Functional Understanding from Structure and Behavior based on a Functional Ontology

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

Abstract. This paper proposes a framework of *functional understanding* for redesign problem solving, which identifies conceptual functional structures of an artifact from its behavioral model. We argue that *a functional ontology* plays a crucial role in functional understanding. Our functional modeling language FBRL[12] enables us to build such an ontology in terms of its mapping primitives between behavior and function called FT. The ontology provides a basis of functional reasoning at the comprehensible conceptual level. It enables the functional understanding system to bridge a gap between physical behavioral level and the conceptual functional level using FTs (called behavior-function mapping) and then to generate functional hierarchies at the conceptual level (functional hierarchy understanding). An example of functional understanding of a power plant is demonstrated.

1 Introduction

The redesign problem is one of the major types of design problems. Given a structural and behavioral model of an existing artifact, the redesign problem solver proposes suitable modifications for satisfying the new requirements. Recently, the importance of capturing *function* of the artifacts as well as *behavior* has been pointed out[1, 4, 5, 9]. *Functional concepts* representing functions of components play a crucial role in the redesign process because they enable the redesign system to understand the requirements and to specify the search space at the functional level which is compatible with the terms used in them. The above observation indicates the redesign task requires a *functional understanding* task that bridges a gap between given structural and behavioral models and functional concepts. Therefore, the redesign task consists of *functional understanding* and *functional reasoning* as shown in Figure 1a. The former is to identify conceptual functional structures from given behavioral models of components used in the target system with structural information among components. The later is reasoning at the functional level to modify the generated conceptual functional structures behavior-independently to fulfil the given requirements. Lastly, such a

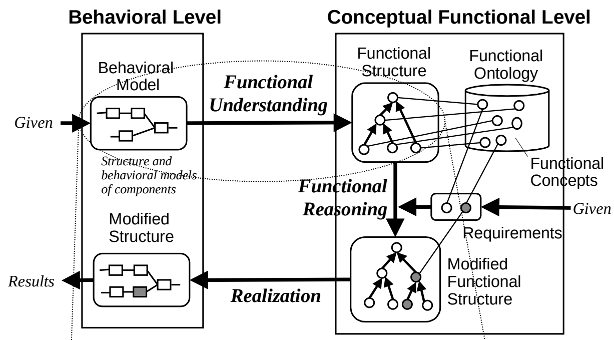

Figure 1a. Redesign based on functional understanding

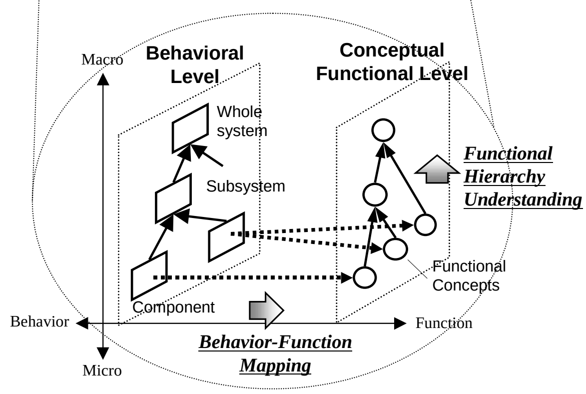

Figure 1b. Functional understanding

structure that can realize the functional structures is generated.

In this article, we discuss a framework of functional understanding for the redesign problem solving. It can be decomposed into *behavior-function mapping* and *functional hierarchy understanding* as shown in Figure 1b. The former is a mapping from a behavioral model to generic functional concepts discussed below. The later is to identify hierarchical “part-of” relations among the functional concepts in the mapping.

Such functional understanding requires a sophisticated conceptualization of functions of artifacts, that is, a *functional ontology*. It consists of *functional concepts*, which are conceptual vocabulary for specifying reasoning space at the functional level as well as for representing functions of components in a specific target domain. They should satisfy the following requirements.

- **Mapping primitives:** They should be explicitly defined using primitives for mapping from behavioral to functional spaces.
- **Independence of implementation:** They should be independent of its implementation, that is, how it is achieved, in order to make them applicable to various systems in the target domain.
- **Abstraction by role:** It should be defined as abstraction of behavior not by grain size but by role.
- **Comprehensiveness and comprehensibility:** Ideally, all functional concepts recognized by human in the target domain should be defined.

The functional ontology provides a basis both for behavior-function mapping and for functional hierarchy understanding. In the behavior-function mapping, it enables enumeration of plausible functional concepts by specifying and limiting the search space. It also enables the hierarchy understanding system to generate functional hierarchies according to some general heuristics in terms of functional concepts.

A lot of functional modeling languages have been proposed for design or diagnosis [2, 8, 10, 12, 13, 15, 17]. Many of them, however, do not try to describe such generic functional models or a functional ontology. For example, in FR[13] and CFRL[17], function is defined as a hierarchical abstraction of behavior, which is different from ours shown in Figure 1b and discussed bellow. Their functional models are described as sequences of partial states of the behavior, so that they depend on their implementations. Thus, the generic definitions of functional concepts cannot be described using FR and CFRL.

Some researchers such as Chandrasekaran[2], Lind[8] and Price[10] share the importance of generic functions similar to those argued here. However, no rich functional ontology satisfying the requirements mentioned above has been built yet. One of its crucial reasons is that they did not discuss the mapping primitives from behavior to function for an explicit definition of a function.

We have proposed a functional modeling language named FBRL[12], which enables us to build such a functional ontology. FBRL introduces *functional toppings* (FTs) which are a set of primitives for mapping from behavioral and functional spaces. As shown in this article, they enable us to define generic functional concepts (see Figure 4). For example, the definition of the function "give heat" consists of "shifting heat energy" (a behavioral condition) and "focus on the medium receiving the heat" (a functional topping). Such explicit definitions are independent of their implementation, and hence they are applicable to various components. Therefore, we can describe general functional knowledge (see Figure 7).

In this article, we propose a framework of a functional understanding system based on FBRL. First, we discuss characteristics of functional models and then define the functional understanding task in Section 2. In Section 3, two knowledge bases for functional understanding, a functional ontology and functional hierarchy patterns, are shown. In section 4, we decompose the functional understanding task into three subtasks. Section 5 demonstrates the functional understanding of a power plant as an example. Section 6 discusses related work.

2 Functional Understanding

In this section, we discuss the relation between given behaviors and functions to be generated and then define the functional understanding task. As Figure 1b outlines, there are two axes for functional compositions of artifacts. The horizontal axis represents *role* intended by the designer and *interpretation* by the understanding system. There are functional models between the behavioral level and the conceptual functional level (not shown in the figure). They are instances of functional concepts. The functional models and functional concepts explain "what is a role" of each behavior. The vertical axis represents *grain sizes* of representation and explains "how it contributes to the whole system".

The behavior can be defined as temporal changes of parameter values. The model for generating the behavior consists of the behavioral models of components and structural information representing connections among components.

On the other hand, a function is defined as a result of *interpretation* of the behavior under the intended goal[12]. Although function implies partial states representing the goal (as described in [2, 10, 15]), a functional model is not composed of only states. For example, the discrimination between the heater's function "give heat" and the radiator's function "remove heat" requires the information whether the heat is needed for the system or not. Therefore, it is needed to identify such mapping primitives between behavior and function that distinguish various functions.

Figure 3 shows the scheme of the FBRL model of a component. FBRL introduces four primitives for specifying the mapping from behavior to function, called functional top-

```
Component:
  ModeChangeCond:     /* Condition for a behavioral mode */
  Mode1
    Behavior
      Objects:        /* Objects input to or output from the component */
      SubComponents:  /* Names of subcomponents */
      MP-Relations:   /* Relations among input materials and output product
      SameClass:      /* Sameness of the class of in/out objects */
      InherentParams: /* Parameters inherent in the component */
      Ports:          /* Connection with neighbor component */
      QN-Relations:   /* Qualitative relations among parameters */
    FT-Sets:
    FT-Set1
      FuncType:       /* Function type: ToMake, ToMaintain, ToHold */
      O-Focus:        /* Focus on a certain class of object */
      P-Focus:        /* Focus on a certain input-output relation */
      Necessity:      /* Necessity of output objects */
    FT-Set2
      .....
  Mode2
    Behavior
      ......
    FT-Sets:
      ......
```

Figure 3. The Scheme of FBRL Component Model

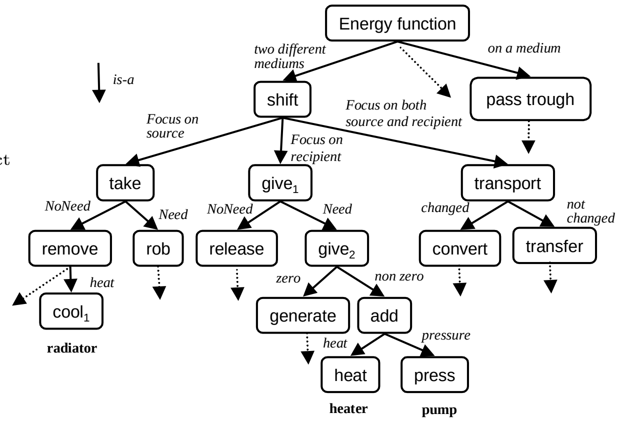

Figure 4. A part of the Functional Ontology

pings (FTs). There are four types of FTs; (1)O-Focus representing focus on attributes of objects, (2)P-Focus representing focus on ports (interaction to neighboring components), (3)FuncType representing types of contribution (extended one of those defined by Keuneke[6]), and (4)Necessity of objects. For more details of the FBRL modeling, see [12].

Each functional model in FBRL corresponds to a *functional concept*. A functional concept has a conceptual term representing the function. Such terms enable the various concrete functional models to match with the generic functional knowledge discussed below. Moreover, they provide comprehensible explanations of the target artifacts[12].

The functional hierarchies represent *functional-part-of* relations among functions in different grain sizes. Although they in many cases correspond to the structural hierarchies (i.e., system-subsystem-component relations) on the behavioral level as discussed in [14], it is not always the case. The "structure" of target systems is one of the *viewpoints* for generating functional hierarchies. A target system can be functionally understood as some different functional hierarchies according to the different viewpoints.

Moreover, there are relations among functions in the same grain size. We call them *functional relations*. In FBRL, they are categorized into two types, *causal-type* functional relations defined by causal relations in behavior, and *structural-type* relations defined by structure. The former has subtypes such as *providing, preventing* and *efficiency-type*. The later has subtypes such as *series, parallel, sequential* and *simultaneous*.

The functional understanding task is to identify function models, functional concepts, functional relations, and functional hierarchies from the given behavior models at the finest grain size and the behavior. The behavior is generated by a

```
(define-function-class take-energy (?fd)
   :label ("take energy")
   :subtype-of ('shift-energy)
   :exists (?f ?b ?obj0 ?obj1 ?obj2 ?pin ?pout)
   :e-def
     (and (function-part ?fd ?f) (behavior-part ?fd ?b)
       (has-energy-objects ?b (?obj0 ?obj1 ?obj2))
       (has-ports ?b (?pin) (?pout))
       (exists-in ?obj0 ?pin) (exists-in ?obj1 ?pout))
   :b-def (and (mp-relation ?obj0 ?obj1)
               (mp-relation ?obj0 ?obj2))
   :f-def (and (focus-on-in-port ?f ?pin)
               (focus-on-out-port ?f ?pout)))

(define-function-class remove-energy (?fd)
   :label ("remove energy")
   :subtype-of ('take-energy)
   :exists (?f ?b ?obj ?port ?oc)
   :e-def
     (and (function-part ?fd ?f) (behavior-part ?fd ?b)
       (has-object ?b ?obj) (energy-object ?obj)
       (has-port ?b ?port) (out-port ?port)
       (exists-in ?obj ?port) (is-a ?obj ?oc))
   :b-def ()
   :f-def (and (focus-on-out-port ?f ?port)
            (specify-necessity ?f 'NoNeed ?port ?oc)))
```

Figure 5. Functional Concepts "take" and "remove"

qualitative reasoning system from the behavior models.

3 Knowledge for Functional Understanding

This section describes two knowledge bases crucial for functional understanding. One is the functional ontology and the other is the functional hierarchy patterns in terms of the functional terms in the ontology.

3.1 Functional Ontology

The functional ontology consists of *functional concepts* organized in an *is-a* hierarchy with clews of classification shown in Figure 4. The definition of a functional concept consists of a label representing the concept and conditions of behavior and functional toppings. For example, the definitions of the functional concepts "take" and "remove" are shown in Figure 5. The functional concept "take energy" is defined as "an energy flow between two mediums" (a behavioral condition), and "focus on the medium transferring the heat" (a functional condition). Moreover, the definition of "remove" is that of "take" plus "the heat is unnecessary". Thus, "take" is a general (super) concept of "remove".

The functional ontology provides conceptual vocabulary for describing functional knowledge such as the functional hierarchy patterns described bellow. Such conceptual vocabulary allows the knowledge to be reusable because they are independent of implementation.

3.2 Functional Hierarchy Patterns

A function is achieved by a sequence of sub-functions. A combination of a super-function and its sub-functions is called a *functional hierarchy pattern*. Its definition consists of a super-function, sub-functions, functional relations among sub-functions, and behavioral conditions. These functions are described in terms of the functional concepts. For example, a super-function "heat object" has two sub-functions; "generate heat" and "give heat". There should be a providing-type functional relation among them. The behavioral condition is that the objects receiving the heat are identical. Figure 6 shows its description.

In general, a function has some functional hierarchy patterns achieving it. Figure 7 shows those achieving a function "obtain material". Note that Figure 7 shows *is-achieved-by* relations among the functional concepts, while Figure 4 shows *is-a* relations. The definitions of the functional concepts in Figure 4 are independent of "how it is achieved".

```
(define-functional-hierarchy-pattern
       heat-by-giving-heat-energy (?fhd)
   :label ("heat object")
   :exists (?sfd ?fd0 ?fd1 ?fd2 ?f0 ?f1 ?f2 ?b0 ?b1 ?b2
            ?obj01 ?obj02 ?obj12 ?obj21 ?obj22 ?obj2e)
   :fh-def
      (and (has-super-function-description ?fhd ?fd0)
        (has-sub-function-descriptions ?fhd ?sfd)
        (has-a-sub-function-description ?sfd ?fd1)
        (has-a-sub-function-description ?sfd ?fd2)
        (sub-function-number ?sfd 2) (neq ?fd1 ?fd2)
        (is-a-function ?fd0 'heat)
        (is-a-function ?fd1 'generate-heat-energy)
        (is-a-function ?fd2 'give-heat-energy))
   :e-def
      (and (focus-on-objects ?fd0 ?obj01 ?obj02)
        (focus-on-out-object ?f1 ?obj12)
        (focus-on-objects ?f2 ?obj21 ?obj22)
        ...)
   :b-def (and (obj-connection ?obj12 ?obj2e)
            (is-identical ?obj01 ?obj21)
            (is-identical ?obj02 ?obj22))
   :f-def (providing-functional-relation ?fd1 ?fd2))
```

Figure 6. A Functional Hierarchy Pattern: "heat" = "generate heat" + "give heat" (part)

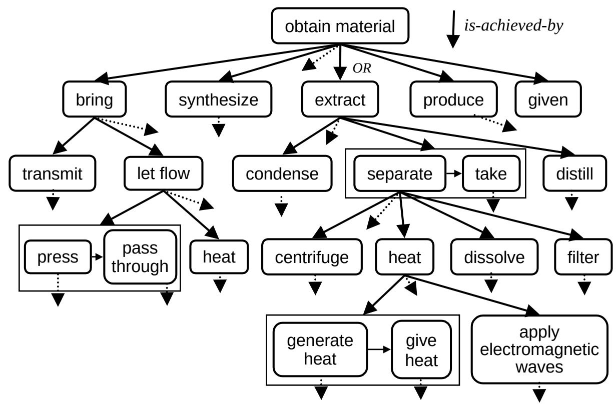

Figure 7. Functional Hierarchy Patterns for "obtain" (Part)

4 Processes of Functional Understanding

The functional understanding task can be decomposed into the following three subtasks as shown in Figure 8. First, given the structure of an artifact and behavioral models of components used, all possible functional interpretations of behavior of each component are generated (called *behavior-function mapping*). Next, the functional relations among generated functional interpretations are identified (called *functional relation understanding*). Lastly, from the functional interpretations and relations among them, functional hierarchies are generated (called *functional hierarchy understanding*). Although many candidates of the functional interpretations are generated by the first step, plausible functional interpretations are identified by the second step and the third step.

4.1 Behavior-function mapping

The behavior-function mapping is decomposed into two subtasks, to generate candidates exhaustively, and to screen out meaningless ones by matching them with functional concepts. Although it is in principle difficult because the search space of function is huge, the mapping primitives, the FTs in FBRL, play a crucial role to limit the search space. Thus, possible candidates can be exhaustively generated as all tuples of possible values of FTs context-independently.

Next, the understanding system tries to match the gener-

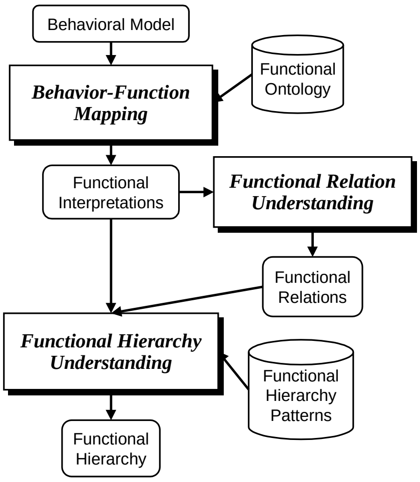

Figure 8. The steps of the functional understanding

ated candidates of functional interpretations with definitions of the functional concepts. If a candidate matches with no functional concept, it is screened out as a meaningless interpretation assuming the completeness of the ontology in the functional space.

4.2 Functional Relation Understanding

The causal-type functional relations are identified according to causal relations in the behavior generated by the qualitative reasoning engines. We use our original one[7] which is categorized as a type of the reasoning method proposed by de Kleer and Brown[3]. On the other hand, structural-type functional relations are identified according to information of connections among components.

If a functional interpretation does not contribute to any other functions, it is screened out because such an interpretation is not plausible.

4.3 Functional Hierarchy Understanding

The functional hierarchy patterns enables the understanding system to generate super-functions from given sub-functions in the bottom-up manner. First, given a set of functions, grouping of given functions is done according to a specific condition. Next, the understanding system searches for such functional hierarchy patterns that match the functions in each group. When the functions satisfy the conditions of the sub-functions in a functional hierarchy pattern, the system instantiates a functional model from the functional concept of the super-function of the pattern. The conditions of sub-functions in terms of the functional concepts can match various functional models. Thus, such knowledge is reusable.

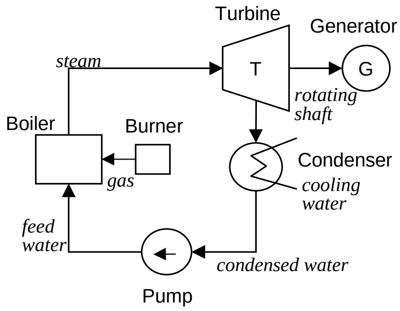

Figure 9. A Power Plant

Because many functional hierarchies can be generated from a set of functions, the knowledge for selecting super-function is needed. We prepare twelve heuristics for it. The heuristics play two roles, (1)to specify the condition for grouping the functions, and (2)to determine preferences of super-functions. Some of them are adjusted by users. It enables the system to generate various functional hierarchies.

5 Example

This section shows an example of the functional understanding of a simple model of a power plant shown in Figure 9.

5.1 Behavior-function mapping

Figure 10 shows the behavior-function mapping of the boiler. The input shown in the left part of the figure consists of inflow objects, outflow objects, connection ports, and behavior constraints such as MP-relations representing material-product relations among objects. Firstly, possible values of FTs are generated exhaustively. For example, a functional interpretation f_3 consists of O-Focus on the "phase" parameter and P-Focus on the inlet water and the outlet steam. Then, the functional interpretation is successfully matched with a functional concept "vaporize". Many possible functional interpretations such as "give heat", "expand" and "remove heat" are generated. Last two concepts cannot be function of the boiler. They will be deleted by the next step.

5.2 Functional Relation Understanding

According to behavioral causal relations generated by a qualitative reasoning engine, functional relations among the functional interpretations are identified. A part of functional relations identified is shown in Figure 11. It enables the understanding system to delete the functional interpretations

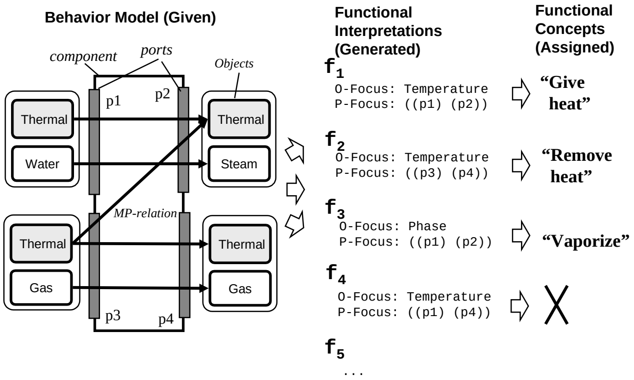

Figure 10. Behavior-function mapping of a Boiler

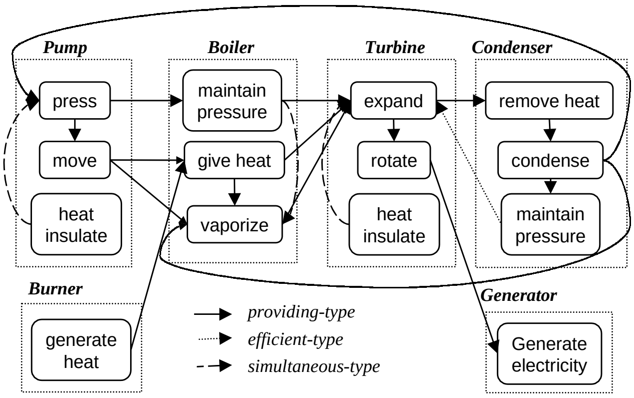

Figure 11. Functional relations of a power plant (part)

"expand" and "remove heat" of the boiler, because they do not contribute to any others.

5.3 Functional Hierarchy Understanding

The functional hierarchies depend on the viewpoints for understanding as discussed in Section 2. We control them by adjusting twelve heuristics as discussed in Section 4.3. Here three examples are shown according to the three different settings of heuristics.

Example 1: In this example, the setting of heuristics is "functional groups made by types of parameters". Thus, the understanding system makes the functional groups such as the functions changing pressure-type parameters as shown in Figure 12. Then, super-functions such as "make pressure gap" are generated.

Example 2: In the case that the settings are "correspondence to the structure" and "complete coverage", the understanding system can generate a function "Rankine cycle" in Figure 13. Many functions such as "adiabatic expansion" correspond to structures such as the turbine.

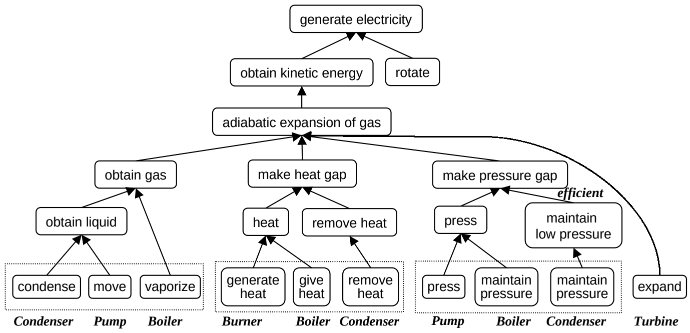

Figure 12. Functional hierarchy of a power plant(1)

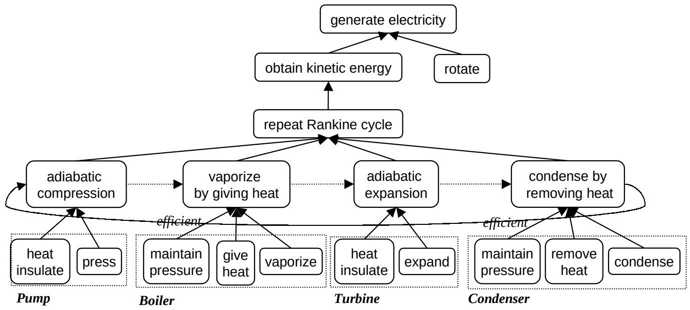

Figure 13. Functional hierarchy of a power plant (2)

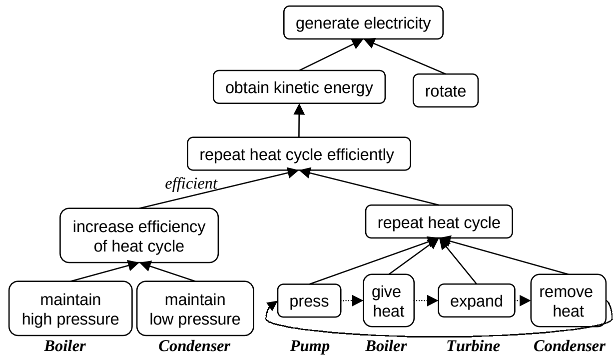

Figure 14. Functional hierarchy of a power plant (3)

Example 3: If the setting of "correspondence to the structure" is omitted, the two major functions "repeat heat cycle" and "increase efficiency of heat cycle" are identified as shown in Figure 14.

6 Related Work

In FR[13] and CFRL[17], a function is described as a sequence of partial states achieving it, called as CPD, on the basis of the different definition of function. A CPD represents a functional hierarchy as behavioral causal relations. The functional understanding based on FR [16] uses templates of CPD. In [16], functional hierarchies are directly generated from the behavioral level without the functional concepts. The functional hierarchies generated in [16] are limited to those associated with structure. In contrast, our two-step decomposition of the task and the functional ontology enable the understanding system to generate various functional hierarchies by reasoning at the conceptual function level according to several viewpoints other than structure.

Recently, Chandrasekaran et al. point out the importance of implementation-independent functional models and propose a representation of functions as effect[2]. The functional models, however, are described by the partial states of behavior. Moreover, they do not try to define generic functional concepts as a basis of functional reasoning.

In [8], few fundamental functions such as “energy storage” are defined. The functional models are, however, described by human modelers.

Price et al. discuss the interpretation of behavior with functional labels[10, 14]. It corresponds only to the behavior-function mapping. Thus, the functional hierarchies in [14] always correspond to given structural hierarchies of the target system.

In [11], Sakao et al. share the goal of this paper with us, that is, the reasoning not at behavior level but at the functional level. In constant with our ontological approach, they discuss general operations of functionality.

7 Summary

We have shown a framework of functional understanding based on FBRL. The main points are as follows;

- **Functional understanding task :** We have defined the functional understanding task as identifying functional concepts and functional hierarchies from given structural and behavioral models. We discussed its characteristics of each of its three subtasks.
- **Functional Ontology :** We presented a functional ontology consisting of functional concepts independent of implementation shown in Figure 4. It enables the functional understanding system to generate meaningful functions from the behavior as shown in Figure 10. Moreover, it provides the vocabulary for describing the functional hierarchy patterns shown in Figure 7.

The functional ontology currently includes the functional concepts in fluid-related systems. It has been successfully applied to an ideal model of a power plant shown here and a concrete chemical plant. The evaluation and extension of the ontology remain as future work. In this article we did not discuss in detail the reasoning control of functional understanding because of the space limitation. We are currently investigating heuristics for controlling functional hierarchy and implementing the understanding system.

Acknowledgment

The authors would like to thank Prof. Mitsuru Ikeda for his valuable comments.

REFERENCES

[1] Chandrasekaran, B., Goel, A.K. and Iwasaki, Y.: Functional representation as design rationale. *Computer*, pp.48-56, January, 1993.

[2] Chandrasekaran, B. and Josephson J.R.: Representing Function as Effect: Assigning Functions to Objects in Context and Out. *AAAI-96 Workshop on Modeling and Reasoning with Function*, 1996.

[3] de Kleer, J., Brown, J. S.: A Qualitative Physics Based on Confluences. *Artificial Intelligence* Vol.24, pp7-83, 1984.

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

[5] Iwasaki, Y. and Chandrasekaran, B.: Design Verification Through Function- and Behavior-oriented representations – bridging the gap between function and behavior –. *Proc. of AI in Design '92*, pp.597-616, 1992.

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

[7] Kitamura,Y. et al.: A method of qualitative reasoning for model-based problem solving and its application to a nuclear plant. *Expert Systems with Application* 10(3/4), pp.441-448, 1996.

[8] Lind, M.: Modeling Goals and Functions of Complex Industrial Plants. *Applied Artificial Intelligence*, 8, pp.259-283, 1994.

[9] Lee, J.: Design rationale systems: understanding the issues. *IEEE Expert*, May/June, pp.78-85, 1997

[10] Price, C. and Pugh, D.: Interpreting Simulation with Functional Labels. *Working papers of QR-96*, pp.198-204, 1996.

[11] Sakao, T., Umeda Y., and Tomiyama T.: Operations of Functionality. *Working papers of QR-97*, pp.327-335 (1997).

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

[13] Sembugamoorthy V. and Chandrasekaran, B.: Functional Representation of Devices and Compilation of Diagnostic Problem-Solving Systems. *In Experience, memory and Reasoning*, pp.47-73, 1986.

[14] Snooke N. and Price C.: Hierarchical Functional Reasoning. *IJCAI-97 Workshop on Modeling and Reasoning about Function*, pp.11-22 (1997).

[15] Umeda, Y. et al. : Function, Behavior, and Structure. *AI in Engineering*, pp.177-193, 1990.

[16] Thadani S., Chandrasekaran, B.: Constructing Functional Models of a Device from its Structural Description. *Working papers of QR-94*, pp.276-285 (1994).

[17] Vescovi, M., Iwasaki, Y., Fikes, R., Chandrasekaran, B.: CFRL: A Language for Specifying the Causal Functionality of Engineered Devices. *Proc. of AAAI-93*, pp.626-633, 1993.