same time *economical*, that is, it provides terms that are necessary and

⁴Caution: Commonsense ontology is used in the technical sense in which it is used in the AI knowledge representation literature, and refers to the tacit knowledge about the world shared by humans as they go about their everyday life. A significant part of the formal ontology work is to develop the concepts and terms to represent this knowledge, which would be required for human-level performance in natural language understanding and problem solving.

sufficient as building blocks for additional concepts in the domain. The economy requirement assures that the basic terms in the content theory do not proliferate, and that the meanings of the new terms composed out of the basic terms can be precisely characterized. Continuing with the audio engineering example, the insight that oscillation and amplification have an underlying mathematical formulation in which they differ in the properties of a parameter provides an economy of representation.

At times, there are insights about the behavioral terms that transcend a particular domain, and are directly built on top of unifying themes in the commonsense ontology. For example, many people have noticed that there is a commonality of behaviors that are of interest in phenomena involving flow: whether it is liquid flow, or of electricity, or even of information. Behavioral terms such as storage (e.g., liquid containers/batteries/buffers) and conduits and their flow-carrying capacities are abstractions that apply to all domains involving flow. Identifying a set of basic terms of such behaviors (functions, when these behaviors are desired or intended) such that different behaviors of interest can be composed out of them is important for effective content theories.

7. COMPARISON: SECOND ROUND AND CONCLUDING REMARKS

7.1. FR and FM theories as content theories

FR is a content theory that makes specific ontological commitments about the sorts of objects that are involved in making devices, and FM theories are further refinements of the device ontology in the behavior dimension for certain subclasses of devices.

FM proposals are not driven by the automated reasoning and knowledge representation goals of AI research, and hence, they do not have the same requirements for formality that AI proposals have to have. FR representations, for example, can be automatically processed, by very general domain-independent mechanisms, to support automated simulation of compositions of components. The generality of the FR representational framework can also support reasoning such as whether a proposed design as a composition of components achieves a given device-centric function, and, given additionally the mode of deployment in a world, whether it achieves a specified function-as-effect.

The FM ontologies are not formal; however, the proposed applications of the content theories do not require such a formal language. For example, Modarres (1997) shows how informal (diagrammatic) models of systems behaviors are represented using the proposed vocabulary, and how that model can guide diagnostic reasoning. Thus, as long as we intuitively understand the terms *generate*, *destroy*, and so forth, if necessary by following the examples, we know what the terms mean and can use them to model systems.

Stone and Wood (2000) are more ambitious for the uses of their vocabulary, but still their intended applications do not require much in the way of formal representations. They think that once the ontology that they propose is generally accepted, that is, if everyone would describe the behaviors of their systems using the proposed vocabulary, it would be easy to share components and functions by automatic matching of behaviors of subsystems or fragments of behavior. Similarly, structures (partial designs) can be indexed with the behaviors that they support, and given desired behaviors, design possibilities may be generated from a catalog of behaviorally indexed structures. They mention other applications, but all applications make use of the fact that each behavioral element, for which in natural language there are many synonyms, is represented using a unique term. This makes it possible to standardize behavioral representation and use simple matching algorithms for the various applications. This, of course, requires that the vocabulary proposed has the properties that a good ontology should have.

The FM proposals do not spend much time worrying about representational aspects that deal with the fine details of structure and behavior needed for automated simulation of behavior and design verification.

In my discussion on the challenges in the development of ontologies, I mentioned that they arise from commonsense as well as scientific models of reality. Thus, both proposals are based on a certain combination of scientific terms (mass, force, momentum, etc.) with metascientific commonsense/use-oriented terms such as “turn,” “generate,” “destroy,” and so on. Given the disciplinary provenance of the investigators, the behaviors that interest them relate to physical dynamic phenomena covered by Newtonian mechanics (force, momentum, energy, power, etc.) and commonsense behaviors such as “move,” “turn,” and so forth, what power plants and automobiles do, rather than what computer programs and electronic adders do.

The two FM proposals described had similar goals, but differed in their proposals for the vocabulary. My discussion on why ontology development is challenging is intended to explain the difficulties in coming up with and justifying proposed vocabularies. The appropriate methodology for comparing, verifying, and modifying the vocabulary proposals is one of extensive experimentation, rather than arguments based on intuition.

The points I made in comparing the FR and FM streams also suggest how the two streams might merge over time. Ontologies of the sort being developed by FM modelers are certainly going to be useful for device knowledge representation, because, as I have mentioned, the current body of representational primitives in AI do not have terms for properties, behaviors, and functions for devices in specific domains.

AI device representation can make use of the ontology refinements being developed for various device classes by FM modelers, with additional formalization. I think it would be useful for FM researchers to squarely identify them-

selves as part of ontology research that is going on in various places, mostly driven by information search and sharing opportunities and challenges that arise as a result of the World-Wide Web revolution. The advantage of this identification will be possibly to make use of a body of existing infrastructure for ontology representation and sharing, and to add precision to the FM vocabulary proposals.

ACKNOWLEDGMENTS

This work was prepared through participation in the Advanced Decision Architectures Collaborative Technology Alliance sponsored by the U.S. Army Research Laboratory under Cooperative Agreement DAAD19-01-2-0009. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The author thanks David Brown, Amresh Chakrabarti, and Robert Stone for useful comments and suggestions on earlier drafts.

REFERENCES

- Chakrabarti, A. (1998). Supporting two views of function in mechanical designs. *AAAI Workshop on Reasoning About Function*. Menlo Park, CA: American Association for Artificial Intelligence.
- Chakrabarti, A., & Bligh, T.P. (2001). A scheme for functional reasoning in conceptual design. *Design Studies* 22(6), 493–517.
- Chandrasekaran, B. (1994a). Functional representation and causal processes. *Advances in Computers* 38, 73–143.
- Chandrasekaran, B. (1994b). Functional representation: a brief historical perspective. *Applied Artificial Intelligence* 8, 173–197.
- Chandrasekaran, B., & Josephson, J.R. (2000). Function in device representation. *Engineering with Computers* 16, 162–177.
- Chandrasekaran, B., Josephson, J.R., & Benjamins, V.R. (1999). What are ontologies and why do we need them? *IEEE Intelligent Systems* 14(1), 20–26.
- Chittaro, L., & Kumar, A.N. (1998). Reasoning about function and its applications to engineering. *Artificial Intelligence in Engineering* 12(4), 331–336.
- Deng, Y.-M. (2002). Function and behavior representation in conceptual mechanical design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 16(4), 353–362.
- Falkenhainer, B., & Forbus, K. (1991). Compositional modeling: finding the right model for the job. *Artificial Intelligence* 51(1–3), 95–143.
- Faltings, B. (1990). Qualitative kinematics in mechanisms. *Artificial Intelligence* 44(1–2), 89–199.
- Modarres, M., & Cheon, S.W. (1999). Function-centered modeling of engineering systems using the goal tree–success tree technique and functional primitives. *Reliability Engineering and Systems Safety* 64, 181–120.
- Sembugamoorthy, V., & Chandrasekaran, B. (1986). Functional representation of devices and compilation of diagnostic problem-solving systems. In *Experience, Memory, and Reasoning* (Kolodner, J.L., & Riesbeck, C.K., Eds.), pp. 47–73. Hillsdale, NJ: Erlbaum.
- Stone, R.B., & Wood, K.L. (2000). Development of a functional basis for design. *Journal of Mechanical Design* 122(4), 359–370.

B. Chandrasekaran is a Professor Emeritus and Senior Research Scientist, Department of Computer Science and Engineering, and Director, Laboratory for AI Research, both at Ohio State University. Professor Chandrasekaran received his BEng (Honors) in electrical engineering from A.C. College of Engineering and Technology (Madras University, Karaikudi, India) in 1963 and his PhD in electrical engineering from the Moore School of Electrical Engineering, University of Pennsylvania, in 1967. Dr. Chandrasekaran's research interests include understanding intelligence from a computational perspective, in particular, intelligence as expressed in high-level cognitive activities such as problem solving, understanding, and explanation.