Proc. of The Sixth Pacific Rim International Conference on Artificial Intelligence (PRICAI 2000), pp.723-733, Melbourne, Australia, August, 2000

Functional Understanding based on an Ontology of Functional Concepts

Yoshinobu Kitamura, Toshinobu Sano and Riichiro Mizoguchi

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

Abstract. This article discusses automatic identifications of functional structures of artifacts from given behavioral models of components and their connection information (called functional understanding). We propose an ontology of functional concepts which provides a rich vocabulary representing functions together with clear definitions grounded on behavior. The ontology enables the understanding system to limit the search space at functional level and to screen out meaningless interpretations. Furthermore, the ontology includes a new category of functional concepts named meta-function representing conceptual categories of relationship between functions. It plays a crucial role in consolidation of functions to give criteria of grouping functions, that is, identity of consolidated functions. It enables the understanding system to generate such functional hierarchies that do not correspond to physical structure.

1. Introduction

Functionality of artifacts represents a part of the design rationale, while structure and behavior do not show it [1,2]. Thus, a lot of research has been carried out on functional representation of artifacts such as [3,4,5,6,7,8]. A functional structure of an artifact generally consists of functions of components and a whole-part (aggregation) hierarchy of functions which correspond to a function of the whole system or those of subsystems [5,9]. It is essential for redesign of an existing artifact to understand its functional structure in order to consider the intention of the original design [9,10]. Moreover, the functional hierarchy is useful to diagnose artifacts efficiently [11,12].

Our goal here is to identify functional structures automatically from given behavioral models of components and their connection information, called *functional understanding task*. We focus on two problems here. One is how to limit the search space at the functional level, because human uses a large number of verbs representing functions (we call them *functional concepts*) without their operational definitions as discussed in Value Engineering research [13,14]. Nevertheless, almost all of functional models such as those in [1,4,6] are specific to the target system, and thus only a few generic functional concepts have been proposed [5,7,9]. We need a rich vocabulary of functional concepts with operational definitions in order to limit the search space and give constraints on the functional structures.

The other problem is identity of functions in the functional hierarchy. As pointed out in [5,15], the identity of the component from the viewpoint of function in the functional hierarchies is different from that from the structural (or topological) viewpoint. Then, when the understanding system consolidates (aggregates) functions of components into a super-function and then generates a functional hierarchy, the identification of functional groups of the given structural components is one of the crucial issues. Nevertheless, although functions of components [3,4,5,7] and causal relations among components at behavioral level [6,7] have been investigated, little is known concerning the relationship between functions. This is one of the reasons why the conventional functional understanding systems [3,12,15,16] generate such functional hierarchies that correspond to structure and topology. We need sophisticated conceptualization (categories) at the functional level of causal relations in order to give criteria for grouping functions and identity of consolidated functions in functional hierarchy.

We have been tackling these issues on the basis of Ontological Engineering [17], aiming at explicit specification of conceptualization of functional concepts. We identify an ontology of functional concepts of artifacts, which provides a rich vocabulary for functional representation. The ontology plays a role to limit the search space in functional understanding and to screen out meaningless functional interpretations. The ontology includes a new category of functional concepts named meta-function in order to represent conceptual categories of relationship (interdependence) between functions of components. It plays a crucial role in consolidation of functions to give criteria of grouping functions.

In this article we firstly overview the ontology of functional concepts. Section 3 describes the process of functional understanding. The contribution of this work by comparison with the related work is also discussed.

2. Ontology of Functional Concepts

The ontology of the functional concepts is designed to provide a rich and comprehensive vocabulary for both human and knowledge-based systems. It consists of the four spaces as shown in Fig. 1.

2.1 Base-functions

A base-function of a component is defined as a result of interpretation of a behavior of the component under an intended goal [8]. Fig. 1a shows the energy-related 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) [8]. 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

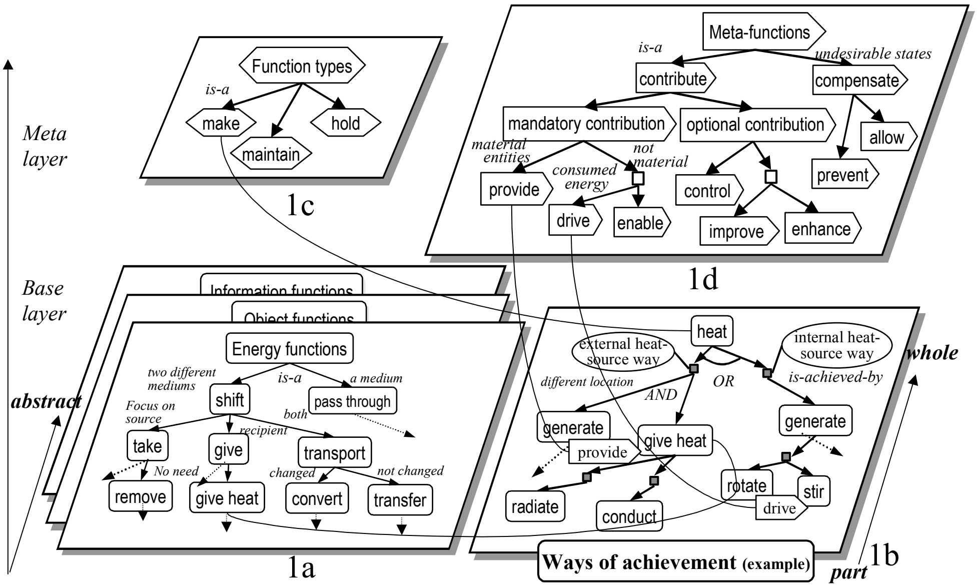

Fig. 1. The ontology of functional concepts (part)

source medium of the transfer" (functional toppings). The definition of "to remove" as a its specialized function is that of "to take" plus "the heat is unnecessary". Note that such definition using FTs is highly independent of its realization, that is, the details of behavior and internal structure of the component.

The ways of achievement represent such realization at the functional level, that is, "is-achieved-by" (whole-part) relation between the base-functions so-called functional decomposition [4,9,18]. We also explicated the background knowledge of the functional decomposition such as the physical law and the intended phenomena (we call it *a way of achievement*). Fig. 1b shows some ways of achievement of "to heat an object" in OR relationship, which are described in terms of concepts in other three spaces. For example, the external heat-source way implies a feature of structure; the location of heat generation is different from the target object.

2.2 Function Types and Meta-functions

The function types represent the types of goal achieved by the function [5]. Keuneke proposes some function types including "ToPrevent" which represents to "keep a system out of an undesirable state of objects" [5]. However, because it focuses on changes of objects associated with the component, the objective of the function is implicit, that is, another function would be affected by the state. Therefore, we redefined the function type as "ToMake", "ToMaintain", and "ToHold"[19] and redefined "ToPrevent" as a kind of a meta-function as below.

The meta-functions (denoted by *mf*) represent a role of a base function called an *agent function* (f_a) for another base function called a *target function* (f_t) [20]. A meta-function is concerned not with changes of objects of these components but with functions of the components, while other two kinds of functional concepts are concerned with existence or changes of objects. We have defined the eight types of meta-functions as shown in Fig. 1d (an is-a hierarchy). We begin definition of meta-functions with the condition where there is a causal relation from the focused parameter of f_a to that of f_t. If the goal of f_t is not satisfied when f_a is not achieved, the f_a is said to have a *mandatory contribution* for the f_t. Although we can intuitively say that f_a has a ToEnable meta-function for f_t in such a case, the authors define a narrower meaning of ToEnable by excepting the cases of ToProvide and ToDrive as follows.

Firstly, when a function f_a generates such an object (or energy) that will be a part of the focused entity of f_t (called *material*), the function is said to perform a meta-function "to provide material" for f_t. When a function f_a generates or transfers such an energy that intentionally consumed by f_t (called *driving energy*), the function is said to have the meta-function "to drive f_t". Lastly, ToEnable meta-function is used for changing a necessary condition for f_t excepting the cases of ToProvide and ToDrive. What we mean by this weak definition is that the conditions such as the existence of the material and that of the driving energy are too obvious to be said to enable a function.

Furthermore, a function f_a having positive effects on the undesirable side effect of a function f_{t1} is said to have a meta-function "to allow the side-effects of f_{t1}". On the other hand, if a serious trouble (e.g., faults) is caused in a function f_{t2} when a function f_a is not achieved, the function f_a is said to have a meta-function "to prevent malfunction of f_{t2}". The details of definitions and examples are shown in [20].

2.3 Application Domains and Assumptions of the Ontology

Up to now we have defined about one hundred and ten base-functions, three function types, eight meta-functions, and about one hundred ways of thirty base-functions. We do not claim completeness of the set of concepts. Note that we define precisely the meaning of concepts for discrimination. The definitions may be narrower than those we use in natural language, because we tend to use them confusingly. The ontology is applied to modeling of a power plant, an oil refinery, a chemical plant, and manufacturing processes [20]. The models in the all applications share many functional concepts except those specific to the chemical domain such as "react". Currently, our ontology assumes the existence of something flowing (or transferred) among components which carries energy (called objects) on the basis of the device ontology. Then, it covers functions in fluid-related plants and does not cover mechanical phenomena. An investigation on functional concepts in different domains is in progress.

3. Functional Understanding

The functional understanding problem is to identify functional structures of an artifact from the given behavioral models of components and connection information. The process of understanding shown in Fig. 2 consists of the following three steps; behavior-function mapping, identification of meta-functions among base-functions, and consolidation of functions to build functional hierarchies as discussed below.

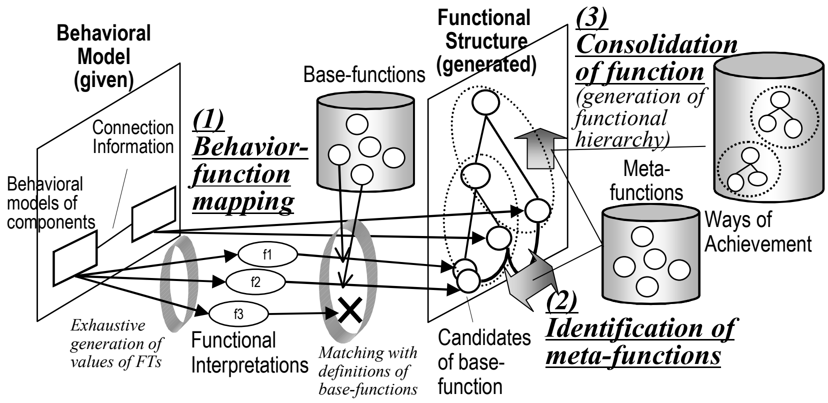

Fig. 2. Three steps of functional understanding

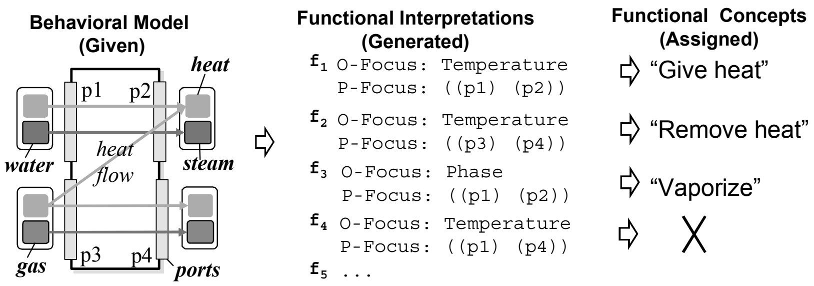

Fig. 3. Behavior-function mapping of a boiler

3.1 Behavior-Function Mapping

Firstly, the understanding system exhaustively generates candidates of base-functions to be performed by each component context-independently. It is enabled by FTs, because FTs can specify mapping from behavior to function and possible values of each FT for a behavioral model are limited. For example, in the case of the boiler shown in Fig. 3, the system generates a functional interpretation f_3 which consists of O-Focus on the "phase" parameter and P-Focus on the inlet water and the outlet steam.

Then, the understanding system screens out meaningless ones by matching them with the base-functions in the ontology. Such functional interpretations that match with no concept in the ontology are screened out as a meaningless interpretation assuming the completeness of the ontology in the functional space. In Fig. 3, the functional interpretation f_3 is successfully matched with a functional concept "vaporize". In contrast, f_4 is screened out as a meaningless interpretation. Although many candidates of the functional interpretations remain, plausible functional interpretations are identified by the following steps.

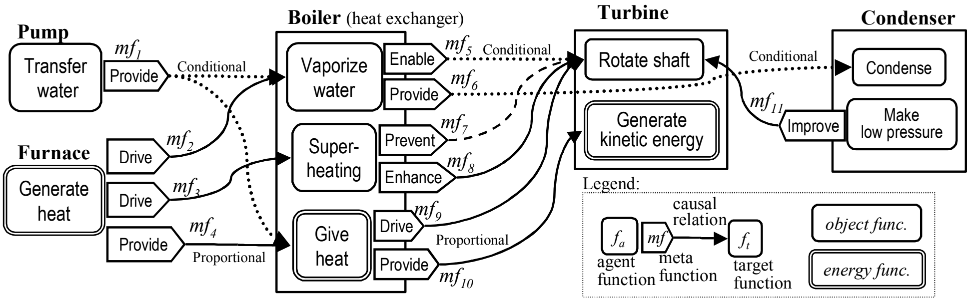

Fig. 4. Identified meta-functions in a power plant (part)

3.2 Identifying Meta-functions

Secondly, the understanding system identifies meta-functions between a pair of generated base-functions using a qualitative reasoning engine for checking causal relations and a diagnostic engine for predicting unintended phenomena. The identification algorithm is described in [20]. For example, imagine that we are given ”to generate heat” function of the furnace and “to vaporize water” function of the boiler in Fig. 4. Firstly, causal relations between the functions are checked. Because there is a mandatory causal relation from the focused parameter of the heat generation (the amount of the heat energy of the combustion gas) to the focused parameter of the vaporization (the amount of the steam), then the heat generation is the agent function and the vaporization is the target function. Next, the conditions of meta-functions are checked. Because the heat energy generated by the furnace is not material (part of) the steam but is consumed by the boiler (i.e., the amount of the energy is reduced) for generating the steam, the meta-function between them is ToDrive (mf_2 in Fig. 4).

On the other hand, because the steam of which phase is gas is a necessary condition of the “to rotate” function of 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 a meta-function ToEnable (see mf_5).

According to identified meta-functions, the understanding system deletes such meaningless functional interpretations that do not contribute to any others. In the example, the functional interpretation f_2 “remove heat” of the boiler shown in Fig.3 is deleted. It represents reduction of heat energy of the combustion gas, which is meaningless in the power plant.

3.3 Consolidation of Base-functions

Lastly, the base-functions generated in the behavior-function mapping are consolidated (aggregated) into super-functions (as a function of a subsystem or the whole system). Then, functional hierarchies are generated basically in a bottom-up manner. If the function of the whole system is not given, some whole functions could be inferred. Such top-most functions that do not have effects to the outside of the system, however, can be rejected according to the assumption of goals of artifacts.

Table 1. Heuristics for generating functional hierarchies

A: Functional concepts heuristics (mandatory)

- ***A1**:Super-function heuristic.* Given a viewpoint for recognition, there always exists a super-function for a functional group.
- ***A2**:Causal relation conservation heuristic*. The causal relations among parameters are conserved in generating functional hierarchies.

B: Preference heuristics (mandatory)

- ***B1**:Serial heuristic*. In serial functions, the system can consolidate functions in the head of chains.
- ***B2**:Simultaneous heuristic*. In functional groups which have parallel-type relations, the system can firstly consolidate simultaneous functions.
- ***B3**:Causal relations heuristic.* The super-function made from sub-functions which have many causal relations is preferred.
- ***B4:** Meta-function preference heuristics. (B4a)* ToDrive represents more cohesive relation than that of ToProvide to any f_t. *(B4b)*ToEnable and ToPrevent are preferred because they are more specific than others.

H: Hierarchical knowledge heuristics (alternative)

- ***H1:** Ways of achievement heuristic.* Functional hierarchies are generated according to the predefined knowledge of ways of achievement.
- ***H2**: Meta-functions heuristic.* Functional hierarchies are generated according to the meta-functions among base-functions. The main function according to meta-functions is interpreted as a super-function.

X: Preference heuristics (optional)

- ***X1**:Parallel-first heuristic.* Such functional groups that have parallel-type relations are preferred.
- ***X2**:Causal-relations-first heuristic.* Such functional groups that have causal relations are preferred.
- ***X3**:Coverage-first heuristic.* Such functional groups that have many functions are preferred.

Y: Grouping heuristics (optional)

- ***Y1**:Structural groups heuristic.* The component of the functions in a functional group should be the same.
- ***Y2**:Energy-groups heuristic.* The energy which the function focuses on should be the same.
- ***Y3**:Medium-groups heuristic*. The medium which the function focuses on should be the same.
- ***Y4**:Attribute-groups heuristic*. The type of functional parameters should be the same.
- ***Y5**:Meta-function-groups heuristic*. The groups are made according to meta-functions.

As discussed in the introduction, the crucial issues are the grouping of the base-functions and selecting super-functions from the candidates. Our approach is based on heuristics and meta-functions. We have identified 16 heuristics shown in Table 1 for grouping of functions (category Y. We call them *grouping heuristics*), knowledge source of super-functions (category H. *hierarchical knowledge heuristics*) and selecting a super-function from candidates (category A, B and X. *preference heuristics*).

Application of the heuristics in category H, X and Y can be specified by users, which enables the system to generate various functional hierarchies. The heuristics in the category H specify the knowledge for generating super-functions, that is, either the ways of achievement shown in Fig. 1b (*H1*) or the meta-functions among base-functions as discussed later (*H2*). The user can select a kind of knowledge alternatively. The heuristics in the category X determine preferences of groups of functions, that is, which groups of functions should be firstly consolidated into a super-function. The users can specify the order of applying the heuristics (or not apply the heuristics). For example, when a user specifies that the *X1*:parallel-first heuristic are preferred than the *X2*:causal-relations-first heuristic (denoted by *X1>X2*), functions in parallel-type relations are firstly consolidated, and then those in causal-type relations are consolidated. It means that additional functions will be integrated into the hierarchy.

The heuristics in the category Y specify the condition for grouping the functions. The users can specify the order of relaxing them (or not apply the heuristics). The understanding system firstly makes the groups of given base-functions according to the all criteria of the category Y specified by users, and then consolidates them into

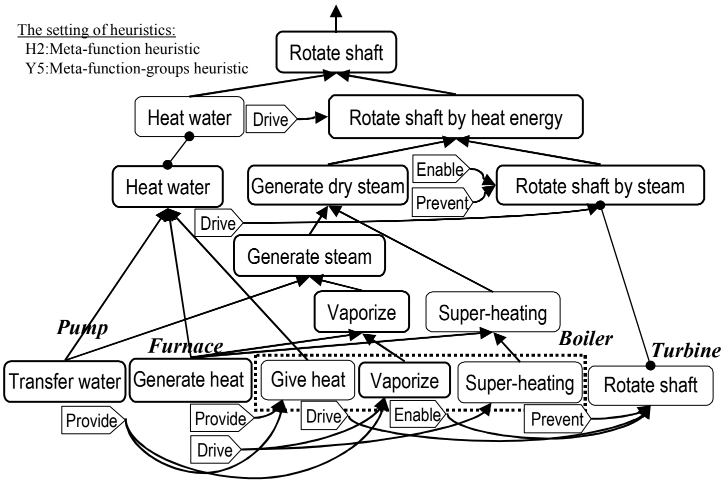

Fig. 5. A functional hierarchy generated according to meta-functions

super-functions according to the user-specified heuristics in H and X and all heuristics in A and B. When there is no functional group to be interpreted, one of the heuristics in the category Y is relaxed according to the specified order. Then new functional groups are made, and then functions in them are consolidated.

On the other hand, those in category A and B represent working assumptions of the system and thus are always applied. For example, *B1*:serial heuristic reflects humans understanding way based on the temporal order. *B3*:causal relations heuristic represents a preference of super-functions supported by many causal relations.

Meta-functions play a crucial role in consolidation. Firstly, because each type of meta-functions has own *strength* to make the functional groups, the grouping and selecting can be done according to the types of the meta-functions among them as well as causal relations and structural relations (e.g., serial, parallel, and simultaneous). It is implemented as *Y5* heuristic in Table 1 for generating groups according to meta-functions and *B4* heuristic for giving strength (i.e., preference) of each meta-functions for the case that some meta-functions contribute to the same target function.

Meta-functions also indicate a main function in the functional group which other functions contribute to. Because the whole function of a functional group can be equal to such a main function, the understanding system can generate a super-function which is equal to the main function (although the target objects of functions are different). According to *H2* heuristic representing this, the super-functions can be generated without the predefined aggregation patterns of functions such as [3,16,18] and our ways of achievements shown in Fig. 1b in the case of *H1* heuristic.

3.4 Examples of the Consolidation

Fig. 5 shows an example of the functional hierarchy of the power plant shown in Fig. 4, which is generated according to meta-functions among base-functions (the heuristics setting is *H2* and *Y5*). Firstly, "to generate heat" and "to vaporize" having To-Drive meta-function are consolidated into a super-function "to vaporize water", before

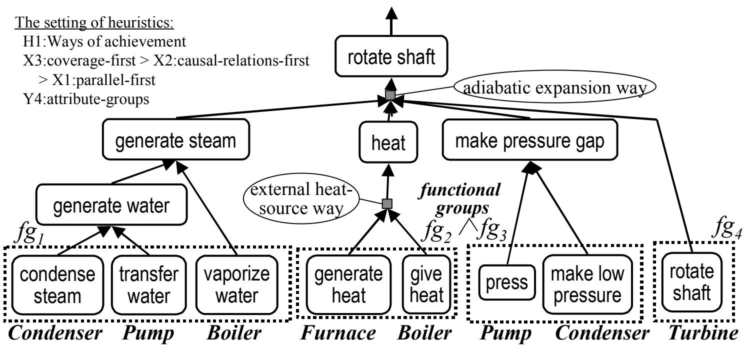

Fig. 6. Another functional hierarchy generated using ways of achievement

the pair of "transfer water" and "to vaporize" having ToProvide is consolidated according to *B4a* heuristic. In the case of "to rotate shaft" which is the target function of three meta-functions, "to generate dry steam" is interpreted earlier than "to heat water" because ToEnable and ToPrevent are more specific than others (*B4b* heuristic).

Fig. 6 shows another example according to another heuristics setting without using meta-functions. In this case, the super-functions are generated according to the knowledge base of ways of achievement of function shown in Fig. 1b. Firstly, the system makes the functional groups such as the functions changing pressure-type parameters (fg_3 in Fig. 6) according to the specified grouping heuristics *Y4*, that is, groups made by kinds of parameters. Next, in the functional group fg_2, the external heat-source way of "to heat" in Fig. 1b matches "to generate heat" and "to give heat", then they are consolidated into a super-function "to heat". After functions in other functional groups are consolidated into each super-function, the groups are relaxed and then these consolidated functions are consolidated into a super-function "to rotate shaft" according to the adiabatic expansion way.

When the user changes the order of applying the heuristics or relaxing, the different functional hierarchies are generated. The user's specification of heuristics can be viewed as a viewpoint for recognition of the target system, and the generated hierarchy reflects the viewpoint. These functional hierarchies are very different from each other. While the first one (Fig. 5) represents how to obtain the driving energy and how to convert the heat energy to kinetic energy, the second one (Fig. 6) represents conditions for kinds of parameters. Some other hierarchies are shown in [19].

4. Related Work and Discussion

Ontology of Functional Concepts

Some sets of "primitives of behavior" are proposed in [7,9,12,21]. We added more intention-rich concepts such as "remove" with unnecessary intention and organized in is-a and part-of hierarchy. In Value Engineering research [13], standard sets of verbs (i.e., functional concepts) for value analysis of artifacts are proposed [14]. There is, however, no machine understandable definition of concepts.

We also identify a new category called a meta-function. The CPD in CFRL [6] represents causal relations among functions. Lind categorizes such relations into Connection, Condition and Achieve [7]. The meta-functions are results of interpretation of such causal relations between functions under the role of the agent function for the target functions without mention of the objects associated with components.

Functional Understanding

The teleological analysis in the de Kleer's work [3] identifies "function" of devices from results of qualitative simulation (i.e., behavior), which is a pioneer work of functional understanding task. Function is, however, defined (and identified) as a causal direction of parameters in his work, while our functional understanding can identify intention-rich concepts in the is-a hierarchy. Moreover, his process of aggregation (called "parsing") is done by some substitution rules according to the topology of the circuits. We decompose it into two phases, that is, identification of meta-functions and consolidation of functions according to them. Meta-functions are detached from the topology and then functional hierarchies which do not correspond to the structure can be generated. In summary, ontological consideration is premature in this work.

The functional understanding based on FR [16] uses templates of CPDs representing functional hierarchies as behavioral causal relations. Thus, functional hierarchies are directly generated from the behavioral model without the functional concepts. They are also limited to those associated with structure. We detached interpretation of function of components from the hierarchical (aggregate) abstraction. Price et al. discuss the interpretation of behavior with functional labels [15]. It corresponds only to the behavior-function mapping.

Furthermore, without the predefined aggregate patterns such as the substitution rules [3] and templates of CPDs [16], our system can generate functional hierarchies according to meta-functions among functions (see Fig. 5) as well as using the predefined general pattern knowledge called the ways of achievements (Fig. 6).

The consolidation theory [22] tries to capture the general rationales of consolidation of components. While we share the goal, its consolidation rules are simple and based on topological relations (e.g., series and parallel) between the limited behavioral primitives. Automatic aggregation in [12] also treats such topological aggregation. We try to explicate the identity of consolidated (aggregated) function as not only such topological relations but also interdependency between functions as meta-functions.

5. Summary

We proposed an ontology of functional concepts including the meta-functions, which contributes to solving the issues of functional understanding task mentioned in Introduction, that is, how to limit the reasoning space and how to identify functions in functional hierarchies. For the first issue, the ontology provides such primitives that are targets in the behavior-function mapping and screens out meaningless interpretations. For the second issue, the meta-function gives identity of functions in the hierarchies and then it enables the system to consolidate (sub-)functions into super-functions as functional hierarchies based on heuristics and meta-functions without the predefined patterns for aggregation. Furthermore, application of the heuristics can be specified by users, which enables the system to generate various functional hierarchies. An investigation on limitation of the ontology mentioned in Section 2.3 is in progress.

Acknowledgments. The authors are grateful to Mitsuru Ikeda for his valuable comments. This research is supported in part by the Japan Society for the Promotion of Science (JSPS-RFTF97P00701).

References

1. Chandrasekaran, B.; Goel, A. K.; and Iwasaki, Y. 1993. Functional representation as design rationale. *COMPUTER*, 48-56.
2. Lee, J. 1997. Design rationale systems: understanding the issues. *IEEE Expert*, 12(3):78-85.
3. de Kleer, J. 1984. How circuits work, *Artificial Intelligence* 24:205-280.
4. Umeda, Y. *et al.*, 1990. Function, behavior, and structure. *AI in Engineering*, 177-193, 1990.
5. Keuneke, A. M. 1991. A. device representation: the significance of functional knowledge. *IEEE Expert*, 24:22-25.
6. Vescovi, M.; Iwasaki, Y.; Fikes, R.; and Chandrasekaran, B. 1993. CFRL: A language for specifying the causal functionality of engineered devices. In *Proc. of AAAI-93*, 626-633.
7. Lind, M. 1994. Modeling goals and functions of complex industrial plants. *Applied Artificial Intelligence*, 8:259-283.
8. Sasajima, M.; Kitamura, Y.; Ikeda, M.; and Mizoguchi, R. 1995. FBRL: A Function and Behavior Representation Language. In *Proc. of IJCAI-95*, 1830-1836.
9 Pahl, G. and Beitz, W. 1998. "Engineering design - a systematic approach", The design council.
10. Goel, A., and Chandrasekaran, B. 1989. Functional Representation of Designs and Redesign Problem Solving. *Proc. of IJCAI-89,* 1388-1394
11. Larsoon, J. E.1996. Diagnosis based on Explicit Means-ends Models, *Artificial Intelligence*, Vol.80, pp.29-93.
12. Chittaro, L., Ranon, R., 1999. Automatic derivation of hierarchical representation for flow-based functional models, *Proc. of Tenth International Workshop on Principles of Diagnosis (DX-99)*, pp.45-50.
13. Miles, L. D. 1961. *Techniques of value analysis and engineering*. McGraw-hill.
14. Tejima, N. *et al.* eds. 1981. *Selection of functional terms and categorization*. Report 49, Soc. of Japanese Value Engineering (In Japanese).
15. Snooke, N. and Price, C. 1997, Hierarchical Functional Reasoning. In *Proc. of IJCAI-97 Workshop on Modeling and Reasoning about Function*, 11-22.
16. Thadani, S. and Chandrasekaran, B. 1994, Constructing Functional Models of a Device from its Structural Description. In *Working papers of QR-94*, 276-285.
17 .Mizoguchi, R., and Ikeda, M. 1997. Towards ontology engineering. In *Proc. of PACES/SPICIS '97,* 259-266.
18. Bhatta, S. R., and Goel, A. K. 1997. A functional theory of design patterns. In *Proc. of IJCAI-97*, 294-300.
19. Kitamura, Y., and Mizoguchi, R. 1998. Functional ontology for functional understanding, Papers of *Twelfth International Workshop on Qualitative Reasoning (QR-98)*, 77-87.
20. Kitamura, Y., and Mizoguchi, R. 1999. Meta-functions of artifacts, Papers of *13th International Workshop on Qualitative Reasoning (QR-99)*, 136-145.
21. Hodges, J. 1992. Naive mechanics - a computational model of device use and function in design improvisation. *IEEE Expert* 7(1):14-27.
22. Bylander, T., and Chandrasekaran, B. 1985. Understanding behavior using consolidation, *Proc. of IJCAI-85*, 450-454.