An Ontology of Functional Concepts of Artifacts

Yoshinobu Kitamura
and
Riichiro Mizoguchi

AI-TR-99-1
Received March 25, 1999

Artificial Intelligence Research Group
The Institute of Scientific and Industrial Research
Osaka University
8-1, Mihogaoka, Ibaraki, Osaka 567-0047, Japan
Email: {kita,miz}@ei.sanken.osaka-u.ac.jp

Abstract

This paper proposes an ontology of functional concepts of artifacts, which provides a rich and well-organized vocabulary for functional representation. It includes nine types of the roles played by a function for another function, named meta-functions, such as “to enable” and “to drive”. They represent interdependence between functions and an important part of the design rationale of artifacts. It provides vocabulary for explanation to human designers and primitives specifying functional reasoning space for (re)design systems.

1 Introduction

Understanding of functionality is important for human understanding of artifacts, because functionality represents a part of the design rationale [Chandrasekaran *et al.*, 1993; Lee 1997]. It is required essentially for redesign of existing artifacts in order to consider the intention of the original design [Goel and Chandrasekaran, 1989]. A crucial factor here is such an understanding should be shared by designers and the redesign supporting systems. To enable the sharing, we need a well-defined functional vocabulary comprehensible to both kinds of the agents.

Humans use many functional concepts for explanation of the same component. For example, we can explain functions of a boiler in a power plant in terms of several verbs such as "*give heat* to the steam", "*make* vaporization", "*enable* the heat-expansion in the turbine", "*drive* rotation of the turbine-shaft" and "*partly achieve* power-generation". Such functional concepts should be defined with explicit relations to the physical behavior which the reasoning systems can deal with. The crucial issue here is not to assign the correct English labels to the concepts but to articulate such concepts and discriminate them.[1]

Furthermore, each of them captures a different aspect of the target system. In those examples, although the first two refer only the boiler itself, the rest refer other functions of other components. The latter represent the dependency among functions of components, and hence are important to represent the justification of the existence of the function, i.e., the design rationales. Characterization of such categories of functional concepts is needed to establish a common vocabulary.

A lot of research has been carried out on functional representation of artifacts. Almost all functional models in the conventional research, however, are specific to the target system. While there are a few generic functional concepts [Keuneke, 1991; Lind, 1994], they are not well organized in, say, is-a hierarchy.

On the other hand, in Value Engineering research [Miles, 1961], standard sets of verbs (i.e., functional concepts) for value analysis of artifacts are proposed [Tejima *et al.*, 1981]. It enables the human designers to share descriptions of functions of the target artifacts. However, they are designed only for humans, and there is no machine understandable definition of concepts.

Furthermore, because the conventional research focuses on abstraction of behavior of components [Umeda *et al.*, 1990; Lind, 1994; Sasajima *et al.*, 1995] or hierarchical abstraction of behavior [Sembugamoorthy and Chandrasekaran, 1986; Vescovi *et al.*, 1993], the dependency among functions such as "to enable" and "to drive" in those examples has been left unknown. We have extended a representation language of behavior and functions of components named FBRL [Sasajima *et al.*, 1995] in order to cope with such dependency.

[1] We do not claim the appropriateness of the labels shown in this paper, because the authors are not native English speakers.

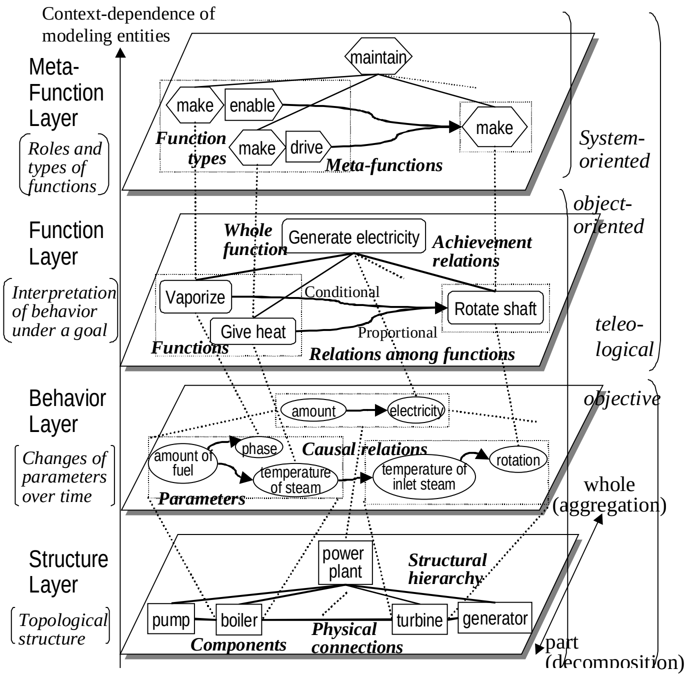

Figure 1: The four layers of an example model

This paper proposes an ontology of functional concepts of artifacts, which provides a rich vocabulary for functional representation including the dependency among functions. An ontology is defined as an explicit specification of conceptualization [Gruber 1993] and theory of a primitive vocabulary for knowledge-based systems [Mizoguchi and Ikeda, 1997]. The functional concepts in our ontology are categorized into four spaces, which are organized in *is-a* or *part-of* hierarchy. They are clearly defined by conditions of behavior and the additional information for teleological interpretation.

We also propose a new category of functional concepts named *meta-function* in order to represent dependency among function. A meta-function represents a role played by a function for another function in order to make the whole system work collaboratively. We have defined nine types of them such as "to enable" and "to drive".

Explicit conceptualization of domain knowledge as ontologies helps several problem solving [Abu-Hanna and Jansweijer, 1994; Borst *et al.*, 1997]. We will mention utility of our ontology in explanation and redesign.

Firstly, we describe the layers of a model and introduce the meta-function layer. Next, the ontology of functional concepts is discussed. Section 4 mentions the utility of the ontology. Section 5 discusses the contribution of this work by comparison with the related work. Section 6 describes the limitations of the ontology.

2 Structure, behavior, function and meta-function

This section overviews the general structure of our modeling scheme using a part of the model of a power plant shown in Figure 1 an example. There are three axes.

The vertical one has four layers. The layers of structure, behavior and function are similar to those proposed in literature such as [Lind, 1994]. The meta-function layer is introduced here. The horizontal axis represents relations among entities of the same grain size. The last axis to the depth represents the grain size of the entities. An entity is a part of deeper one. In each layer, there is a whole-part hierarchy.

Structure layer and Behavior layer. The structure layer describes topological structure of the artifact. We adopt the device-oriented ontology [de Kleer 1984]. It describes the existence of components, physical connections among them, and a structural hierarchy. The behavior layer represents changes of parameter values of entities over time. The horizontal relations represent causal relations among parameters. The whole-part relations represent hierarchical abstractions of behavior.

Function layer. A function of a component is defined as a result of interpretation of a behavior of the component under an intended goal [Sasajima *et al*., 1995]. The functions of the boiler includes “to vaporize water” and “to give heat to water” as shown in Figure 1. The horizontal relations among functions are causal or structural. The causal-type relations are categorized into some subtypes such as proportional and conditional according to causal relations among parameters that functions focus on. The former represents that a parameter value has proportional relation with another parameter. The latter represents the case where the condition is discrete such as the condition that the phase of the steam has to be gas for a turbine. On the other hand, the structural-type relations are categorized into subtypes such as series, parallel, sequential, simultaneous and feedback. The whole-part relations represent that the whole functions are achieved by groups of sub-functions (called achievement relations).

Meta-Function layer. The meta-function layer describes the roles of each function for another function (called meta-functions) and the types of functions (called function types), while other three layers are concerned with existence or changes of objects. For example, the “to vaporize” function of the boiler is said to be ToMake-type and to perform a meta-function “to enable” for the “to rotate” function of the turbine according to the definition shown in Section 3.3. On the other hand, another function “to give heat” is said to perform a different meta-function “to drive” for the same “to rotate”.

3 Ontology of functional concepts

The ontology of the functional concepts consists of the four spaces as shown in Figure 2.

3.1 Base Functions

A function of a component can be represented by a transitive verb of which grammatical subject is the component and of which grammatical objects are the entities coming in and going out of the component. Although a function depends on the context, the description itself should be local. In comparison with the meta-functions, we call the functions *base functions*.

A behavior can be interpreted from object-related aspect (called object functions), energy-related aspect (called energy functions) or information-related aspect (called information functions). For example, the object function and the energy function of a turbine are “to rotate the shaft” and “to generate kinetic energy”, respectively.

Figure 2a shows the energy base functions organized in an is-a hierarchy with clues of classification. A base function is defined by conditions of behavior and the information for its interpretation called Functional Toppings (FTs) of the functional modeling language FBRL (abbreviation of a Function and Behavior Representation Language) [Sasajima *et al*., 1995]. There are three types of the functional toppings; (1)O-Focus representing focus on attributes of objects, (2)P-Focus representing focus on ports (interaction to neighboring components), and (3)Necessity of objects.

For example, a base function “to take energy” is defined as “an energy flow between two mediums” (a behavioral condition), and “focus on the source medium of the transfer” (functional toppings). Moreover, the definition of “to remove” is that of “to take” plus “the heat is unnecessary”. Thus, “to take” is a general (super) concept of “to remove” as shown in Figure 2a.

Note that definition of base function using FTs is highly independent of its *implementation* by which we mean that details of behavior and internal structure of the component. For example, P-Focus specifies not concrete location but abstract interaction with the neighboring components. The implementation is represented by the ways of achievement discussed in section 3.4.

3.2 Function Types

The function types represent the types of goal achieved by the function [Keuneke, 1991]. We have redefined the function type as “ToMake”, “ToMaintain”, and “ToHold”. For example, consider two components, an air-conditioner (as a heating device) and a heater, having the same function “to give heat”. The former keeps the goal temperature of a room. The latter does not. These are said to be “ToMaintain” and “ToMake”, respectively. We may say that the object words of these verbs are the base functions of components.

3.3 Meta-Functions

The meta-functions (denoted by *mf*) represent a role of a base function called an agent function (f_a) for another base function called objective function (f_o). Figure 3 shows examples of the meta-functions $\{mf_1 \ldots mf_{11}\}$ in a simple model of a power plant (part). Note that the furnace which is a sub-component of a boiler is separated from the boiler as a heat exchanger for explanation.

We have defined the nine types of meta-functions as shown in Figure 2c and Table 1. Table 1 shows the necessary conditions for the meta-functions of relations among functions (discussed in Section 2) and behavior of the parameter on which f_a focused in the $f_{o.}$

To Provide (provider role)

When a function f_a generates (or transfers) the *materials* which another function f_o intentionally processes, the function f_a is said to perform the meta-function "to provide material" for the function f_o. The material of f_o can be defined as input objects (or energy) which will be a part of the output objects on which the function f_o focuses. For example, the "to transfer water" function of the pump has the meta-function "to provide" for the "to vaporize" function of the boiler (see mf_1 in Figure 3).

When what f_o focuses on is energy (i.e., f_o is an energy function), the source energy of the output energy is called *material energy*. For example, the "to generate heat" function of the burner has the meta-function "to provide" for the "to give heat" function of the boiler (see mf_4), because the combustion gas carries the heat energy. In general, energy functions have the meta-function "to provide" for other energy functions such as mf_4 and mf_{10}.

To Drive (driver role)

We call such non-material energy that essentially causes the internal process of the function f_o and then is consumed by the process as *driving energy*. The function which generates or transfers such a driving energy is said to have the meta-function "to drive f_o". For example, the "to give heat" function of the boiler has the meta-function "to drive" for the "to rotate" function of the turbine (see mf_9), because the heat energy is not material of revolution of the shaft and is consumed by the rotation.

On the other hand, for "to generate kinetic energy", the same function performs not "to drive" but "to provide" (see mf_{10}), because the heat energy is material of the kinetic energy. In general, the "to drive" is a role of an energy function for an object function.

	Object/ Energy	Manda-tory	Type of relation	Material	Con-sumption
Provide	*	✔	*	✔	*
Drive	Energy	✔	Proportional	✘	✔
Enable	*	✔	Conditional	✘	✘
Improve	*	✘	Proportional	*	*
Enhance	Energy	✘	Proportional	*	*

*(Legends: ✔ must be, ✘ must not, * don't care)*

Table 1: The necessary conditions of the meta-functions (part)

To Enable (enabler role)

This meta-function is used for representing a condition playing a crucial role in f_o except "to provide" and "to drive" (see Table 1). For example, because the steam of which phase is gas plays a crucial role in occurrence of the heat-expansion process in the turbine and the phase is neither material of rotation nor the consumed energy, the "to vaporize" function of the boiler is said to have the meta-function "to enable" (see mf_5). On the other hand, it performs different meta-function "to provide" for the "to condense" function of the condenser (see mf_6), because the functions focus on the same phase parameter.

To Allow and To Prevent

These two meta-functions are concerned with the undesirable side effects of functions. A function f_a having positive effects on the side effect of a function f_{o1} is said to have a meta-function "to allow the side-effects of f_{o1}".

The "undesirable side effect" is defined in a relation with another function f_{o2} or the whole system. If a serious trouble (e.g., faults) is caused in a function f_{o2} when a function f_a is not achieved, the function f_a is said to have a meta-function "to prevent malfunction of f_{o2}". For example, the "super-heat" function of the boiler prevents malfunction of the turbine (mf_7), because the steam of low temperature would damage the turbine blade. In gen-

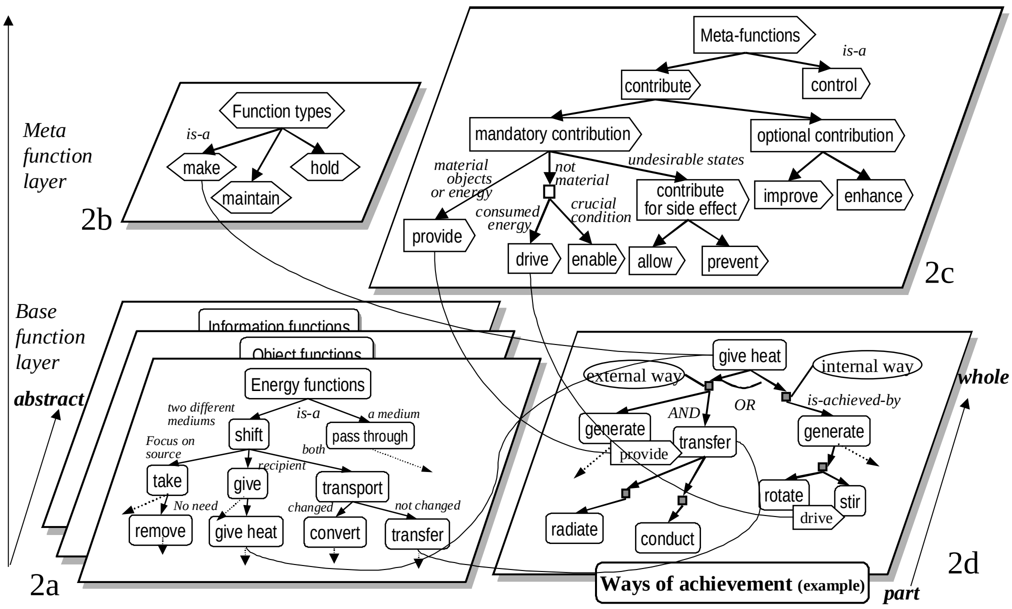

Figure 2: The four spaces of the ontology of functional concepts (part)

eral, almost all f_a performing "to allow" for f_{o1} also have a meta-function "to prevent" for the function f_{o2}.

To Improve and To Enhance

These meta-functions represent *optional* contribution for f_o, The discrimination between "to improve" and "to enhance" is made by increment of the amount of the input energy. For example, the "to keep low pressure" of the condenser contributes to the efficiency of the "to rotate" function (see mf_{11}) without increment of input energy. On the other hand, the "to super-heat" function of the boiler optionally increases the amount of the input energy (mf_8).

To Control (controller role)

When a function f_a regularizes the behavior of f_o, its meta-function is said to be "to control f_o". For example, consider a control valve which changes the amount of flow of the combustion gas for the boiler in order to maintain the amount of flow of the steam. It is said to have the meta-function "to control" for the "to vaporize" function of the boiler (not shown in Figure 3). Although the base function of the control valve from the aspect of the information flow is also represented by the same word "to control", the meanings are different. The function type of a macro component which consists of the boiler and the control valve is called to be "ToMaintain".

3.4 Ways of achievement

A function of f_u can be achieved by the different groups of sub-functions. We call a group of sub-functions $\{f_1, \ldots f_n\}$ constrained by the relations among them (such as meta-functions) *a functional method* of an achievement of f_u. On the other hand, we call the basis of the method *a functional way*. The way is the result of conceptualization of the physical law, the intended phenomena, the feature of physical structure, or components used.

Figure 2d shows some ways of achievement of "to give heat to an object", which are described in terms of concepts in other three spaces. There are two ways, that is, the external and internal heat source ways. According to the external heat-source way, it is decomposed into two sub-functions, that is, "to generate heat" and "to transfer heat". The former should perform the meta-function "to provide" for the latter. In the figure, the latter function can be decomposed according to "radiation way" or "conduction way".

Note that Figure 2d shows is-achieved-by (whole-part) relations among the functional concepts in OR relationship, while Figure 2a shows is-a relations as the definitions of them, which are independent of "how to realize them".

4. Utility of the Functional Ontology

To help organize domain knowledge. The ontology can help the modeler to describe a model of a concrete system with the classes of entities and the constraints among them as discussed in [Abu-Hanna and Jansweijer, 1994].

To provide vocabulary for explanation. We have developed an explanation generation system based on this functional framework (partly discussed in [Sasajima *et al.*, 1995]). It has been successfully applied to a simple model of a power plant and a concrete chemical plant. The explanation of the latter is verified by domain experts. It can provide explanations at several abstraction levels in terms of well-defined sharable functional concepts and thus facilitate to share the understanding of the target systems by human designers.

To explain rationales as a system. The meta-functions are used for explaining rationales of organization of the target system from the viewpoint of the system engineering without mention of changes of entities specific to the target system, that is, what types of collaboration with other functions are done by the functions in order to make the whole system work.

To specify reasoning space for (re)design. Currently, the investigation on a redesign system to propose improvements of existing artifacts is in progress on the basis of this functional framework [Kitamura *et al.*, 1998]. In general, functional representation enables the (re)design system to reason at the functional level as discussed in a rich literature (e.g., [Bradshaw and Young, 1991; Goel and Chandrasekaran, 1989; Hodges, 1992; Bhatta and Goel, 1997]). The ontology provides us a comprehensive vocabulary for design knowledge and specifies the reasoning space. For example, as discussed in [Hodges, 1992], the design system can select a component suitable to realize a function from the component library indexed in terms of the systematized functional concepts in the ontology.

To enable redesign with consideration of dependency among functions. The meta-functions enable us to capture the dependency among functions at an abstraction

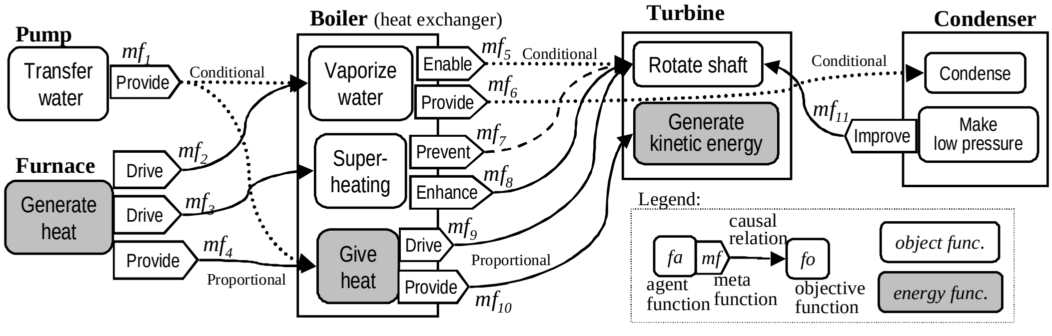

Figure 3: Meta-functions in a power plant (part)

level. As a way of the dependency management [Lee, 1997], the model of dependency among functions in terms of meta-functions enables the redesign system to propose drastic improvements of an existing artifact with consideration of such dependency in the original design.

To enable automatic identification of functional structure. Our design system includes a functional understanding module which automatically identifies plausible functional structures from the given structural and behavioral models [Kitamura and Mizoguchi, 1998]. Although the functional understanding task is in principle difficult because the search space of function is huge, the ontology plays a role to limit the search space. The ontology provides such primitives in the functional space that are targets in the mapping, and screens out meaningless functional interpretations. The functional understanding plays a role to reveal the dependency among functions of the given artifacts.

5. Related Work and Discussion

The functions in [de Kleer, 1984; Umeda *et al.*, 1990; Lind, 1994; Sasajima *et al.*, 1995; Chandrasekaran and Josephson, 1996] represent abstracted behavior of components and are defined as base functions in our framework. The meta-function represents a role for another function without mention of changes of incoming or outgoing objects of components and hence is totally different from such base functions.

In [Sembugamoorthy and Chandrasekaran, 1986; Vescovi *et al.*, 1993], function is defined as a kind of hierarchical abstraction of behavior. It corresponds to the is-achieved-by relations among functions in our framework. The consolidation theory [Bylander and Chandrasekaran, 1985] tries to capture the general patterns of aggregation of the system. In a literature on design, many general "patterns" of synthesis are proposed (e.g., [Bradshaw and Young, 1991; Bhatta and Goel, 1997]). Our general ways of achievement, however, explicitly represent the feature of achievement such as theory and phenomena. The importance of such key concepts in design is pointed out in [Takeda *et al*., 1990]. They enable the redesign system to facilitate the smooth interaction between models at the structural and functional levels.

The CPD in CFRL [Vescovi *et al.*, 1993] represents causal relations among functions which correspond to relation among functions on the function layer in our framework. Lind categorizes such relations into Connection, Condition and Achieve [Lind, 1994]. Rieger identifies "enablement" as a type of the causal relation between states and action [Rieger and Grinberg, 1977]. The meta-functions are results of interpretation of such causal relations between functions under the role of the agent function for the objective functions.

Kueneke proposes four types of functions; ToMake, ToMaintain, ToPrevent, and ToControl [Keuneke, 1991]. Although we inherit ToMake and ToMaintain as the function types, we distinguish the rest as the meta-functions and extend them in order to explicate the objective function contributed to or to be controlled. Our other functional type "ToHold" is similar to "ToAllow" identified in [Bonnet, 1992].

In [Hodges, 1992; Borst *et al.*, 1997], the sets of "primitives of behavior" are proposed. Lind identifies a few general functions such as "storage of energy" which are categorized into multiple levels [Lind, 1994]. We added more intention-rich concepts such as "remove" with unnecessary intention to the set of the base functions and organized in is-a and part-of hierarchy. Furthermore, we identify some types of the meta-functions.

6. Limitations

Generality*:* Currently, our ontology is designed for such a system that has something flowing among components which carries the energy. It has been successfully applied to modeling and explanation generation of a power plant, a chemical plant, and a manufacturing process. Their models share many functional concepts except those specific to the chemical domain such as "react". The ontology currently does not cover mechanical phenomena.

Completeness and Naturalness. We do not claim completeness of the set of concepts. Note that we define precisely the meaning of concepts for discrimination. The definitions may narrow than those we use in the natural language, because we tend to use them confusedly.

Meta-functions in the achievement relations. The meta-functions discussed in this paper are concerned with dependency among functions of the same grain-size. That among functions of the achievement relations in the different grain-size is under consideration.

7. Summary

We proposed an ontology of functional concepts including the meta-functions, aiming at a common vocabulary comprehensible to human designers and a redesign supporting system. Although our ontology is not formally defined in terms of the logic such as Ontolingua [Gruber, 1993], we have defined clearly the meaning of functional concepts using the mapping primitives to the behavior called FTs. Such definitions enable the redesign supporting system to facilitate the smooth interaction between models at the structural and functional levels. We are currently designing a language named FBRLII supported by this functional ontology.

References

[Abu-Hanna and Jansweijer, 1994] A. Abu-Hanna, and W. Jansweijer, Modeling domain knowledge using explicit conceptualization, *IEEE Expert*, 9(5):53-63, 1994.

[Bhatta and Goel, 1997] S. R. Bhatta, and A. K. Goel. A functional theory of design patterns. In *Proc. of IJCAI-97*, 294-300, 1997.

[Bonnet, 1992] J.C. Bonnet. Towards a formal representation of device functionality. TR 92-54, Knowledge Systems Laboratory, Stanford University, 1992.

[Borst *et al.*, 1997] P. Borst, H. Akkermans, and J. Top. Engineering ontologies. *Int. J. Human-Computer Studies* 46:365-406, 1997.

[Bradshaw and Young, 1991] J. A. Bradshaw and R. M. Young. Evaluating design using knowledge of purpose and knowledge of structure. *IEEE Expert* 6(2):33-40, 1991.

[Bylander and Chandrasekaran, 1985] T. Bylander and B. Chandrasekaran. Understanding behavior using consolidation, *Proc. of IJCAI-85*, .450-454, 1985.

[Chandrasekaran *et al.*, 1993] B. Chandrasekaran, A. K. Goel, and Y. Iwasaki. Functional representation as design rationale. *COMPUTER*, .48-56, 1993.

[Chandrasekaran and Josephson, 1996] B. Chandrasekaran and J. R. Josephson. Representing function as effect: assigning functions to objects in context and out. In *Proc. of AAAI-96 Workshop on Modeling and Reasoning with Function*.

[de Kleer, 1984] J. de Kleer. How circuits work, *Artificial Intelligence* 24:205-280, 1984.

[Goel and Chandrasekaran, 1989] A. Goel and B. Chandrasekaran, Functional Representation of Designs and Redesign Problem Solving. *Proc. of IJCAI-89,* 1388-1394, 1989.

[Gruber, 1993] T. R. Gruber. A translation approach to portable ontology specifications, *Knowledge Acquisition*, 5(2):199-220, 1993.

[Hodges, 1992] J. Hodges. Naive mechanics - a computational model of device use and function in design improvisation. *IEEE Expert* 7(1):14-27, 1992.

[Keuneke, 1991] A. M. Keuneke, A. device representation: the significance of functional knowledge. *IEEE Expert*, 24:22-25, 1991.

[Kitamura and Mizoguchi 1998] Y. Kitamura and R. Mizoguchi, Functional ontology for functional understanding, Papers of *Twelfth International Workshop on Qualitative Reasoning (QR-98)*, .77-87, 1998

[Kitamura *et al.*, 1998] Y. Kitamura, T. Ueda, M. Fuse, and R. Mizoguchi. Towards Redesign of Manufacturing Processes based on Functional Understanding, Poster proceedings of *AI in Design 98,* 1998.

[Lind, 1994] M. Lind, M. Modeling goals and functions of complex industrial plants. *Applied Artificial Intelligence*, 8:259-283, 1994.

[Lee, 1997] J. Lee. Design rationale systems: understanding the issues. *IEEE Expert*, 12(3):78-85, 1997.

[Miles, 1961] L. D. Miles. *Techniques of value analysis and engineering*. McGraw-hill, 1961.

[Mizoguchi and Ikeda, 1997] R. Mizoguchi and M. Ikeda. Towards ontology engineering. In *Proc. of PACES/SPICIS '97,* 259-266, 1997.

[Rieger and Grinberg, 1977] C. Rieger and M. Grinberg. Declarative representation and procedural simulation of causality in physical mechanisms. In *Proc. of IJCAI-77*, 250-256, 1977.

[Sasajima *et al.*, 1995] M. Sasajima, Y. Kitamura, M. Ikeda, and R. Mizoguchi. FBRL: A Function and Behavior Representation Language. In *Proc. of IJCAI-95*, 1830-1836, 1995.

[Sembugamoorthy and Chandrasekaran, 1986] V. Sembugamoorthy and B. Chandrasekaran. Functional representation of devices and compilation of diagnostic problem-solving systems. In *Experience, memory and Reasoning*, 47-73, 1986.

[Takeda *et al*., 1990] H. Takeda, P. Veerkamp, T. Tomiyama, and H. Yoshikawa. Modeling design processes. *AI Magazine*, 11(4), 37-48, 1990.

[Umeda *et al*., 1990] Y. Umeda et al. Function, behavior, and structure. *AI in Engineering*, 177-193, 1990.

[Tejima *et al.*, 1981] N. Tejima, *et al*. (eds.). Selection of functional terms and categorization. Report 49, Soc. of Japanese Value Engineering (In Japanese), 1981.

[Vescovi, *et al.*, 1993] M. Vescovi, Y. Iwasaki, R. Fikes, and B. Chandrasekaran. CFRL: A language for specifying the causal functionality of engineered devices. In *Proc. of AAAI-93*, 626-633, 1993.