

A Functional Concept Ontology and Its Application to Automatic Identification of Functional Structures^{*1}

Yoshinobu Kitamura^{*2}, Toshinobu Sano^{*3}, Kouji Namba^{*4}, and Riichiro Mizoguchi

The Institute of Scientific and Industrial Research, Osaka University,
8-1, Mihogaoka, Ibaraki, Osaka 567-0047, Japan

Abstract. Functionality of artifacts as well as structure and behavior plays crucial roles in problem-solving such as design. 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 a functional concept ontology which provides a rich vocabulary representing functions together with clear definitions grounded on behavior. The ontology enables the automatic identification system to make the search in the functional space tractable and to screen out meaningless interpretations. Furthermore, the ontology includes a new category of functional concepts named meta-functions representing conceptual categories of interdependency 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 identification system to generate such functional hierarchies that do not correspond to physical structure.

Keywords. Functional modeling and reasoning, Modeling of artifacts, Design Support

1. Introduction

The performance of problem-solving systems about physical systems such as design and diagnosis heavily relies on contents of the model of the target system. In the model-based systems (MBS) based on the qualitative reasoning technique (QR), the model of an artifact usually consists of local models of components and their topological connections (so-called "structure"). The component models include physical attributes of entities and constraints over them representing physical laws or characteristics of the component's behavior (behavioral model). Such models are basically not subjective but objective and are easily obtained from physical laws governing behaviors in the domain and usual design documents.

However, such behavioral and structural models do not represent "intention" of the designer such as "why a component exists there" and "which causal relations are intended". Functionality of artifacts represents a part of such designers' intention so-called design rationale (DR) [1,2]. Thus, a lot of research has been carried out on functional representation of artifacts [3-8]. Although there is no consensus on definition of functionality of artifacts, most functional models represent intended goals or roles of the behavior and then they are dependent on the "context" of the components in contrast with behavioral models independent of context. A functional structure of an artifact generally is represented as a whole-part (aggregation) hierarchy of functions each of which typi-

cally corresponds to a function of a component, a subsystem, or the whole system [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]. In fact, designers usually decompose the required goal of the target system into sub-functions that can be achieved by components according to an appropriate way of achievement among many alternatives, called functional decomposition [9]. Given a functional structure representing such functional selection (the intention of the original design), to adopt another alternative of the way of achievement enables us to change the original functional structure into a different structure with possible drastic improvements. Such redesign, which is our future research goal, requires a functional structure of the given artifact which is the goal of this research. Moreover, the functional hierarchy is useful to diagnose artifacts efficiently [11, 12].

A lot of research has been carried out on capturing design rationales in design research in order to consider such original intention [2]. In almost all of such capturing systems, DRs are hand-written by human designers in ad-hoc manners. In functional representation research, the functional models representing DRs are described by humans for each target system in such manners [1,4,6], because there is neither standard vocabulary nor constraints on contents of the models.

Our goal here is to identify functional structures automatically from given behavioral models of components and their connection information, called *functional understanding task*. Automatic identification of a functional structure of an artifact enables us to redesign the

^{*1} This paper is an extended and revised version of the paper entitled "Functional Understanding based on an Ontology of Functional Concepts" in the proceedings of the Sixth Pacific Rim International Conference on Artificial Intelligence (PRICAI 2000), pp.723-733, 2000.

^{*2} Corresponding author: Phone +81-6-6879-8416, Fax: +81-6-6879-2123, E-mail: kita@ei.sanken.osaka-u.ac.jp

^{*3} Currently, Onkyo Corporation, ^{*4} Currently, Nippon Telegraph and Telephone Corporation

artifact based on functional structures from structural and behavioral information which can be easily obtained. It also provides an operational method for bridging a gap between the behavioral and structural levels and the functional level.

We focus on two problems here. One is how to make the search in the functional space tractable, 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, 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 make the search in the functional space tractable, 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 functional viewpoint in the functional hierarchies is different from that from the structural (or topological) viewpoint. Consider a telephone as the whole target system as an example shown in [5]. It can be decomposed into a handset and a chassis from the structural (or topological) viewpoint. On the other hand, from the functional viewpoint, it can be decomposed into a transmitter, a receiver and a notifier. These elements can be decomposed into structural components, e.g., a microphone and a speaker. Thus, they are sub-systems and are recognized according to not structure but functionality. Then, when a functional structure identification system consolidates (aggregates) functions of components into a super-function to generate a functional hierarchy, the identification of functional groups of the given structural components is one of the crucial issues. Nevertheless, little is known concerning the relationship between functions, though functions of components [3,4,5,7] and causal relations among components at behavioral level [6,7] have been investigated. This is one of the reasons why the conventional automatic identification systems of functional structures [3,12,15,16] generate such functional hierarchies that exactly correspond to the structural hierarchy. 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 the functional hierarchy.

We have been tackling these issues on the basis of Ontological Engineering [17,18], aiming at explicit specification of conceptualization of functional concepts. We designed a functional concept ontology, which provides a rich vocabulary for functional representation. The ontology plays a role to make the search in the functional space tractable and to screen out meaningless functional interpretations. The ontology includes a new type 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 our general framework of functional modeling of artifacts. Section 3 describes the functional concept ontology. Section 4

describes the process of automatic identification of functional structures. The contribution of this work by comparison with the related work is also discussed.

2. Our Approach for Functional Models of Artifacts

This section overviews our approach for functional models of artifacts and framework of them as summarized as shown in Figure 1. This framework is an extended one of that of our functional modeling language FBRL (abbreviation of a Function and Behavior Representation Language) [8]. There are three axes. The horizontal one represents objective-teleological interpretation, that is, from behavioral level (including structure) to functional level. In this axis, the model at the functional level consists of two layers; the base-function layer and the meta-function layer. The vertical axis represents the grain size of the entities. An entity is a part of upper one. In each layer, there is a whole-part (aggregation) hierarchy. The last axis to the depth represents relations among entities of the same grain size.

2.1 Behavioral Level (Structure and Behavior)

The model at the behavioral level is objective without intention of designers. We adopt a device-oriented ontology [18] which is extended one of that in the system engineering and a component-oriented ontology [19]. The device ontology specifies the device-centered view of artifacts which regards any artifact as composition of devices (components) which process input to produce output which is what the users need. The device ontology imposes a frame or viewpoint on physical processes (phenomena) in the physical process ontology [20] to introduce an engineering perspective. That is, it introduces the concepts of a black box equipped with input and output ports. The major difference between the device ontology and the physical process ontology is that while device ontology has an agent, which is considered as something that plays the main actor role in obtaining the output, the physical process ontology does not have such an agent but has participants, which only participate in the phenomena being occurring. Needless to say, such an agent coincides with a device in the device ontology. Such agent role is crucial to recognize functions, because the concept of a function has to be attributed to an agent (a device), which is viewed as a main actor to achieve the function.

A device has input and output ports through which it is connected to another device (as shown in the axis to depth in Figure 1). A device consists of other devices of smaller grain size and usually is organized in a whole-part hierarchy of sub-devices (as shown in the vertical axis).

Something that can be considered as that it goes through a device from the input port to the output port during which it is operated by the device is called an *object*. Examples of an object include substance like fluid, energy like heat, motion, force, information, etc. An object has attributes whose values change over time.

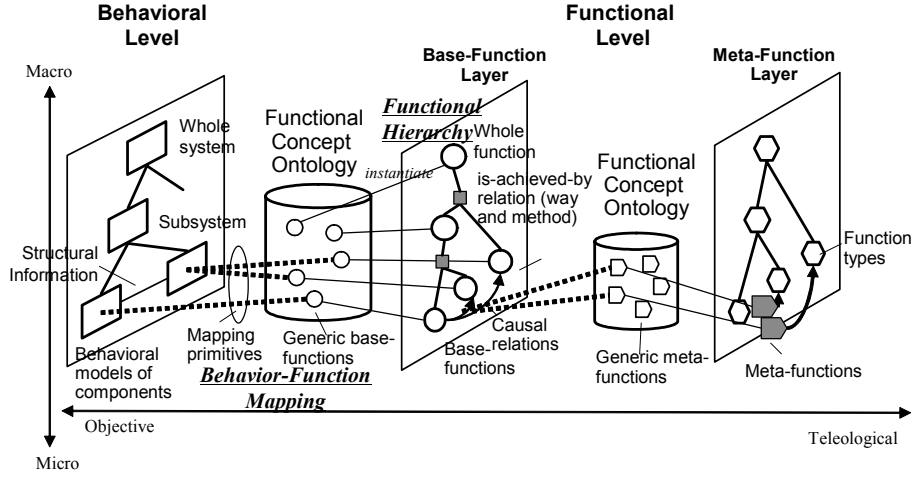


Figure 1: Our Framework of Functional Modeling

The state change of an object is realized by the difference between the states of the object at the input port and that at the output port. The state changes are called as “behavior”.

A behavioral-level model of a target system in FBRL [8] is a set of model of components (devices) representing behavior (mode) of each component, connection information among ports of the components and structural hierarchy of the components as shown in Figure 2. A component can have some behavioral modes. For example, an electronic switch has the switch-on mode and the switch-off mode. As shown in Figure 2, a model of a behavioral mode consists of precondition for the mode, objects, qualitative constraints over attributes of the objects (called QN-relations), and material-product relations each of which represents an output object is made of specific inlet object(s) (called MP-relations). Consider a heat exchanger among two streams of water shown in Figure 3 as an example. It has four ports connecting to neighboring components and objects including inlet cold water, inlet hot water, outlet water, and thermal energies on them. Its behavioral model is described as qualitative constraints (QN-relations) such as those over the temperatures of the streams of water as medium of thermal energy and the heat resistance, and material-product relations (MP-relations) such as the outlet thermal energy at the port No. 2 (denoted by p2) is made of inlet one at the port No.1 (p1) and a part of inlet one of the port No.3 (p3).

2.2 Base-Function Layer

A *base-function* of a component at the base-function layer at functional level is defined as a result of teleological interpretation of a behavior of the component under an intended goal [8]. For example, the functions of a boiler in a steam-powered power plant includes “to vaporize water” and “to heat water”. We call such verbs representing functions of components as (base-) functional concepts. As shown in Figure 1, these functions in a functional model of a specific target system are instances of generic functions in the functional concepts ontology. A function in the model is associated with a behavior realizing it by the mapping primitives called as

Component	
Name-of-component:	/* label of component */
Sub-components:	/* names of sub-components */
Ports:	/* connection with neighbor components */
Inherent-attributes:	/* attributes inherent in the component */
BehavioralMode ₁	/* a behavioral mode */
Precondition:	/* condition for a behavioral mode */
Objects:	/* objects input to or output from component */
MP-Relations:	/* relations among input materials and output products */
SameClass:	/* sameness of the class of in/out objects */
QN-Relations:	/* qualitative relations among attributes */
BehavioralMode ₂	/* second behavioral mode */
....	

Figure 2: The Scheme of the behavioral model in FBRL [8]

the *Functional Toppings* (FTs) of FBRL [8]. The functional toppings represent information about teleological interpretation of behavior according to the designers’ intention. For example, when the heat exchanger shown in Figure 3 is used as a heater or a radiator, it achieves different functions (“heat the cold water” or “cool the hot water”, respectively). The behavior is, however, the same. The functional toppings in Figure 3 show the difference between these two functional interpretations and they are automatically matched with functional concepts in the functional concept ontology. The detail of definitions of functional concepts and functional toppings are discussed in the next section.

There are three points in our approach. The first point is discrimination between the macro-micro hierarchy (the vertical axis in Figure 1) and the teleological interpretation (the horizontal axis). Lind emphasizes the difference between them and calls these relations “whole-part” and “means-ends”, respectively [7]. On the

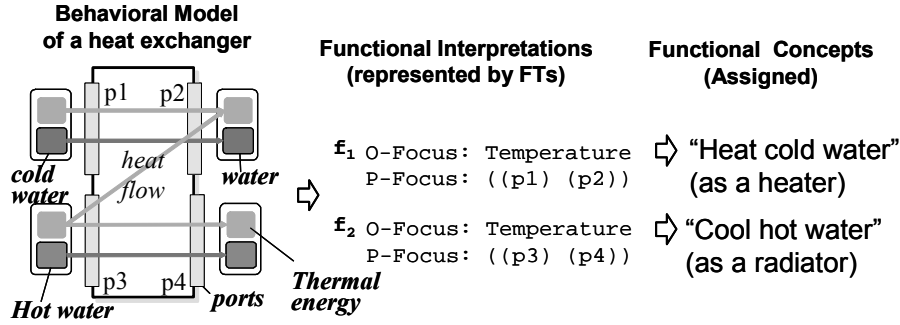


Figure 3. Functional interpretations of a heat exchanger

other hand, FR framework [1] and CFRL [6] define functions as a kind of hierarchical abstraction of behavior and treat the difference of behavior and function as relative relation. Because a function can be achieved by some different sets of sub-functions in different ways, such discrimination is important to keep freedom of design of how to achieve a function by sub-functions. Moreover, dependence of definition of a functional concept on its achievement reduces generality of the definition.

The second point is the ontological approach for the functional level. All entities in the functional level are defined operationally using functional toppings in the functional concept ontology. By "operational", we mean that these functional concepts can be mapped automatically from behaviors with given values of functional toppings as discussed in Section 4. In other words, the functional concept ontology specifies the space of functions and limits functions within the generic functions defined in the ontology. In fact, the automatic identification system screens out the candidates of functional interpretations which match with no functional concept in the ontology as discussed in Section 4.1 on the assumption of comprehensiveness of our ontology. This assumption is discussed in Section 5.2. Although it may reduce the freedom of functional representation in comparison with hand-written functional models, automatic identification of functional concepts reduces cost of modeling and enables us to avoid ad hoc modeling and obtain consistent functional models.

The last point is conceptualization of "way" of functional achievement. A whole-part relation at the functional level represents that a macro function is achieved by a group of sub(micro)-functions (called *achievement relation*). It is similar to functional decomposition in literature [4,9,21]. We also explicated the background knowledge of the functional decomposition such as the physical law and the intended phenomena which allow the decomposition to make sense (we call it *a way of achievement*). Such conceptualization of ways helps us distinguish between "what is achieved" (function) and "how to achieve" (way) and systematize ways of achievement [18].

2.3 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*). For example, the "to vaporize" function of the boiler is said to be To-

Make-type and to perform a meta-function ToEnable for the "to rotate" function of the turbine according to the definition discussed later. The function types represent causal patterns of achievement of goals of each base-function of a component. On the other hand, the meta-functions are result of teleological interpretation of causal relations among base-functions of different components (i.e., inter-components relations). Although both two concepts are grounded on specific causal relations at the behavioral level, causal relations are objective according to physical law while the function types and meta-functions are teleological. Moreover, such meta-function layer refers not to changes of objects of these components but to functions of the components, while other three layers are concerned with existence or changes of objects. Function types represent categories of base-functions from the viewpoint of achievement their goals without referring to specific attributes. Meta-functions are roles (relationship) of two base-functions themselves which are roles of a device (component) for objects within the component.

3. Functional Concept Ontology

The functional concept ontology 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 Figure 4.

3.1 Generic concepts of Base-functions

A *base-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 incoming or outgoing entities of the component. Because function is teleological, a function performed by a concrete component depends on *context* of usage of the component by nature. The context of usage includes designer's intention, other components connected to the component and their configurations. In other words, a function of a component cannot be determined until the component is installed in a specific system with a specific configuration. Although a function of a component depends on other components, the description itself should be local.

We identified three categories of base-function according to the objects it operates on; *energy function*, *entity function* ("entity" here means that physical things including substance except energy), and *information*

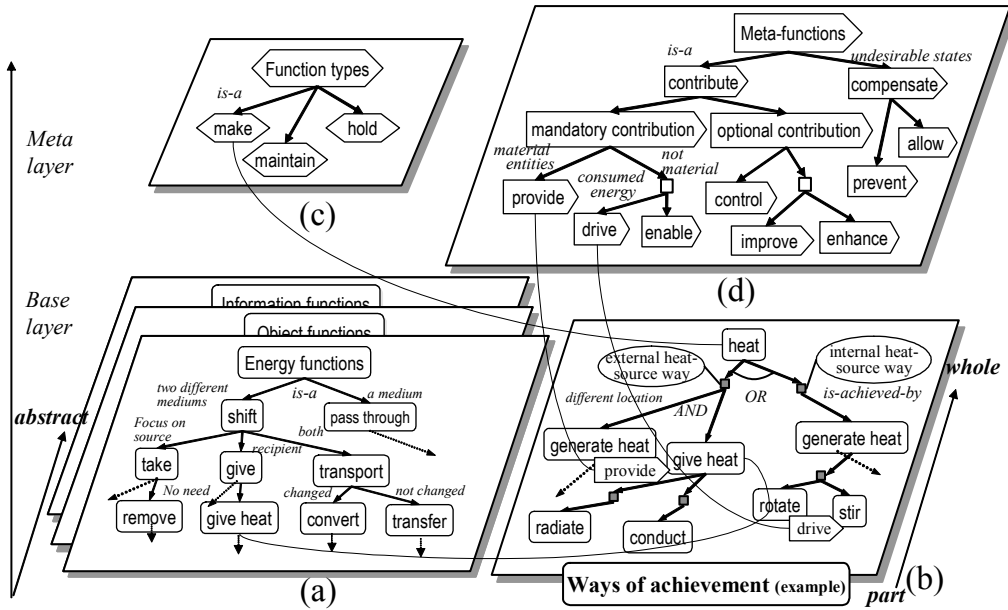


Figure 4. The functional concept ontology (portion)

function. For example, the entity function and the energy function of a turbine are “to generate torque” and “to generate kinetic energy”, respectively.

Figure 4a 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 teleological interpretation called *Functional Toppings* (FTs) of the functional modeling language FBRL [8]. There are three types of functional toppings; (1)*Obj-Focus* representing kind of focused objects, (2)*O-Focus* representing focus on attributes of objects, (3)*P-Focus* representing focus on ports, and (4)*Necessity* of objects. The Obj-Focus specifies kind of objects (entity or energy) which the designer intended to change (focused). The O-Focus specifies attributes intended to change such as temperature. The P-Focus specifies a pair of an inlet port and an outlet port at which the objects intended to change. It means that the designer focuses on the interaction with neighboring components through these ports. The Necessity specifies whether a focused object is needed (used) in other components (denoted by “Need”) or not (“NoNeed”). For details of the functional toppings, see [8].

The values of O-Focus and P-Focus represent *intentional focus* of the function. Such an attribute that is a kind of O-Focus and is an attribute of the object in the focused port indicated by P-Focus is called a *focused attribute*. An object (entity or energy) having the focused attribute is called a *focused object*. Note that “object” represents the role played by entity (or energy) as a target to be affected by the component which plays an agent role.

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 4a.

Entity functions are classified into the following two categories according to the kinds of the focused attribute. The one is called an *amount function* which changes the amount of the focused object or changes its kind or its phase. The other is called an *attribute function* which changes attributes of the objects except the phase. These categories are used in the definitions of the meta-functions.

Note that definition of a base function using FTs is highly independent of its realization by which we mean details of behavior and internal structure of the component corresponding to the function. For example, P-Focus specifies not concrete location but abstract interaction with the neighboring components.

The base functions can be organized into another hierarchical structure according to the is-achieved-by relation which represents that a macro-function is achieved by its micro(part) functions. We also describe a *way of achievement* that represents principles of the achievement such as the physical law and the intended phenomena. Figure 4b 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.

3.2 Function Types

The function types represent the types of goal achieved by the function [5]. Keuneke proposes some function types including “ToPrevent” which is defined as “to keep a system out of an undesirable state of objects” [5]. However, because it focuses on changes of objects associated with the component, another function affected by the undesirable state is implicit. Therefore, we redefined the function type as “ToMake”, “ToMaintain”, and “To-

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

	Object/Energy (focused object of f_i)	Mandatory (f_a is mandatory for f_i)	Type of relation (between f_a and f_i)	Material (the focused object of f_i is made of the focused entity of f_a)	Consumption (the focused object of f_a is consumed by f_i)
Provide	*	Yes	*	Yes	*
Drive	Entity	Yes	Proportional	No	Yes
Enable	*	Yes	Conditional	No	No
Improve	*	No	Proportional	*	*
Enhance	*	No	Proportional	*	*

Legend: Yes: must be, No: must not, * don't care, f_a : the agent function, f_i : the target function

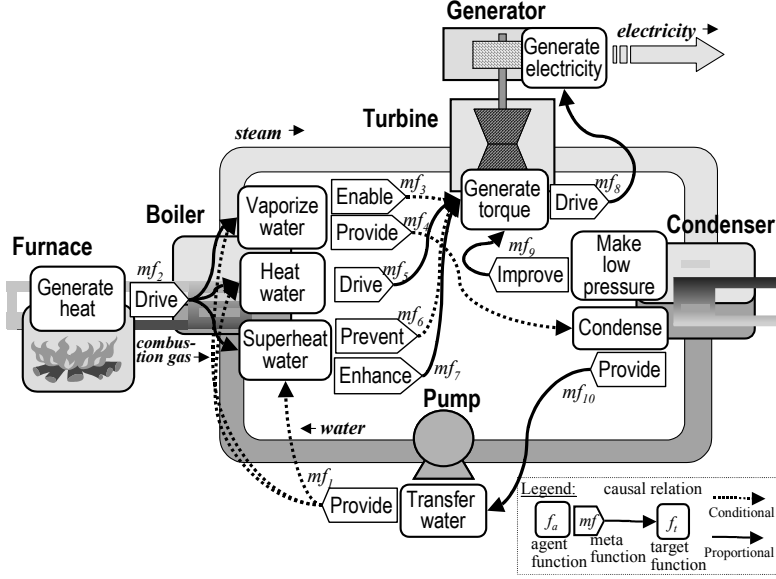


Figure 5: Meta-functions in a power plant (portion. mainly from entity function perspective)

Hold”[22] and redefined “ToPrevent” as a kind of a meta-function as below. 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. “ToHold” is used when no special effort is made for the goal. For example, the function of a pipe, “to pass fluid through”, is said to be “ToHold”.

3.3 Types of 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 a *target function* (f_i) [23]. A meta-function is concerned not with changes of objects of these components but with functions of the components, while the 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 Figure 4d (an is-a hierarchy) and Table 1 (the necessary conditions). Figure 5 shows examples of a part of the meta-functions $\{mf_1 \dots mf_{10}\}$ among mainly entity functions (i.e., not energy-oriented aspect but entity-oriented aspect) in a simple model of a power plant. Note that the furnace which is a sub-component of a

boiler is separated from the boiler as a heat exchanger for explanation.

The conditions for the meta-functions refer to the definitions of the agent and target base-functions and the relations among them. The focused object (entity or energy), the focused attribute, and the categories of the base-functions such as the amount function and the attribute function defined in Section 3.1 play a crucial role in the definitions. The relations among functions are *causal* or *structural*. The causal-type functional relations between two functions represent the causal relations among the attributes focused by the functions. The attributes focused by functions (i.e., attributes teleologically intended to change by the designer. we call them focused attributes as discussed in 3.1) can be identified using FTs in the functional model. Causal relations among physical attributes can be determined using qualitative constraints (called QN-relations) in the behavioral model shown in Figure 2. The causal-type relation is categorized into some sub-types such as *proportional* and *conditional* according to relationship among attributes that functions focus on. The former represents that an attribute value has proportional relation with another attribute. 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 relation represents topological and temporal relations among functions. It is also categorized into subtypes such as *series*, *parallel*, *sequential*, *simultaneous* and *feedback*. Because connection among components implies flow of objects with time delay, topology of connection of components implies temporal relations of their functions.

We begin definition of meta-functions with the condition where there is a causal relation from the focused attribute of f_a to that of f_i . If the goal of f_i is not achieved when f_a is not achieved, the f_a is said to have a *mandatory contribution* to f_i . Although we could intuitively say that f_a has a ToEnable meta-function for f_i in such a case, the authors define a narrower meaning of ToEnable by excepting the cases of ToProvide and ToDrive as follows.

ToProvide (Provider role)

When a function f_a generates (or transfers) the *materials* which another function f_i intentionally processes, the function f_a is said to perform a meta-function “to provide material” for the function f_i . The intentionality means that the designers intend the changes of its attributes as its function and can be represented by FTs. The material of f_i can be basically defined as input objects (entity or energy) which will be a part of the output objects on which the function f_i focuses. The focused objects and focused attributes can be determined by FTs as discussed in Section 3.1. If f_i is an amount function (see Section 3.1 for the definition), the output entities are mixture of the source materials or synthesized from the materials. For example, the “to transfer water” function of the pump has a ToProvide meta-function for the “to vaporize” function of the boiler (see mf_1 in Figure 5). If f_i is an attribute function, the function changing the same attribute of the focused material object is said to have a ToProvide meta-function. If f_i is an energy function, the source energy of the output energy is called material energy. Such a material energy is transferred to a different medium by f_i or transformed into a different kind of energy. For each function, in general, there exists another function as an agent function performing a ToProvide meta-function. Especially, from the energy aspect, many of energy functions perform “to transfer” or “to transform”, meta-functions among them are usually ToProvide.

ToDrive (driver role)

We call such energy that *essentially* causes the internal process of an entity function f_i *driving energy*. The necessary conditions for the essentialness are (1)not material of the focused object of f_i , (2)a proportional causal relation to the focused attribute of f_i , (3)intentionally consumed by the process of f_i (see Table 1). What we mean by “intentionally consumed” does not include the unintentional loss such as that during transfer of energy, which can be represented using the necessity FT. In many cases, a part of the driving energy is transformed into the different kind of energy and the focused entity of f_i brings the transformed energy.

The function which generates or transfers such a driving energy is said to have meta-function “to drive f_i ”.

For example, the “to heat water” function of the boiler has a ToDrive meta-function for the “to generate torque” function of the turbine (see mf_3), because the amount of the heat energy transferred by the heating has a proportional causal relation to the torque and is consumed by the rotation.

ToEnable (enabler role)

This meta-function is used for representing a mandatory condition playing a crucial role in f_i except ToProvide and ToDrive (see Table 1). What we mean by this weak definition is that the conditions such as the existence of the material and the existence of the driving energy are too obvious to be said to enable the function. ToEnable is primarily a role of an attribute function for an entity function. On the other hand, ToProvide is a role of an amount function for an entity function or a role of an energy function for an energy function.

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 a meta-function ToEnable (see mf_3). On the other hand, it performs a different meta-function ToProvide for the “to condense” function of the condenser which condense the steam to the water (see mf_4), because the functions focus on the same phase attribute.

ToAllow and ToPrevent

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_{i1} is said to have a meta-function “to allow the side-effects of f_{i1} ”. The “undesirable side effect” is defined in a relation with another function f_{i2} or the whole system. The “positive effect” means such a causal relation that increase of the focused attribute of f_a causes decrease of the side effect.

If a serious trouble (e.g., faults) will be caused in a function f_{i2} when a function f_a is not achieved, function f_a is said to have a meta-function “to prevent malfunction of f_{i2} ”. For example, the “to super-heat” function of the boiler prevents malfunction of the turbine (mf_6), because the steam of low temperature would damage the turbine blade by water particles. For almost all f_a performing a ToAllow meta-function for f_{i1} , in general, there exists a ToPrevent meta-function for another function f_{i2} .

ToImprove and ToEnhance

These meta-functions represent *optional* contribution to f_i . The “optional” means that the effect of f_a is not mandatory to f_i , that is, f_i can be performed without the effect of f_a . The discrimination between ToImprove and ToEnhance is made by whether augmentation of the output can be made without increase of the amount of the input energy or not. For example, the “to make low pressure” of the condenser contributes to the efficiency of the “to generate torque” function (see mf_9) without increment of input energy, so its meta-function is “ToImprove”. On the other hand, the “to super-heat” function of the boiler optionally increases the amount of the input energy to

make a contribution. Its meta-function is “ToEnhance” (mf_7).

ToControl (controller role)

When a function f_a regularizes the behavior of f_i , its meta-function is said to be “to control f_i ”. In other word, values of the focused attribute of f_a having ToControl are changed to maintain the values of the focused attributes of f_i . 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 at a certain value. It is said to have a meta-function ToControl for the “to vaporize” function of the boiler (not shown in Figure 5).

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 said to be ToMaintain-type.

4. Functional Understanding

The functional understanding problem is to identify functional structures of an artifact from the given behavioral structural models of components. The process of functional understanding consists of the three steps shown in Figure 6. First, given behavioral component models, all possible functional interpretations of behavior of each component are generated (called behavior-function map-

ping). Next, the meta-functions of generated functional interpretations are identified (called identification of meta-functions). From the functional interpretations and meta-functions among them, functional hierarchies are generated (called consolidation of base-functions). Although many candidates of the functional interpretations are generated by the first step, plausible functional interpretations are identified in the second step and the third step. In this section, we discuss these steps in detail.

4.1 Behavior-Function Mapping

Firstly, the identification 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 Figure 7, the system generates a functional interpretation f_3 which consists of O-Focus on the “phase” attribute and P-Focus on the inlet water and the outlet steam.

Then, the identification 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 Figure 7, 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 can-

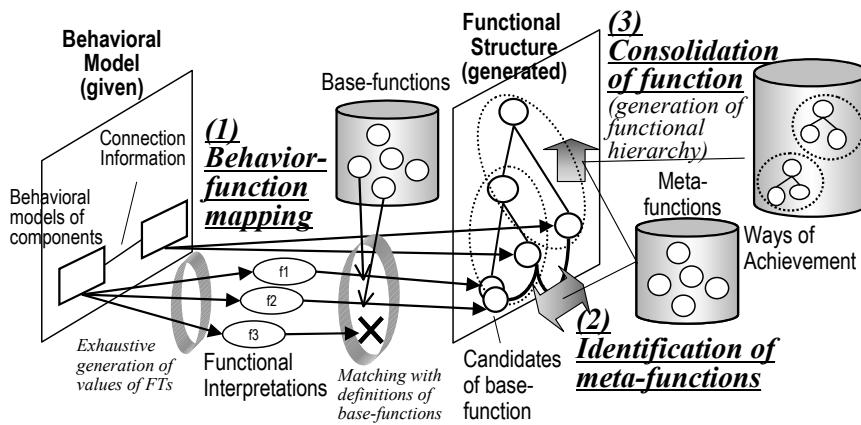


Figure 6. Three steps of automatic identification of functional structures

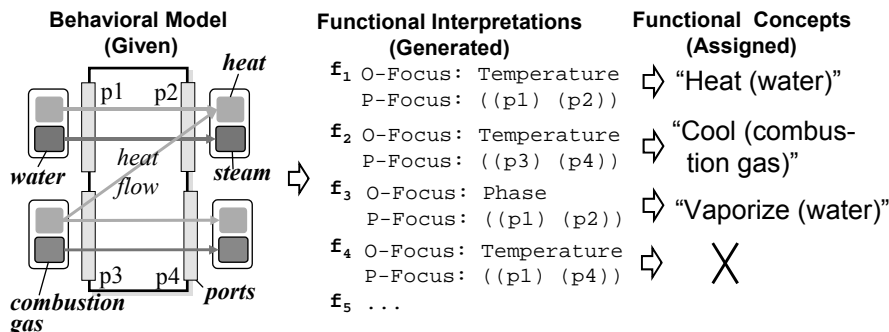


Figure 7. Behavior-function mapping of a boiler

didates of the functional interpretations remain, plausible functional interpretations are identified by the following steps.

4.2 Identifying Meta-functions

Given a pair of candidates of base-functions, the type of a meta-function between them can be identified according to the definitions shown in the Table 1. Firstly, the attributes focused by each function of the given pair of base-functions (called focused attributes) and the objects focused (focused objects) are identified using the values of O-Focus and P-Focus of the function as discussed in Section 3.1. Next, the causal relation among the focused attributes is checked using the behavioral constraints (QN-relations in FBRL model [8]) in the models of components of the two functions and the connection information among the components. If there is mandatory causal relation, whether or not the focused object of the target function is made of entities propagated from the focused objects of the agent function is checked using the material-product relations [8] in the behavioral model and the connection information. Because this is the necessary condition of ToProvide, if the condition is true, the agent function is said to perform ToProvide to the target function. If not, the conditions for ToDrive shown in Table 1 (types of causal relations and consumption of energy) are checked using the conditions of the behavioral modes and the behavioral constraints. If the condition is satisfied, then the agent function is said to perform ToDrive. If the type of causal relation is conditional, the agent function is said to ToEnable.

The key information used for identification is summarized as (1)relationship among entities, (2)connection information and (3)functional representations. The relationship among entities such as material-product relations can be identified using the behavioral model. The connection information is provided by the structural model. The functional representation using O-Focus and P-Focus enables the identification module to identify the focused objects and the focused attributes.

In order to identify the ToPrevent meta-functions, the knowledge of unintended abnormal phenomena (so-called general fault models) is also needed (Note: At this moment, our implementation does not cover this knowledge. Although completeness of such fault models is difficult to obtain, we have identified general fault models which can capture wide range of faults [24]).

According to identified meta-functions, the identification system deletes such meaningless functional interpretations that do not contribute to any others. In the example, the functional interpretation f_2 “cool (the combustion gas)” of the boiler shown in Figure 7 is deleted. It represents reduction of heat energy of the combustion gas, which is meaningless in the power plant (no contribution to any other).

4.3 Consolidation of Base-functions

Lastly, the base-functions generated in the behavior-function mapping are consolidated (aggregated) into macro-functions (typically as a function of a subsystem

or the whole system). Then, functional hierarchies are generated basically in a bottom-up manner. If the function (or goal) of the whole target system is given, the identification system generates a functional hierarchy whose root (top-most) function (i.e., the function of the system as a whole) is equal to the given function. Otherwise, various root functions are generated. However, on the assumption that the whole system achieves a function which has some effects to the outside of the system, the top-most functions should be associated with the attributes of the output from the target system. For example, in the power plant, the top-most functions should be associated with the electricity of the generator, or the temperature of the cooling water output from the condenser. Thus, the top-most functions are “to generate electricity” or “to warm the (cooling) water”. Although we do not recognize the latter as the function of the power plant in general, such recognition requires the model including the context which the power plant is used in.

General Procedure of Consolidation and Necessary Knowledge

Firstly, from the given base-functions, groups of base-functions are made according to some criteria (called grouping criteria). Because each functional group will be consolidated into a macro-function, such a criterion determines the extent of a macro-function, its identity, and the cohesiveness of its sub-functions. The most usual one of the grouping criteria is “structural”, which represents inclusion relationship of component and (sub-) systems [5,15]. This criterion is given externally by users. According to this criterion, each of macro-functions corresponds to one of sub-systems or the whole system. In general, functional structure does not necessarily coincide with the structural hierarchy. For example, we can find a macro-function representing changes of energy and another macro-function representing changes of substance. In such cases, a group of functions is sometimes formed by a set of components across the inter-subsystem boundary. In order to do this, we need other criteria related to the continuity of substance or types of energy.

Next, the base-functions in each functional group are consolidated into a macro-function. In this phase, because there are some functional groups to consolidate, knowledge is needed to determine which group is consolidated firstly.

The last knowledge needed for consolidation is for determining a macro-function for sub(micro)-functions. In the literature about automatic identification of functional structures [3,16,21], predefined aggregation “patterns” provide a macro-function for a specific group of sub-functions. In our system, the ways of achievement plays this role.

Generation of a kind of functional hierarchy, however, can be done by the following different manner. There is a kind of functional hierarchy in which a macro-function is equal to a main sub-function in a group of sub-functions (we call this type “part-focused consolidation”). For example, a macro-function “to distill liquid oil” can be achieved by the sub-functions “to vaporize the liquid oil” and “to distill the oil gas”. The second

Table 2. Heuristics for consolidation

A: Consolidation heuristics (mandatory)

- **A1:Macro-function heuristic.** Given a viewpoint for recognition, there always exists a macro-function for a functional group.
- **A2:Causal relation conservation heuristic.** The causal relations among attributes in the finer grain size are conserved in courser grain-sizes consolidated.

B: Selection heuristics (mandatory)

- **B1:Serial heuristic.** In serial functions, the system can consolidate functions in the head of chains.
- **B2:Causal relations heuristic.** The macro-function made from micro-functions which have most causal relations is preferred to others.
- **B3: Meta-function selection heuristics. (B3a)** ToDrive represents more cohesive relation than that of ToProvide to any f_i . **(B3b)**ToEnable and ToPrevent are preferred to other meta-functions because they are more specific than others.

H: Hierarchical structure 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 macro-function.

X: Selection heuristics (optional, application order)

- **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: Partition heuristics (optional, relaxing order)

- **Y1:Structural boundary heuristic.** The boundaries between functional groups coincide with those of components or (sub-)systems.
- **Y2:Energy-type boundary heuristic.** The boundaries coincide with changes of types of the focused energy
- **Y3:Entity-type boundary heuristic.** The boundaries coincide with changes of kinds of the focused entity or its phase.
- **Y4:Attribute-type boundary heuristic.** The boundaries coincide with changes of types of the focused attributes.

Z: Grouping heuristics (optional)

- **Z1:Meta-function-groups heuristic.** The groups are made according to meta-functions.

sub-function plays a main role in the sequence of sub-functions, which is called the main sub-function in this “is-achieved-by” relation. The information for teleological interpretation (the functional toppings in our framework) of the macro-function is equal to the main sub-function, although their target objects (i.e., the phase of the oil) are different. Such hierarchy can be generated without the predefined patterns if we can identify the main function in a group of base-functions. In our system, the meta-function enables us to do so as discussed later.

Heuristics for Consolidation

In our approach, the required knowledge discussed thus far is realized as the fifteen heuristics classified into six categories shown in Table 2 and meta-functions discussed in the previous section. The roles of the heuristics are to form groups of functions (category *Y* and *Z*. We call *Y* and *Z* *partition heuristics* and *grouping heuristics*, respectively), to switch knowledge source of macro-functions (category *H*. *hierarchical structure heuristics*) and to select a macro-function from candidates (category *A* (consolidation heuristics), *B* and *X* (*selection heuristics*)).

Although heuristics *A* and *B* are mandatory and all of them have to be applied, others are optional. The mandatory heuristics are needed to force generated hierarchies to be reasonable for human recognition. The application of optional heuristics (*H*, *X*, *Y* and *Z*) can be specified by users, which enables the system to generate various functional hierarchies. There are three types of specification of application of the optional heuristics. That of heuristics *H* is the first type which is like a switch of the knowledge source. That of heuristics *X* and *Z* is the second type which gives order of application. That of heuristics *Y* is the third type which gives order of relaxing of heuristics.

Knowledge source selector *H* determines which knowledge for generating macro-functions to use, that is, either the ways of achievement shown in Figure 4b (*H1*)

or the meta-functions among base-functions as discussed later (*H2*).

The heuristics in the category *X* determine preferences of groups of functions, that is, which groups of functions should be firstly consolidated into a macro-function. Users can specify the order of application of the heuristics (or no application of the heuristics). The earlier application of a heuristic means that functions in such a group that satisfies the heuristic’s condition are invisible in coarser grain-sizes. For example, when a user specifies that the *X1*:parallel-first heuristic are preferred to 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. Thus, the functions in parallel-type relations could be invisible in the coarser grain-size.

The heuristics in the category *Y* define boundaries between functional groups and then partition sub-functions into functional groups. We identified the four types of conditions according to major concepts in the device ontology, i.e., component, energy, medium, and attribute. For example, *Y1*:Structural boundary heuristic defines boundaries according to boundaries of physical structure (i.e., components, or (sub-)systems). In the same manner, the other three heuristics *Y2*, *Y3* and *Y4* define boundaries of functional groups according to types of focused energy (e.g., thermal and kinetic), type of focused entity including difference of its phase (e.g., steam, water, and naphtha), and types of focused attributes, respectively.

At the first phase of the consolidation, the strictest partitioning condition of all specified heuristics makes the smallest groups of the finest grain-size. Then, the identification system relaxes the condition and then makes larger groups of coarser grain-sizes. Thus, users can specify the order of relaxation (or no application of the heuristics). The identification system firstly forms the groups of given base-functions according to the all criteria of the category *Y* specified by users, and then consolidates them into macro-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 order specified by the user. Then new functional groups are made, and then functions in them are consolidated.

The heuristic in the category *Z* specifies the cohesiveness of functions for grouping them. Although the goal of the heuristics *Y* and *Z* is the same, types of application are different. If *Z1* is specified, *Z1* is not relaxed. We discuss this heuristic in the next subsection.

On the other hand, those in categories *A* and *B* represent working assumptions of the system and thus are always applied. *A1*:Macro-function heuristic represents a strategy to consolidate functions, that is, to form functional groups corresponding to macro-functions. This also argues that a functional hierarchy depends on the viewpoint of recognition. *A2*:Causal relation conservation heuristic represents a basic characteristics for aggregation of causal relations.

B1:Serial heuristic reflects humans understanding way based on the temporal order. *B2*:Causal relations heuristic represents a preference of macro-functions supported by many causal relations, because many causal relations reflect close interaction among functions. Note that *X3* is concerned with number of micro-functions, while *B2* is concerned with number of causal relations among the micro-functions. *B3* is explained in the following section.

Roles of Meta-functions in Consolidation

Meta-functions play a crucial role in consolidation. Firstly, because each type of meta-functions has own *strength (cohesiveness)* to form functional groups, the grouping and selection can be done according to the types of the meta-functions among them as well as causal relations and structural relations. It is implemented as *Z1* heuristic in Table 2 for generating groups according to meta-functions and *B3* heuristics for giving strength (i.e., preference) of each meta-functions.

Z1 heuristic enables us to form functional groups from two base-functions connected by a meta-function, since a meta-function represents collaboration between functions in order to achieve macro-functions. *B3* (*B3a* and *B3b*) heuristics are needed to determine which group should be firstly consolidated in the case that some meta-functions contribute to the same target function (i.e., some groups are formed by *Z1* heuristic). According to *B3a*, the groups having *ToDrive* are preferred to those having *ToProvide*. Its reason can be explained that *ToDrive* has narrower extent than *ToProvide* does because a material of a target function provided by an agent function of *ToProvide* could be also a material of other functions followed by the target function, while the driving energy generated by an agent function of *ToDrive* is usually *not* the driving energy of the following functions. On the other hand, according to *B3b*, the

groups having *ToEnable* or *ToPrevent* are preferred to those having other meta-functions because these two meta-functions are concerned with more specific conditions than general conditions of other meta-functions.

Meta-functions also indicate a main function in a functional group to which other functions contribute. Because the whole function of a functional group is equal to such a main function in the part-focused consolidation as discussed before, the identification system can generate a macro-function which is equal to the main function, though the target objects of functions are different. According to *H2* heuristic representing this, the macro-functions can be generated without the predefined aggregation patterns of functions such as those in [3,16,21] and our ways of achievements shown in Figure 4b in the case of *H1* heuristic.

H2 heuristic is usually used with *Z1* heuristic. When *Z1* heuristic is specified, users' specification of application of heuristics in category *X* has no meaning, because the groups according to meta-functions are usually not parallel (*X1*), always imply causal relations (*X2*), and always include two base-functions (*X3*). On the other hand, if users specify application of both heuristics in category *Y* and *Z1*, functions are firstly partitioned according to the specified *Y* heuristics and then two functions having a meta-function in each partition are grouped. The *Y* heuristics will be relaxed as mentioned before, while *Z1* heuristic is not relaxed once specified.

Examples of Consolidations in a Power Plant

Figure 8 shows an example of a functional hierarchy of a part of the power plant shown in Figure 5 (some functions are omitted for simplifying explanation), which is generated by the implemented functional understanding system discussed in the next subsection according to meta-functions among base-functions (the heuristics setting is *H2* and *Z1*). Firstly, the system forms each functional group from two base-functions combined by a meta-function. Among them, the functional group which consists of "to vaporize water" of the boiler and "to generate torque" of the turbine is firstly consolidated into a macro-function "to generate torque", because these two base-functions are combined by a *ToEnable* meta-function which is preferred to other meta-functions ac-

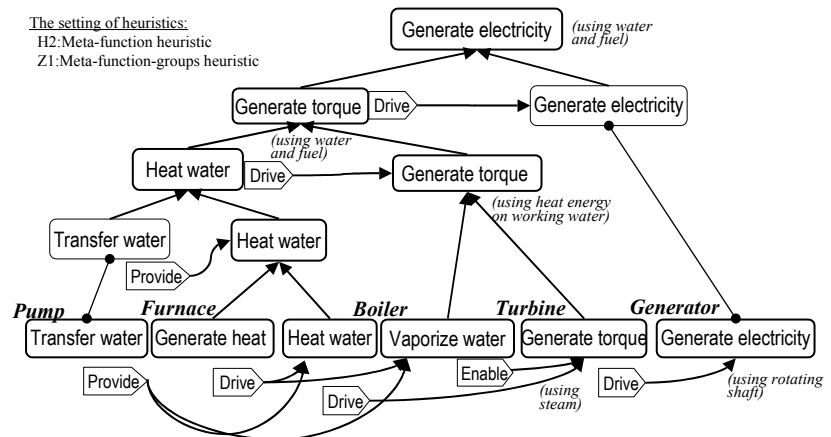


Figure 8: A functional hierarchy generated by the system according to meta-functions

According to *B3b* heuristic. Next, “to generate heat” of the furnace and “to heat water” of the boiler having ToDrive meta-function are consolidated into a macro-function “to heat water”, before the pair of “to transfer water” and “to heat water” having ToProvide meta-function is consolidated according to *B3a* heuristic. The rest of the hierarchy is constructed according to the temporal order (i.e., from the head of the causal chain. In this example, the pump is assumed as the head of the causal chain), because only ToDrive meta-functions remain.

Figure 9 shows another hierarchy generated by the implemented system according to another heuristics setting which is added *Y1*:structural groups heuristic to the previous setting (*H2* and *Z1*). In this case, the system firstly forms the functional groups (indicated by gray boxes in Figure 9) whose boundaries coincide with those of sub-systems such as the boiler sub-system (which consists of the furnace and boiler) and the generator sub-system (the turbine and the generator). In each functional group, two consolidations into “to heat water” and “to generate electricity” according to meta-functions in the same manner in the previous example are done. Because there is no meta-function in functional groups made according to the structure at this point, *Y1* heuristic is relaxed (Note that *Z1* heuristic is never relaxed). Then, the rest of the hierarchy is constructed according to meta-functions.

Figure 10 shows a possible hierarchy according to another heuristics setting without using meta-functions (this example is not by the implemented system). In this case, the macro-functions are generated according to the knowledge base of ways of achievement of function shown in Figure 4b. Firstly, the system makes the functional groups such as the functions changing pressure-type attribute (fg_3 in Figure 10) according to the specified grouping heuristics *Y4*, that is, groups made by kinds of attribute. Next, in the functional group fg_2 , the external heat-source way of “to heat” in Figure 4b matches “to generate heat” and “to give heat” (which is an energy function from energy-oriented aspect of the entity function “to heat water” shown in Figure 5), then they are consolidated into a macro-function “to heat”. After functions in other functional groups are consolidated into each macro-function, the condition of *Y4* is relaxed and then these consolidated functions are consolidated again into a macro-function “to generate torque” according to the thermal steam way.

When the user changes the order of applying the

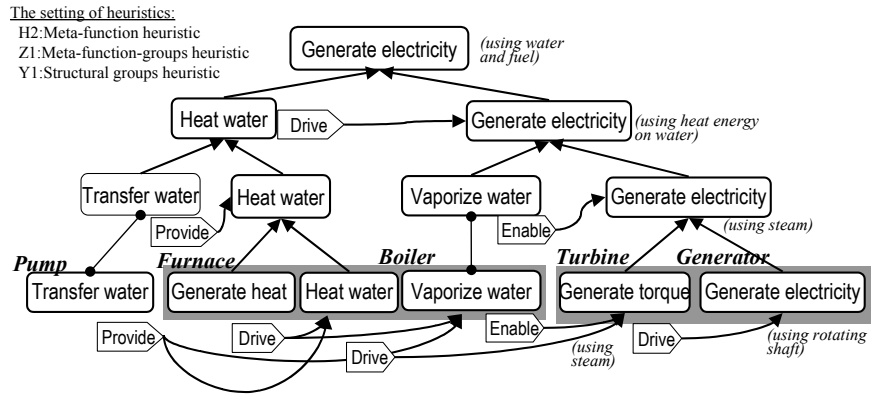


Figure 9: Another hierarchy generated by the system with *H2*, *Z1* and *Y1*

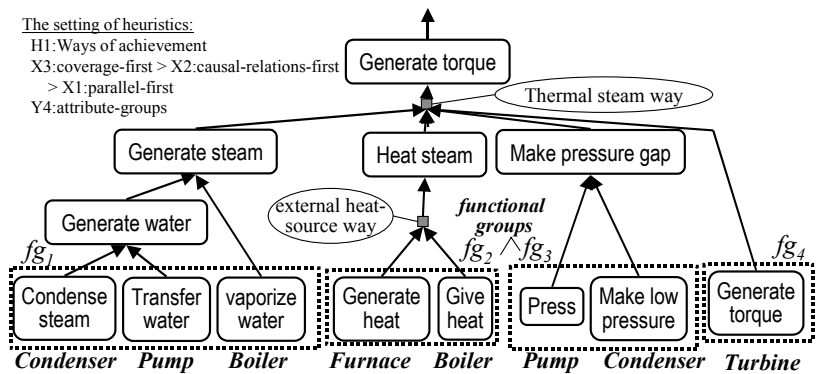


Figure 10: A functional hierarchy using ways of achievement (by hand simulation)

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, though all of them make sense. While the first one (Figure 8) represents how to obtain the driving energy and how to convert the heat energy to kinetic energy, the last one (Figure 10) represents conditions for kinds of attributes. The second one (Figure 9) is generated according to the structural viewpoint specified by *Y1* heuristic and then has a function corresponding to the generator sub-system. There is no function corresponding to the turbine sub-system, because it has two independent sub-functions, that is, “to vaporize” and “to heat”, which have no meta-function. Meta-functions enable the identification system to generate hierarchies according to cohesiveness including interdependency beyond structure. Some other possible hierarchies are shown in [22].

4.4 Implementation as a part of an explanation generation system

The functional understanding system has been implemented using Allegro Common Lisp on UNIX and Microsoft Windows. The current implementation supports only identification of meta-functions (except ToPrevent), consolidation of the part-focused type using meta-

functions and heuristics switches of category *Y*, because we concentrate on meta-functions and *X* heuristics have nothing to do with meta-function as discussed above. The functional understanding system is a part of an explanation generation system which can provide explanation of artifacts in natural language (Japanese) using functional concepts.

Figure 11 shows a part of an oil refinery plant as an example. The main process of the plant in the model is that the crude oil is firstly heated by the pre-heater (E13), taken off the gas ingredient by the pre-flush drum (V12), vaporized by the heater (H1), distilled by the distiller (C11) and then liquefied by the overhead drum (V13). The flow-rate of the oil flowing into the heater is controlled by the control valve (FC17). Given the behavioral model and structural model, firstly, the user inputs the name of component for explanation (the heater, H1, in this case). Then, the system shows the following list of available explanations; 1: functional explanation concerning entities, 2: behavioral explanation concerning entities, 3: functional explanation concerning energy, 4: behavioral explanation concerning energy, 5: explanation of meta-functions concerning entities, and 6: explanation of meta-function concerning energy. Next, in this example, the user selects the option 5 for the meta-functions of functions of the heater (H1). After the user inputs the ID-number of two functions of the heater (1), the explanation is generated;

“The heating function of the heater H1 drives the distiller C11 to distill the oil”

The explanation is not pre-defined but dynamically generated according to the result of identification of meta-functions (as mentioned in Section 4.2) which is based on the fact that the thermal energy given by the heater is consumed in the distilling process.

The another generated explanation of another function of the heater is;

“The vaporizing function of the heater H1 enables the distiller C11 to distill the oil”,

because it is crucial condition that the phase of the inlet oil is gas for the distilling process.

After the user inputs the kinds of explanation (in this example hierarchical explanation of the whole system) and the optional heuristics in category *Y* and *Z* (option 1 as only *Z1* heuristic), the system generates a functional hierarchy according to the meta-functions and then explains it in top down manner.

“The whole system extracts naphtha from crude oil, which is achieved by that the distilling function provides the material for the extracting function of the overhead drum (V13)”

This explanation includes the top-most function (to extract) of the whole system and its two sub(micro)-functions (to distill of the distilling subsystem and to extract of the overhead drum) together with a meta-function (ToProvide) between them. Note that this top-most function is one of possible ones, because the top-most function is not given by user in this

example as discussed in Section 4.3. The explanation is recursively generated until the all functions of components are covered.

Figure 12 depicts a diagram of the same functional hierarchy discussed above (The current system does not support the function of diagram drawing). Here, there are two ToEnable meta-functions. One is mentioned above. The other is among the liquefying function of the pre-flush drum (V12) and changing-flow function of the control valve (FC17), because the control valve is designed to control flow of liquid. You can see that these two ToEnable meta-functions lead the functional understanding system to consolidate those functions firstly according to the B3b heuristic. Then, according to the *B1* and *B3a* heuristics, the functions having ToDrive meta-functions are consolidated. Lastly, the functions having ToProvide are consolidated.

In summary, the functional understanding system enables the explanation generation system to generate various explanations including functional explanation from the behavioral models without predefined explanation. According to the users’ selection of a kind of explanation and viewpoint for generation of functional hierarchies (though the current implementation supports *Y* heuristics only), the system can generate explanation for

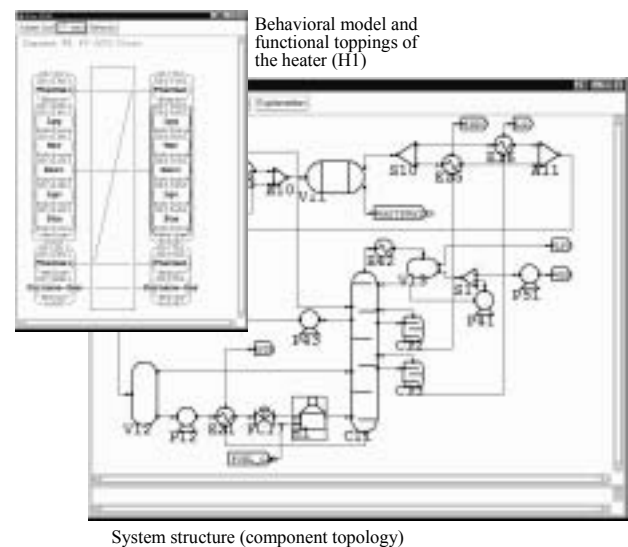


Figure 11: The Oil Refinery Plant in the Implemented System

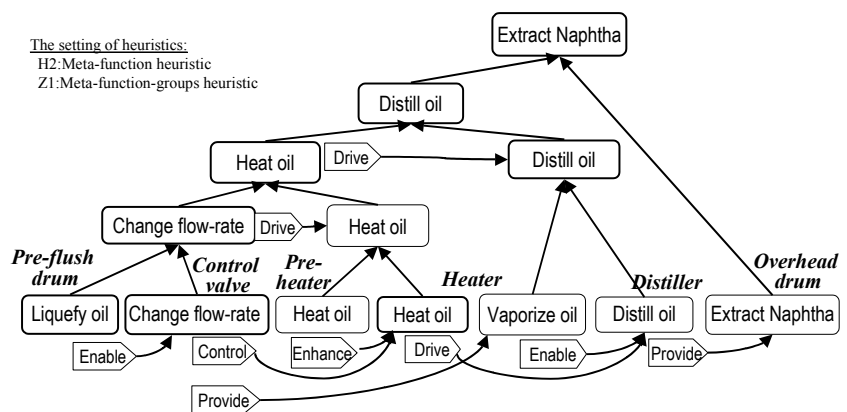


Figure 12: The identified functional hierarchy of the oil refinery plant

the needs.

5. Discussion

5.1 Key concepts enabling automatic identification of functional structure

One of the key points of this research is definition of functional concepts together with operational relationship with behavior. Such relationship enables a computer system to identify the functional structure from behavioral and structural models. In order to define such relationship, we have articulated such *primitive concepts* that enables mapping from a behavioral model to a functional model. Using such mapping primitives, we have defined base-functions and meta-functions as discussed thus far. The following list summarizes the mapping concepts used in definitions of the functional concepts where (from “*other-concepts*”) means that the concepts are defined using (derived from) the *other-concepts*;

(1) Functional toppings

- Kind of focused object (as Obj-Focus in FTs, see section 3.1 for definition)
- Focused attribute (as O-Focus in FTs, see section 3.1 for definition)
- Focused port (as P-Focus in FTs, section 3.1)
- Necessity (as Necessity in FTs, section 3.1)

(2) Important information in the behavioral model

- Material-Product relation (as MP-relation [8] in the behavioral model, see section 2.1 and [8])
- Causal relations (from QN-relation [8] in the behavioral model and the structural model, see Section 2.1 and [8])
- Types of causal relations among focused attributes; proportional, and conditional (from the behavioral model and the structural model, see section 3.3)
- Mandatory of causal relation (from causal relations, section 3.3)

- Types of structural or temporal relations among functions; series, parallel, sequential, simultaneous, and feedback (from the behavioral model or the structural model, section 3.3)

(3) Teleologically focused behaviors

- Focused object (from Obj-Focus, O-Focus, P-Focus and Ports [8] in the behavioral model, see section 3.1)
- Types of focused object; entity, energy, and information (from O-Focus and P-Focus, section 3.1)
- Types of focused attribute; amount and attribute (from O-Focus and P-Focus, section 3.1)
- Roles of medium in energy transfer; source and recipient (from P-Focus and MP-relation, section 3.1)
- Material object (entity or energy) (from MP-relation and types of focused attribute, section 3.1)
- Driving energy (from material, MP-relation and intention of consumption from Necessity, section 3.3)

All of them shown in (2) and (3) are defined (or derived) using given information in the behavioral model or the structural model. Those in (1) (i.e., values of FTs) in functional models can be exhaustively generated from the behavioral model, because the possible values of FTs are limited when a specific behavioral model is given. Such conceptualization makes the identification system to concentrate the information in the given models and then to generate functional structure without other knowledge source (except consolidation heuristics and fault models).

5.2 Limitation of our approach

The list of key concepts also shows the limitation of our approach for functional modeling and its automatic identification. Firstly, all information shown in (2) and (3) can be retrieved from the behavioral model according to additional functional toppings. It means that the behavioral models of a component should include all possible behaviors to be interpreted as function. In other words, it requires behavioral models independent of context of their usage (e.g., configuration of systems) and the functional understanding system tries to guess designer’s intention under the given configuration. Although it is hard to describe models completely independent of context, it is much easier to describe possible behaviors, which are objective, than possible teleological interpretations depending on many contexts.

Secondly, all functional concepts appearing in the functional model should be defined in the functional concept ontology using functional toppings as mapping primitives between behavior and function shown in (1) in the list. As discussed in Section 2, although this ontological approach may reduce the freedom of description, it enables us to obtain well-controlled functional description without ad hoc modeling. The current set of functional concepts consists of about one hundred base-functions, three function types, and eight meta-functions. We cannot claim completeness of the set of concepts. Note that we defined not domain-specific functions but general functions common in many domains. In fact, almost all of energy functions are applicable to all kinds of energy. If needed, we have to add domain-specific functions as subtypes of the leaf nodes. Even if the set does not include a domain-specific function for a behavior, the functional understanding system does not fail to capture the function. It just interprets the behavior as a general function rather than a function specific to the behavior in the domain. In design research [9], very few (from 3 to 24) generic functions are defined. They are highly abstracted. Although one might think the set of functional concepts is huge, the huge set is not the set of function but of the set of “ways” (how to achieve it). In fact, in Value Engineering research [14], 158 verbs are proposed as a standard general set for representing functions of artifact. Although it includes functions for human feeling also, we concentrate on functions changing physical attributes. Concerning meta-functions, because meta-functions do not refer to specific attributes, the set of meta-function is not huge. We are currently investigating other types of meta-functions about temporal rela-

tions.

Thirdly, computational cost of this approach for large-scale application is intuitively cheap. Firstly, the behavior-function mappings can be done in each component ($O(n)$; where n is the number of the components). Secondly, the identification of meta-functions can be done for all pairs of the generated base-functions ($O(n^2)$; where n is the number of base-functions). Lastly, the consolidation process based on meta-function (when the H2 heuristic is specified) is done for each meta-function. In the case of the H1 heuristic, the all knowledge of ways of achievement are tried to match with functions in each functional group, because the Y heuristics partition the functions of the whole system into some functional groups. Then, the order of complexity is not exponential but polynomial for numbers of components of the target system.

Lastly, our functional understanding system can generate functional hierarchies of the part-focused consolidation type according to meta-functions without predefined aggregation patterns as discussed in Section 4.3. A hierarchy having no main sub-function equal to the macro-function, however, cannot be generated using meta-functions. For example, “to control an attribute” function of a general control system can be achieved by “to measure its current value” of a sensor, “to calculate adjustment of a manipulated attribute” of a controller and “to change the manipulated attribute” of an actuator. In this achievement relation, there is no main sub-function equal to the macro-function. Such hierarchies can be generated using the knowledge of ways of achievement.

5.3 Application domains, evaluation, and limitation

The ontology has been applied to modeling of a power plant, an oil refinery plant, a chemical plant, a washing machine, a printing device, and three manufacturing processes [23]. Firstly, the model of the power plant shown in Figure 6 mainly focuses on thermal energy carried by steam. The device ontology is very appropriate for such fluid-related plants. The functional concepts used in this example are mainly related with thermal energy, temperature and phase of water. On the other hand, the oil refinery plant and the chemical plant also treat changes of ingredients of the target objects. So, the entity functions about ingredients such as “to distill” and “to react” are used.

Next, the washing machine and the printing device treat changes of not only several attributes of objects (cloths and papers) but also force and motion of objects. For example, in the washing machine, the movement of cloth is important because it causes friction for washing to get rid of dirt of cloth. We extended the device ontology to cope with force and motion and treats force and motion as a kind of objects. Although the many of functional concepts for force and motion share with the generic energy functions, special concepts concerning their directions and position are added. On the other hand, in the fluid-related plants such as the power plant mentioned above, although the flow rate (i.e., velocity and volume) of fluid and its pressure are also important, they can be treated as same as the other attributes such as

temperature because the flow direction along the connection among components is implicit.

The last application domain is manufacturing process. We describe functional models of a manufacturing process of painting, two slicing machines (using wire-saw and rotating blade) of ingot of semiconductor, and an etching process. Such manufacturing process can be recognized as a sequence of manufacturing activities changing of attributes of target objects. So, the device ontology can cope with them. Although a functional structure of a specific product is different from manufacturing activities for the product, the functional concepts can be used in both models as a vocabulary (concepts) because manufacturing activities can be recognized as functions of manufacturing devices. In our experiments, these models mainly use functional concepts representing changes of attributes of objects especially numbers such as “to separate” and “to join”, force and motion.

In summary, 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, manufacturing processes and simple dynamic mechanical domains, though it does not cover static force balancing and complex mechanical phenomena based on shape of objects. The functional concepts about information are under more precise investigation. An investigation on functional concepts in different domains is in progress.

From these functional models of concrete systems, we extracted one hundred and ten ways of achievement in total for thirty-five base-functions. Such knowledge of ways of achievement is general and then applicable to many systems. In fact, the ways of achievement for “exerting physical force” are commonly used in some of the examples. For example, many of washing machines use the fluid way for exerting force on cloth, while in the slicing machine the same way is also used in order to remove scrapings. It means that the functional concept ontology provides us a common vocabulary for functional knowledge and it shows feasibility of reusable functional knowledge based on the functional concept ontology. It, however, is not easy to generalize ways in the concrete functional models, because we often recognize specific combination of primitive ways as a way and then we have to decompose it into primitive ways. An investigation on guidelines for description of ways is in progress.

The functional understanding system has been applied to a power plant and an oil refinery plant shown in Section 4. Although it is difficult to evaluate generated functional structures in nature, we confirmed all the consolidated macro-functions appear in textbooks of their plants. So, at this moment, the functional understanding system is applied to fluid-related plant concerning thermal energy and entity functions about temperature, phase, and ingredients of fluid. Heuristics for consolidation in Table 2 are general for domains and do not include domain-specific heuristics.

6. Related Work

6.1 Functional Concept Ontology

Some sets of “primitives of behavior” are proposed in [7,9,12,25]. We identified primitives of function and added more intention-rich (i.e., more teleological) concepts such as “remove” to them. Furthermore, we categorized functions into three fundamental categories and organized them in is-a hierarchies (Figure 4a,c and d) and part-of (is-achieved-by) hierarchies (Figure 4b). 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. The functional concepts in our ontology have operational definitions with clear relationship with behavior.

The features of our definitions of functional concepts can be summarized as (1) use of mapping primitives with behavior and (2) independence of implementation. By the former, we mean that the functional concepts are defined by additional information to behavioral models using functional toppings (FTs) as primitives and then the information specifies the mapping from behavior to function. It enables us to ground the functional concepts on behavior and then to realize the behavior-function mapping as discussed in Section 4.1.

Concerning the latter point, the implementation of a function should be viewed as a composition of (a) behavior and structure dependent part and (b) behavior structure-independent decomposition of a function into sub-functions. To maximize the freedom of functional interpretation and then generality of functional concepts, the dependency should be minimized. Thus, we distinguish the behavior-function relation (a) from the “is-achieved-by” relation between functions (b). Lind shares it with us and calls these relations “means-ends” and “whole-part”, respectively [7]. On the other hand, in CFRL [6], function is defined as a kind of hierarchical abstraction of behavior. Their functional models are described as sequences of partial states of the behavior, so that they depend on their implementations.

We also introduced a new type called meta-function. The CPD in CFRL [6] represents causal relations among functions. Lind categorizes such relations into Connection, Condition and Achieve [7]. Rieger identifies “enablement” as a type of the causal relation between states and action [26]. 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.

Horváth et al. discuss design concept ontologies for comprehensive methodology for handling design concepts in conceptual design, which include structure and shape as well as functionality [27]. For example, in the structural view, the connected entities are specified by positional, morphological, kinematical, and functional descriptors. We concentrate on functionality and then will categorize connections among devices according to their functions, that is, kinds of transmitting force or

motion in the mechanical domain in future work.

6.2 Automatic identification of functional structures

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 attributes 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 of the target system and then functional hierarchies which do not correspond to its structure can be generated.

The automatic identification of the Functional Representation uses templates of CPDs representing functional hierarchies as causal relations [16]. Thus, functional hierarchies are directly generated from the behavioral model without the functional concepts. The hierarchies built are also limited to those associated with structure. We detached interpretation of function from aggregation of function. 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 Figure 8 and 9) as well as using the predefined general pattern knowledge called the ways of achievements (Figure 10).

The consolidation theory [28] tries to capture the general rationales of consolidation of components. While we share the goal, their consolidation rules are simpler than ours and depend heavily 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.

7. Application to a Redesign Support System based on Functional Structures

The functional understanding system discussed in this paper is planned to be a part of an intelligent redesign support system. Here we overview the goal and effects of the system. The main goal is not to redesign the target systems automatically but to provide human designers various kinds of functional knowledge (ways of functional achievement shown in Figure 4b) across domains which can be a hint for possible alternatives for redesign. In order to provide appropriate functional alternatives for the original design, the functional structure of the original design is needed. However, the inputs for a redesign

system are usually structural and behavioral models of existing artifact. So, the redesign system has to bridge the gap between the given behavioral level and the functional level for reasoning. Thus, the functional understanding is needed as the first step of the redesign process as shown in Figure 13.

Such functional structures contribute to innovative redesign by humans. Given an existing artifact and a new requirement, change of the way of achievement a function enables us to propose new functional structures satisfying the requirement. For example, in a type of washing machine, the main function “to remove dirt from cloth” is achieved by friction caused by rotating a screw in the water. Given this functional structure and reduction of damage of cloth as a new requirement, the redesign supporting system firstly detects the candidates of causative ways of damage of cloth which includes the frictional way. Next, it provides alternative ways without friction including the centrifugal way to remove something as possible candidates for designer’s selection. If the designer selects the centrifugal way, we would get a new functional structure and a new product. This type washing machine has been on sale in Japan recently.

Such modification enables drastic improvement, in this case, the reduction of damage of cloth caused by the friction. It is enabled by generality of functional knowledge in different domains. Because many inventions are based on techniques well-known in different domains [29], if designers can consult a wide range of such functional knowledge in different domains, it facilitates innovative design. We are currently investigating such a knowledge-base system. TechOptimizer [30] is a software product based on a theory for innovative design [29], which contains generic principles of invention. It, however, just searches highly abstract principles for given criteria. It has no functional models of the target systems. Our system tries to provide appropriate alternatives for specific functional structures which are identified by the functional understanding system.

8. Summary

We proposed the functional concept ontology including the meta-functions, which contributes to solving the issues of functional understanding task mentioned in Introduction, that is, how to make the search in the functional space tractable 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

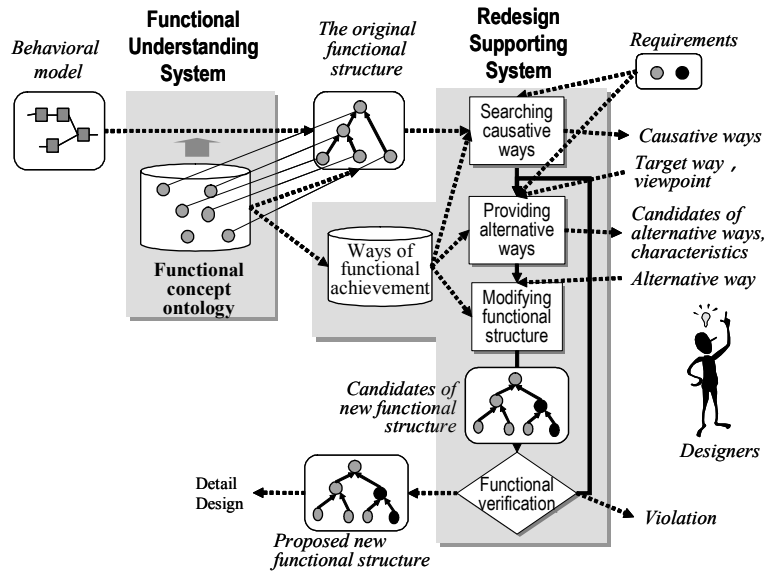


Figure 13. The framework of the redesign supporting system

can be specified by users, which enables the system to generate various functional hierarchies. An investigation on limitation of the ontology mentioned in Section 5 is in progress.

Acknowledgments. The authors are grateful to Mitsuru Ikeda for his valuable comments. Our thanks also go to Munehiko Sasajima, Kazuya Kawamura, Toshio Ueda, Kouki Higashide, and Masaru Takahashi for their contribution. 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., Ishii, M., Yoshioka, M., Shimomura, Y., and Tomiyama, T., 1996. Supporting conceptual design based on the function-behavior-state modeler. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 10, 275-288, 1996.
- [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. and Ranon, R., 1999. Automatic derivation of hierarchical representation for flow-based functional models, *Proc. of Tenth International Workshop on Principles of Diagnosis (DX-99)*, 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] Mizoguchi, R., and Kitamura, Y., Foundation of Knowledge Systematization: Role of Ontological Engineering, *Industrial Knowledge Management - A Micro Level Approach*, Rajkumar Roy Ed., pp.17-36, Springer-Verlag, London, 2000
- [19] de Kleer, J., Brown, J. S. 1984.: A Qualitative Physics Based on Confluences, *Artificial Intelligence*, 24, 7-83.
- [20] Forbus, K. D. 1984. Qualitative Process Theory, *Artificial Intelligence*, 24, 85-168, 1984.
- [21] Bhatta, S. R., and Goel, A. K. 1997. A functional theory of design patterns. In *Proc. of IJCAI-97*, 294-300.
- [22] Kitamura, Y., and Mizoguchi, R. 1998. Functional ontology for functional understanding, *Papers of Twelfth International Workshop on Qualitative Reasoning (QR-98)*, 77-87.
- [23] Kitamura, Y., and Mizoguchi, R. 1999. Meta-functions of artifacts, *Papers of 13th International Workshop on Qualitative Reasoning (QR-99)*, 136-145.
- [24] Kitamura, Y., and Mizoguchi, R. 1999. An Ontological Analysis of Fault Process and Category of Faults, *Proc. of Tenth International Workshop on Principles of Diagnosis (DX-99)*, pp.118-128.
- [25] Hodges, J. 1992. Naive mechanics - a computational model of device use and function in design improvisation. *IEEE Expert* 7(1):14-27.
- [26] Rieger, C., and Grinberg, M. 1977. Declarative representation and procedural simulation of causality in physical mechanisms. In *Proc. of IJCAI-77*, 250-256.
- [27] Horvah, I., Kuczogi, Gy., Vergeest, J. S. M.: Development and Application of Design Concept Ontologies for Contextual Conceptualization, in *Proc. of 1998 ASME Design Engineering Technical Conferences DETC*, CD-ROM: DETC98/CIE-5701, ASME, New York.
- [28] Bylander, T., and Chandrasekaran, B. 1985. Understanding behavior using consolidation, *Proc. of IJCAI-85*, 450-454.
- [29] Sushkov, V.V. Mars, N.J.I., and Wognum, P.M., Introduction to TIPS: a Theory for Creative Design, *Artificial Intelligence in Engineering*, 9, 1995
- [30] Invention Machine Co., TechOptimizer, <http://www.invention-machine.com/products/techoptimizer.cfm>, 1999