

Ontology-based Description of Functional Design Knowledge and its Use in a Functional Way Server

Yoshinobu Kitamura and Riichiro Mizoguchi

The Institute of Scientific and Industrial Research, Osaka University

8-1, Mihogaoka, Ibaraki, Osaka, 567-0047, Japan

Tel: +81-6-6879-8416, Fax: +81-6-6879-2123, E-mail: {kita,miz}@ei.sanken.osaka-u.ac.jp

Abstract

In conceptual design of engineering devices, a designer decomposes a required function into sub-functions, so-called functional decomposition, using a kind of functional knowledge representing achievement relations among functions. However, such knowledge about functionality of engineering devices is usually left implicit because each designer possesses it. Even if such knowledge is found in documents, it is often scattered around technical domains and lacks consistency. Aiming at capturing such functional knowledge explicitly and sharing it in design teams, we discuss its systematic description based on functional ontologies which provide common concepts for its consistent and generic description. We propose a new concept named "way of achievement" as a key concept for capturing such functional knowledge. Categorization of typical representations of the knowledge and its organization as is-a hierarchies are also discussed. The generic concepts representing functionality of a device in the functional knowledge are provided by the functional concept ontology, which makes the functional knowledge consistent and applicable to other domains. We also discuss development of a design supporting system using the systematized knowledge, called a functional way server. It helps human designers redesign an existing engineering device by providing a wide range of alternative ways of achievement of the required function in a manner suitable for the viewpoint of each designer and then facilitates innovative design.

Keywords:

Knowledge modeling; Knowledge sharing; Ontology; Functionality; Design support

1. Introduction

Engineering design can be defined as mapping from a requirement specification at the functional level into a set of attribute values of concrete things. So, functionality plays a crucial role in the conceptual design of engineering devices [Pahl 88, Keukene 91, Chandrasekaran 93, Umeda 96]. For example, a designer often decomposes a required function into sub(micro)-functions, so-called functional

decomposition [Pahl 88]. As a result, a designer obtains a micro-macro hierarchy of functions, which represents how the required (macro-)function is achieved by sub(micro)-functions, as a conceptual skeleton of the product realizing the requirement. Because there are many methods to achieve a specific function in general, designers should select an appropriate one from many alternatives. Such an activity requires knowledge of how to achieve a function, which represents achievement relations among functions. We call such knowledge about functions functional knowledge. Because many inventions are based on techniques well-known in different domains [Sushkov 95], if designers are supported by a server of a wide range of such functional knowledge in different domains, their innovative design capability is augmented.

However, the design know-how including such functional knowledge used in the conceptual design phase is usually left implicit in spite of that the advancement of computer technologies has enabled easy access to information related to structure and/or shape of artifacts. General knowledge so-called "design catalog" also mainly focuses on mechanism concerning shape and link. Even if such knowledge is found in documents, it is often scattered around technical domains and improperly categorized. For example, a categorization of connection-methods found in a textbook published by an academic society is not consistent and ill structured, because it is categorized according to different properties. Moreover, the textbook includes many categorizations based on the non-fundamental characteristics which can be derived from deeper principles.

We focus on the following three causes for such inconsistency, non-reusability, and improper categorization. The first one is that different frameworks for conceptualization are used when people try to describe knowledge in different domains. They can capture functions (even behavior as discussed in Section 2) in various manners without clear definitions of a general framework. We need fundamental concepts for capturing target devices from functional viewpoint.

The second cause is there are several functional concepts (i.e., verbs representing function) without clear definitions as pointed out in Value Engineering [Tejima 81]. Common vocabulary is necessary for reuse of functional knowledge in different domains. Moreover, it should be machine

understandable in order to manage the knowledge. Nevertheless, only a few generic functions have been proposed in functional representation research to date [Pahl 88, Hodges 92, Chittaro 93, Lind 94]. They are at a very high level of abstraction and inflexible so that it is hard to incorporate designer's intention into the functional model. On the other hand, value engineering has developed a standard set of verbs representing function [Tejima 81]. Although it is well-organized, definitions of verbs are only for human consumption. We need a rich and comprehensive vocabulary of functional concepts with operational definitions.

The last cause of inappropriate functional knowledge representation is that structure (organization) of viewpoint-independent and consistent modeling of functional knowledge is not fully investigated yet. When designers select a method of achievement of a required function as mentioned before, the criteria for the selection depend on his/her own viewpoint and thus are different from each other. For example, a designer may decompose the required function "to connect two objects" into "to insert a screw into a hole" and "to tighten the screw" by selecting the "screw way". The other designer, however, may prefer "hook-fit clasp" in which the objects are connected by "to hook the crook" from the viewpoint of manufacturing (assembling) of the product, because manufacturing in the case of "hook-fit clasp" is easier than that of the screw way. In conventional knowledge description, such knowledge is used to be described for each viewpoint, and the dependency on viewpoints is implicit. We need a systematic way of description of the knowledge and treatment of such viewpoints.

The needs for consistent and sharable description of functional knowledge can be summarized as fundamental and generic concepts for capturing and describing the functional knowledge. Such specification of conceptualization is generally called an "ontology" [Gruber 93, Mizoguchi 00]. In other words, we need such ontologies that guide conceptualization of artifacts from the functional point of view.

We have developed functional ontologies including a device-centered ontology and a functional concept ontology, aiming at systematization of functional knowledge for design [Mizoguchi 00, Kitamura 00]. The role of the former is to provide a consistent viewpoint for capturing the target artifacts in order to solve difficulties caused by the first cause of inconsistency and dependency on domains. That of the latter is to provide well-defined concepts for description of knowledge, and then to give a basis for systematization of knowledge.

The aims of this article include to discuss knowledge description based on those functional ontologies and to show the effects of those ontologies in knowledge systematization. Firstly, for the first cause of inconsistency, we overview the device ontology and define fundamental concepts such as "device" and "behavior" for consistent viewpoint for capturing target devices. Section 3 describes a framework of functional modeling and the functional

concept ontology as a solution to the second cause.

As a theory for solving the third cause, in Section 4, we propose a new key concept for capturing functional knowledge concerning achievement of functions, called way of function achievement. In Section 5, categorization of typical representations of the knowledge and its organization in *is-a* hierarchies are discussed. Such a categorization helps us describe viewpoint-independent knowledge and explicitly deals with viewpoints for knowledge organization.

In Section 6, we present successful deployment of our framework into a production company and analyze the success factors. Section 7 shows an implementation of the ontologies and an ontology-based environment for description of functional knowledge.

In Section 8, we discuss use of the knowledge based on the ontology. We show a knowledge-based system called a functional way server to help human designers decompose the required function in the conceptual design phase by providing various ways of achievement of the function. The functional concept ontology and the systematized functional knowledge play crucial roles in these systems.

Section 9 discusses related work and limitations of our ontologies and application domains followed by concluding remarks.

2. Device Ontology

The device ontology specifies the device-centered view of artifacts which regards any artifact as composition of *devices* which process input to produce output which is what the users need. The device ontology imposes a frame or viewpoint on physical processes in the physical process ontology [Forbus 84] to introduce an engineering perspective. That is, it introduces the concepts of a black box equipped with input and output ports, which is called a device. 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.

Device ontology specifies the roles played by the elements that collectively constitute a device. The concept of role is a hot topic in ontological engineering because an object plays different roles in different situations, and the fact has been a major source of failure in conceptualization of the world [Mizoguchi 00].

A naïve idea of device ontology was born in system engineering. It is composed of components and connection between them and has been extensively used in many engineering areas as well as in design community as is discussed in [Pahl 88]. However, it has no criterion of which role should be played by which component and the

assumptions behind the ontology is implicit, and therefore modeling of artifacts based on that ontology can be ad-hoc. Even worse, it is hard to compare it against other ontologies and its limitations are not clear. De Kleer and Brown introduced an idea of conduit into the naïve device ontology [de Kleer 84a]. His ontology still leaves the identification of objects unclear. The authors propose an extended device ontology based on Ontological Engineering which includes four different concepts of behavior, introduces concept of *medium*, and covers mechanical domains also [Mizoguchi 00]. The following is summary of the definitions.

Device, object, conduit, medium, and behavior

A *device* has input and output ports through which it is connected to another device. A *device* consists of other devices of smaller grain size and then is organized in a whole-part hierarchy of sub-devices.

An *object*¹ is something that can be considered as what it goes through a device from the input port to the output port during which it is processed by the device. What can play the role of *object* includes substance like fluid, energy like heat, motion, force, information, etc. An *object* has attributes whose values change over time.

A *device* is something that operates on an *object* that goes through the device. 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. What happens inside the device is invisible at that grain size. Although it seems to be visible when going down into the level of smaller grain size, what is seen there is different from what would be seen when one looks into a device at the same grain size.

A *conduit* is defined as a special type of *device* that can be considered as it transmits an *object* to output port without any change in an ideal situation. Examples include a pipe, a shaft, etc. We exclude conduit from device, because the transmitting function is performed by all devices and then a conduit can be recognized that it has no specific function. Moreover, it can act as “medium” as discussed in the following.

A *medium* is something that holds an *object* and enables it to flow among devices. For example, steam can play the role of a medium because it can hold heat energy. In some domain, conduit can play the role of medium. For example, while a shaft can be a conduit for rotating motion, at the same time, it plays the role of medium of rotation by rotating.

We identified four kinds of definitions of *Behavior*. Figure 1 illustrates simplified situations for behavior definition. **B0 behavior** is defined as the change of an attribute value of an *object* at the same location over time. Typical example is change of the temperature of fluid at some observation point over time. Note that what is observed is a different thing at any time. This is exactly same as the observation of a real phenomenon and coincides with what numerical

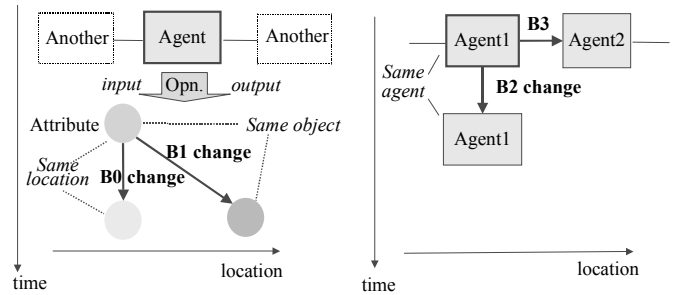


Figure 1 - Four different definitions of behavior

simulation obtains.

B1 behavior is defined as the change of an attribute value of an *object* from that at the input port of a device to that at the output of the device. For example, the increase of the temperature of steam occurred during it goes through a super-heater is B1 behavior. The key difference between B0 and B1 is that while B0 behavior concentrates on the location of the observation rather than identity of the observed *object*, B1 behavior on the identity of the observed *object* rather than the location.

B2 behavior is defined as the change of something inside of a device rather than input/output ports. The “something” could be *motion* of a part of the device or inner state of the device. For example, “rotation of fins in a fan” is an inner behavior of “to fan”, “a shaft is twisted” is an inner behavior of “to transmit torque”, etc.

B3 behavior is defined as any behavior to another *device*. The phrases “a cam shrinks a spring” and “a cam pushes up a rod” are the examples of B3 behavior. The important aspect here is *B0* and *B1* behaviors are concerned with *objects* rather than *devices*.

Another definition of behavior, which looks very similar to B1 behavior at first glance, is found in [Chandrasekaran 93] where function is defined as *B1 behavior* and the behavior corresponding to the function is defined as series of *B1 behavior* of sub-devices of smaller-grain size.

In our extended device ontology, we define “behavior” as *B1 behavior*, because it represents “effect” of the device on objects flowing among devices and then it matches with our concept of function as discussed below. In the rest of this article, we use the term “behavior” instead of *B1 behavior* for simplicity.

Role of the device ontology

The aim of the device ontology is to impose a proper viewpoint from which one can successfully model a system in various domains in a way consistent with each other. It is not an easy task to build models of a lot of artifacts in a consistent way. For example, “a gear pair changes torque”, “a cam shrinks a spring” and “a cam pushes up a rod” are inconsistent with each other in the hidden computational models. While the first one is B1 behavior based on the extended device ontology, the latter two are B3 behavior based on a different ontology, say, inter-device operation ontology. The organization of knowledge including these models will lose consistency.

¹ The label of this concept “object” represents neither “entity”, “substance” nor “purpose” but “target” of activity of a device.

Use of the same framework for capturing the target world is one of the necessary conditions for consistent and interoperable models. The extended device ontology allows us to build interoperable models without inconsistency and provides us with a guideline for modeling process.

Note that our claim is **neither** that the device ontology enables all kinds of description of all kinds of artifacts, all models should be described on the basis of the device ontology, **nor** the device ontology is a unique solution. We claim that a solid foundation like this device ontology is needed for systematic description of functional knowledge.

3. Framework of Functional Modeling and the Functional Concept Ontology

Modeling Framework

In our functional modeling framework [Sasajima 95, Mizoguchi 00, Kitamura 02], the model of artifacts consists of a behavioral model and a functional model as shown in Figure 2. According to the device-oriented ontology, at the behavioral level, “structure” of the behavioral model describes the existence of devices (components), topological connections among them, and a micro-macro hierarchy among components and (sub-)systems. The behavioral model of components represents changes of attribute values of objects which flow through the component (B1 behavior). The structural micro-macro relations represent hierarchical organization of behavior.

On the other hand, at the functional level, a (base-)function of a device is defined as a result of teleological interpretation of a behavior of the component under an intended goal [Sasajima 95]. 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 2, 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 the Functional Toppings (FTs) of FBRL [Sasajima 95]. The functional toppings represent information about teleological interpretation of behavior according to the designers’ intention. 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

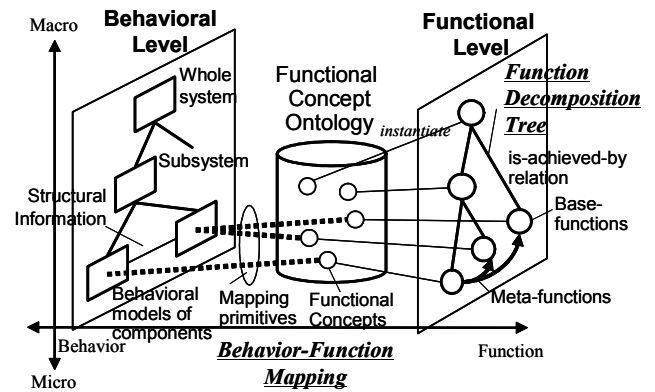


Figure 2 - Framework of functional models of artifacts

[Sasajima 95].

The micro-macro relations among functions represent that the macro-functions are achieved by the sequences of sub(micro)-functions (called achievement relations). We also call such a tree (hierarchy) a *function decomposition tree*. The next section discusses this relation in detail.

Functional concept ontology

The functional concept ontology [Kitamura 00,02] specifies generic functional concepts using FTs. The functional concepts are categorized into *base-function* of devices, *function type* and *meta-function* representing interdependency among base-functions. Currently, it includes about one hundred base-functions, four function type, and eight meta-functions [Kitamura 99,02]. The definitions are scarcely depends on the device, the domain or the way of its implementation so that they are very general and usable in a wide range of areas. The detail of definitions of functional concepts is discussed in Section 7.

Roles of the functional concept ontology

The functional concept ontology specifies the space of functions and limits functions within the generic functions defined in the ontology. It enables us to map functional concepts with behavior automatically and to identify plausible functional structures from a given behavioral model [Kitamura 00,02]. In fact, our automatic identification system screens out the candidates of functional interpretations which match with no functional concept in the ontology on the assumption of comprehensiveness of our ontology.

Although our ontological approach may reduce the freedom of functional representation in comparison with hand-written functional models, it enables us to avoid ad hoc modeling and obtain consistent functional models. It also contributes to reusability of functional knowledge in terms of the concepts in the functional concepts ontology.

4. Ways of Function Achievement

As mentioned in Introduction, although knowledge about

function achievement plays a crucial role in conceptual design, the conventional description of such knowledge tends to be scattered around technical domains, to lack consistency of viewpoints, to be ill-structured, and/or to be categorized by the non-fundamental characteristics which can be derived from deeper principles.

The key points of our systematization are;

- (1) Common functional concepts provided by the functional concept ontology
- (2) The concept of the way of function achievement
- (3) Categorization of the knowledge

The first one as discussed in the previous section provides us with common vocabulary for representation and then makes the functional knowledge reusable. The second one enables us to explicate the background knowledge of function achievement and thus gives us key concepts for organization of the knowledge, which realizes consistent categorization. The last one enables us to distinguish the inherent one from derivable ones, and thus makes the knowledge well-structured.

This section describes the second point, way of function achievement. The last point is discussed in Section 5.

The concept of “way of function achievement”

Figure 3 shows a top-most part of a function decomposition tree of a power plant as an example. The required function “to generate electricity” is decomposed into “to generate torque” and “to generate electricity”. The former sub-function is further decomposed into more micro-functions including “to vaporize water” and “to heat water”, “to generate torque” and “to condense steam”. Thus, the function achievement relations of artifacts discussed above are results of the functional decomposition in the conceptual design process.

Such a traditional functional decomposition model lacks information at the behavioral level and represents only “how” the macro-function is achieved but does not represent “why” the sequence of the sub-functions can achieve the macro-function. .

Here we introduce the concepts of *method of function achievement* and *way of function achievement*. When a macro-function can be achieved by a sequence of sub(micro)-functions, we call the sequence of sub-functions constrained by the relations among them *a method of the achievement*. On the other hand, we call the background knowledge of functional decomposition such as physical principles, theories, phenomena, and structure as the basis of the achievement *a way of the achievement*. For example, in Figure 3, the basis of the second functional decomposition can be represented as “thermal steam way” whose intended phenomena is (adiabatic) heat expansion of gas. Such a way of achievement of a function can represent partial information of behavioral and/or structural one.

General knowledge of a way of function achievement

We call general knowledge of achievement of a function

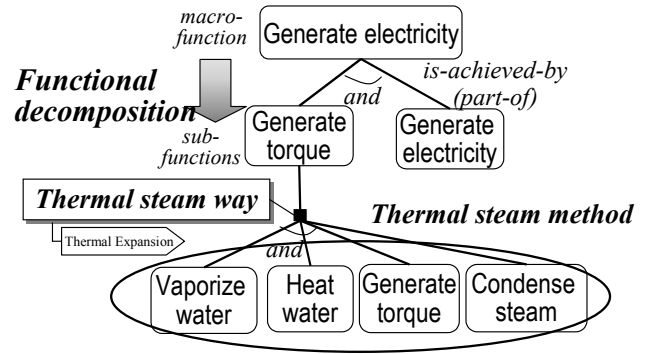


Figure 3 - A function decomposition tree of a power plant (portion)

the *ways of function achievement*. A description of a way of achievement of a function consists of the function as a macro-function, a set of sub(micro)-functions, temporal and causal constraints among sub-functions, principle of the achievement, conditions for use of the way, and characteristics of products using the way. Although it includes description of the method of function achievement, we call it *a way of function achievement*, focusing that it includes description of principle of the achievement.

The principle of achievement represents physical principle or physical phenomena in order to achieve the macro-function. It governs the characteristics of the product using the way. For example, in the chemical way for connection, the chemical cohesiveness determines that the product cannot be disassembled without disrupt. The characteristics are described in a qualitative manner.

The macro- and micro-functions are described in terms of general functional concepts which are provided by the functional concept ontology [Kitamura 00]. Each of the provided functional concepts has general meaning of a base-function independent of target objects and components. Thus, the knowledge in terms of them can be applicable to other domains.

Role of the concept of “way of achievement”

The concept of way of achievement enables us to detach “how to achieve” (way) from “what is achieved” (function). For example, “to weld steel” is not just a function but function with a way in which the steel is melted. It should be decomposed into the “connecting function” and “fusion way”. This increases generality and capability to cover wide range of ways such as the bolt and nut way as an alternative to connect.

5. Categorization of Knowledge of Function Achievement

We categorize typical representations of the knowledge into an inherent knowledge based on their principles and the other representations derivable from the inherent one as follows. Figure 4 shows the categorization and their examples.

The is-a hierarchy of ways

The ways of achievement of a function are organized as an *is-a* hierarchy according to their principles. Because the principles are inherent properties of ways, we can recognize it as a right *is-a* hierarchy. For example, in Figure 4, ways of connection are categorized into mechanical, chemical, and physical ways according to the principles for constraints of positions followed by further specialization. The other kinds of knowledge can be derived by reorganizing this *is-a* hierarchy. Note that this *is-a* hierarchy represents abstraction of principles in order to achieve each macro-function, that is, key information about how to achieve the function. On the other hand, the *is-a* hierarchy of functional concepts shown in Figure 6 represents abstraction of functions themselves, that is, what goal is achieved.

Attributes tree

For selection of ways, ways are classified according to their attributes in a form of so-called decision tree. Each leaf node represents a way. Its structure depends on the viewpoints. In Figure 4, an attribute tree from the viewpoint of disassembly is shown. From the viewpoint, the important characteristics of a product using a specific way of connection include whether the product can be disassembled without disrupt or not and whether it can be done without deformation or not. Such characteristics appear as branch nodes in the tree. Each of the ways of connection is classified by values of such characteristics and appears as a leaf node. This type of trees is sometimes confused with *is-a* hierarchy. They are the same in the sense that both are classification of ways, but are different from each other in that while the structure of the decision tree can be changed according to the purpose of the classification of interests and the intermediate nodes do not have to correspond to meaningful and reasonable categories of ways, the structure of *is-a* hierarchy is determined by its

inherent property and hence it is unique. Confusion of this difference has been one of the causes of the inappropriate organization of way knowledge we have used to date.

Function decomposition tree

This is a kind of a product model at the functional level like one shown in Figure 2. It consists of sub-functions and description of the way as shown in Figure 3. This tree is constructed as a result of decision of the product designer. In Figure 4, two pipes are being connected by motion transmitting function (precisely, motion in horizontal direction in the figure) of the crook and the clasp, because the same motion of the two pipes implies the same (fixed) distance between them.

Manufacturing activity decomposition tree

It consists of manufacturing activities (process) of a product. This tree of a product and the functional decomposition tree of the product are usually different from each other. In the case of static functions such as connection, however, the principles of the ways in the both trees are often the same. Note that manufacturing activities of a product can be recognized as functions of a manufacturing device for the product. Thus, the common functional concepts can be used in the both trees.

General function (or manufacturing activity) decomposition tree

It consists of all possible ways of achievement of a function (or manufacturing activity) in *OR* relationship. It is a kind of functional decomposition trees independent of a specific device. All the *OR* branches in a path from the root to a leaf node in a tree of this type collectively define a *composite way* which is combination of some primitive ways defined in the *is-a* hierarchy as shown in Figure 4. This type of tree is not stored in the system as it is. Like attribute trees, it is also generated by connecting each piece of functional concepts organized in an *is-a* tree upon request. In general,

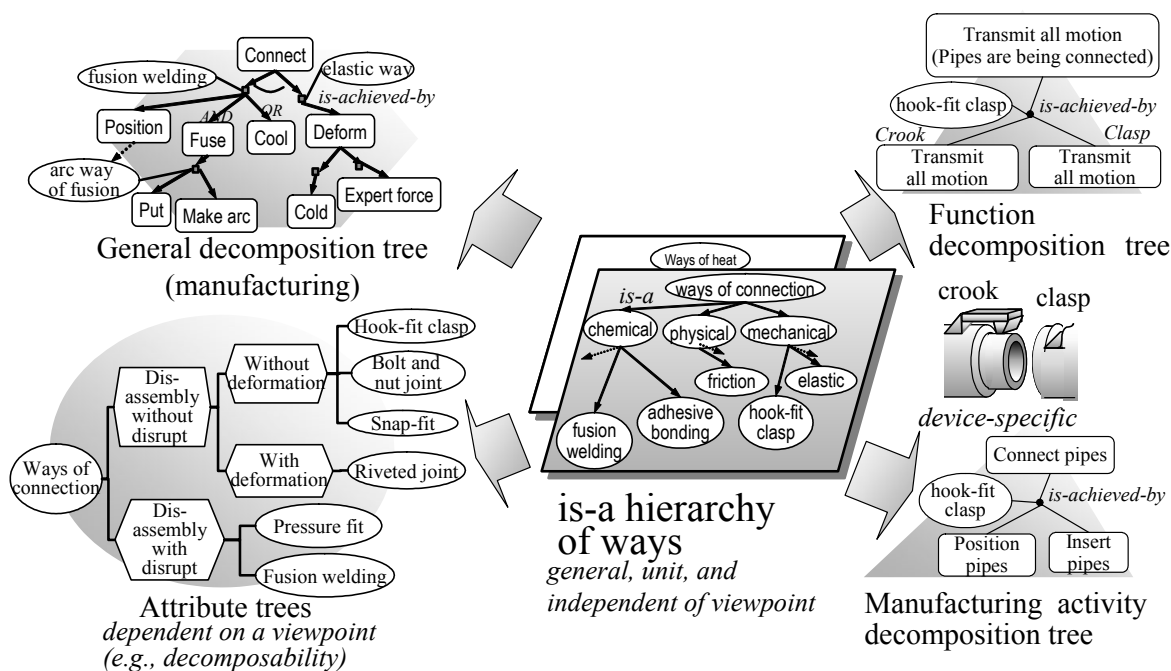


Figure 4 - Categories of knowledge of ways and their examples

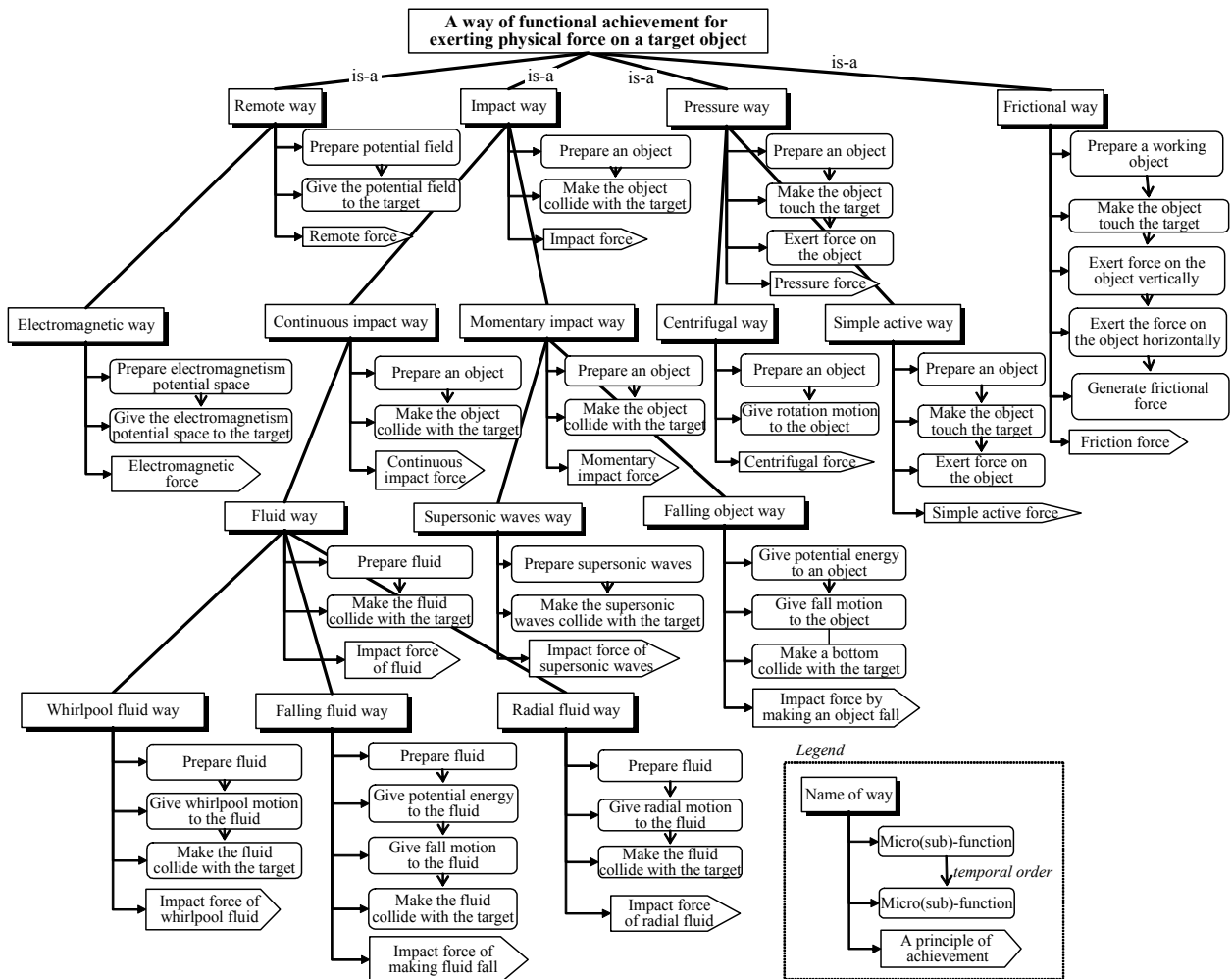


Figure 5 - The knowledge base of ways of achievement for “to exert physical force” (is-a hierarchy)

the ways described in textbooks are often composite one. For example, “the arc welding way” shown in a textbook is a composite of two primitive ways; “the fusion welding way of connection” and “the arc way of fusion” as shown in Figure 4. Note that the second way is not the way of connection but of fusion which is a micro-function of connection. These composite ways often cause an inappropriate structure of conventional organization of way knowledge. In our framework, they can be properly described as such composite ways in the general functional decomposition trees.

6. Deployment of the Framework

Deployment in production systems and evaluation

Our framework is being deployed in Production Systems Engineering Division of Sumitomo Electric Industries for sharing functional design knowledge of production systems. After one year study of our theory, they started test use in May, 2001. They and we described function decomposition trees of about 15 production facilities in production systems for semiconductor. The preliminary evaluation by the engineers was excellent. They said that this framework

enabled them to explicate implicit knowledge possessed by each designer and to share it among team members. It was easy for designers to learn the framework based on the device ontology. They decided to deploy it and started the development of a knowledge collecting software.

The success factors of the deployment are summarized as follows; (1)clear discrimination between function (goal) and way (how to achieve the goal) which contributes to reusability of the knowledge, and (2)clear discrimination among *is-a* and *part-of* relations, that is, the *is-a* hierarchy of function and that of way, and the *is-achieved-by* (a kind of *part-of*) hierarchy of function, and (3)explicit viewpoint specification by the extended device ontology.

Building a knowledge-base of ways

We have described generic 104 ways of achievement of 26 functions from five examples; a washing machine, a printing device, slicing machines for ingot of semiconductors (using wire or rotating blade), and an etching device. The last three was obtained in cooperation with Sumitomo Electric Industries. Firstly, we described functional decomposition trees of them. Next, we generalized the ways appearing in the examples. Then, we

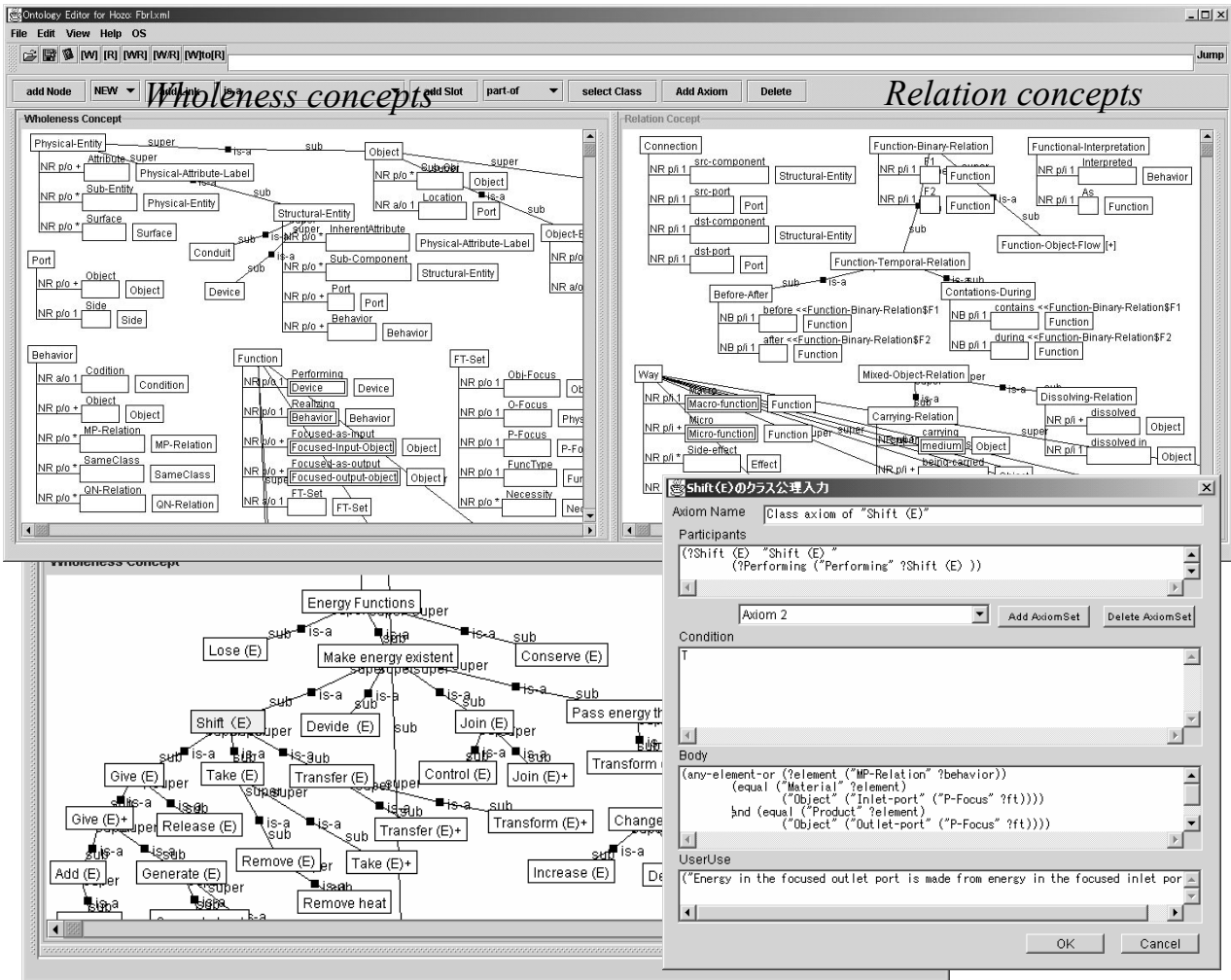


Figure 6 - The implemented device ontology (the above) and the functional concept ontology (the below) (portion)

tried to find the underlying principles and then organized them into is-a hierarchies as discussed above. Lastly, we added other ways based on the principles extracted.

As an example, Figure 5 shows an is-a hierarchy of ways of achievement for “exerting physical force”. In the figure, a box, a round box, and a pentagon represent a concept of a way, a sub(micro)-function of the macro-function in the way, and a principle of the way, respectively. The ways for exerting physical force are categorized into four types according to types of force; *remote force*, *impact force*, *pressure force*, and *friction force*. The *impact way* is furthermore categorized into sub-types: the *continuous impact way* and the *momentary impact way* according to length of the time interval of application of force.

All the sub-functions shown in Figure 5 can be further decomposed into finer sub-functions according to other *is-a* hierarchies of ways of achievement for each sub-function in the same manner. The results of decomposition can be used to form a functional decomposition tree or a general functional decomposition tree as discussed in the previous section. Thus, is-a hierarchies of ways of achievement can generate “is-achieved-by” relations (a kind of part-of relations) among functions.

The knowledge of ways of achievement is general and then applicable to many systems. In fact, the ways shown in Figure 5 are commonly used in some of the five examples. For example, many of washing machines use the *fluid way* for exerting force to cloth, while in the slicing machine the same way is also used in order to remove cutting dust.

7. Ontology-based Environment for Description of Functional Knowledge

The framework of description of functional knowledge discussed thus far has been implemented using our environment for building and using ontologies named Hoza [Kozaki 00]. The ontology editor of Hoza helps ontology authors define concepts in ontologies as class-frames and describe instance models based on the ontologies by instantiating the classes. In this environment, a class-frame of a concept consists a label of the concept, a super-concept in the is-a hierarchy, slots of part concepts (part-of relation denoted by p/o), slots of attributes (a/o), and axioms. Definition of each slot consists of a class constraint for slot values (instances) and a role of the slot values in the frame’s context. Concepts are categorized into the

wholeness concepts composed by the part concepts and the relation concepts between the concepts. For more details, see [Kozaki 00].

We implemented our functional ontologies using the ontology editor of Hozo and then can use Hozo as an environment for description of functional knowledge and functional models. Note that Hozo is not specific to the functional ontologies but a general tool for ontologies. Thus, although its user interface is not helpful for naive authors who are not familiar with ontologies, it helps demonstrate fundamental effects of functional ontologies for description of functional knowledge.

Implementation of functional ontologies

Figure 6 shows a portion of the implemented functional ontologies in Hozo. The upper window shows definitions of the device ontology as a modeling language of function and behavior named FBRL (Function Behavior Representation Language) [Sasajima 95]. It includes the class-frames of device, behavior, function, and functional toppings (FTs). For example, a device is a sub-concept of a structural entity and consists of physical attributes, sub(part)-components representing whole-part relations among devices, ports for connection with other devices, and its behaviors. The FTs represent information of teleological interpretation of a behavior and then represent mapping from the behavior to a functional concept. A set of FTs is composed of four items, Obj-Focus, O-Focus, P-Focus and Necessity (as denoted by p/o slots in the figure) as mentioned in Section 3. They enable us to define functional concepts explicitly in machine understandable forms as follows.

The lower window in Figure 6 shows a portion of the functional concept ontology [Kitamura 99, 00]. It consists of base-functions for functions of devices, function types (we redefined those in [Keuneke 91]), and meta-functions representing collaborative roles in interdependency among base-functions. The base functions are categorized by kinds of target objects such as substance, energy, information, force and motion. The lower window in Figure 6 shows a portion of an is-a hierarchy of the energy-related base-functions.

Each concept in the hierarchies is defined using FBRL and hence operational. For example, an energy function, *To shift energy*, is defined as a behavioral constraint: *focused energy moves between two different mediums*. It can be defined by the following three axioms;

- P-Focus on an inlet port and an outlet port.
- Energy in the focused outlet port is made from energy in the focused inlet port.
- Mediums of the focused energies are different.

The right window in Figure 6 shows the window for describing the second axiom. The “participant” field defines arguments (parameters) appearing in the axiom body, “axiom body” defines constraints over arguments. The constraint should be true for all instances of the class if the “condition” is true (the symbol “T” means “always

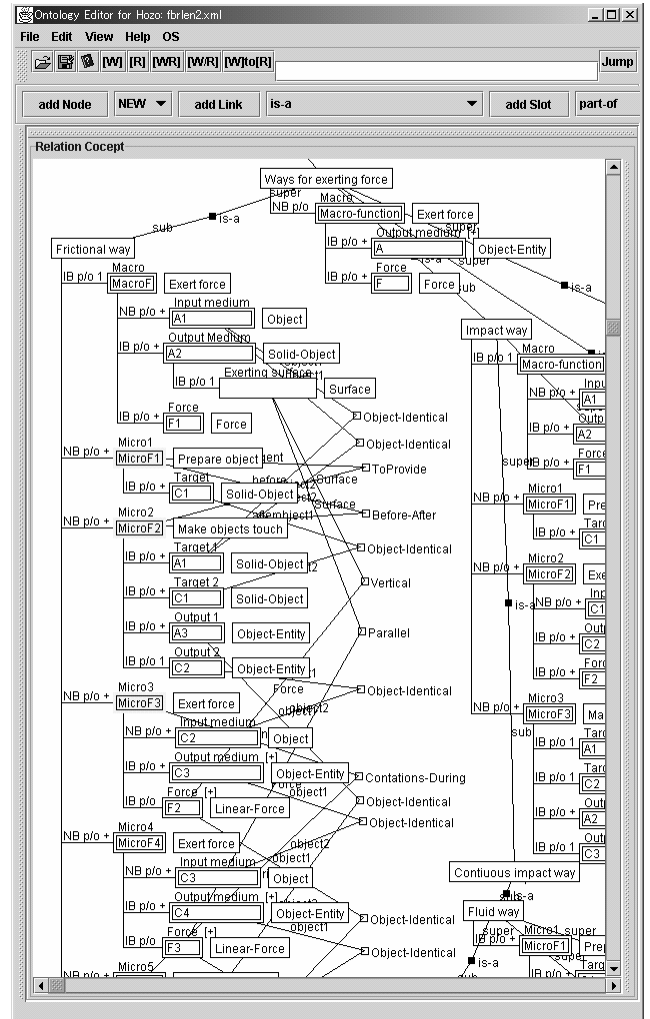


Figure 7 - An example of implemented ways

true”). The “UserUse” field can be used for comments.

To take, a sub-function of *to shift* in the is-a hierarchy, is defined as that of *to shift* with an additional FT, P-Focus on the port of energy provider. Moreover, *To remove* is defined as that of *to take* with an additional FT, the energy taken is unnecessary as *Necessity* FT.

These definitions demonstrate high independence of their implementation, while function is clearly related to structure and behavior.

Description of ways

A way of functional achievement is defined as a subclass of the way relation-concept. The class of “way relation” is defined in the device ontology shown in the upper window of Figure 6. It consists of a macro-function, micro-functions, principle for the functional achievement and side-effects.

Figure 7 shows an example of the implemented ways. It shows an implementation of the frictional way for exerting physical force in Figure 4. Its macro-function is “to exert force” and its sub-functions including “to make an object touch another object” and “to exert force to the objects”. There are (should be) relations among sub-functions such

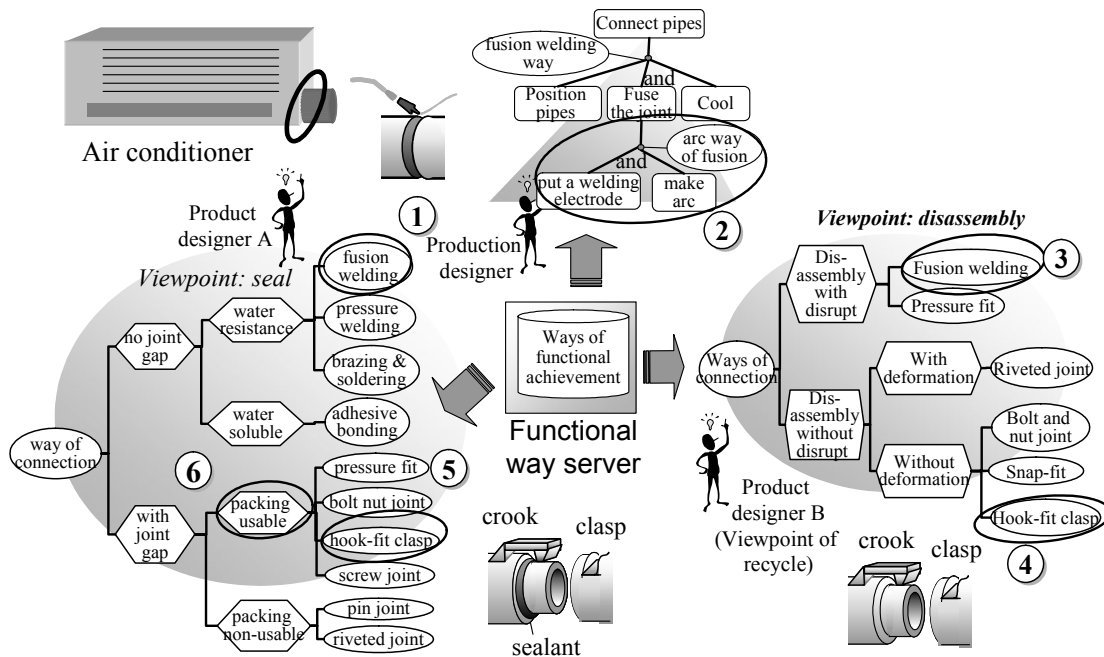


Figure 8 - A scenario of design using the functional way server

as temporal relation (in the figure, Contains-During²) and meta-functions (ToProvide).

8. Functional Way Server

This section shows a design support system named a functional way server as an application of the functional knowledge described based on the ontologies. The server is designed to support conceptual design of engineering devices, providing suitable ways of achievement of the function that designers consider. The server contains the knowledge of ways of function achievement of as *is-a* hierarchies. Given a required function and a viewpoint of the designer, the server reorganizes the knowledge in a manner suitable for the viewpoint and then shows ways in the form of an attribute (decision) tree described in Section 5. A viewpoint represents a context of designer's thinking process. It consists of a phase in the product lifecycle (design, manufacturing, use, and recycling), focused attribute set, grain size, and a domain of interest such as mechanical and electrical.

Here we show a possible scenario of design using the functional knowledge server (see Figure 8). Let us suppose a concurrent design team is designing a connection between an air-conditioner and a pipe. Firstly, a product designer A is selecting a way of connection from the viewpoints of seal in order to seal the coolant in the pipe. Given the viewpoint "seal", the server looks up suitable attributes such as "joint gap" and "water resistance" associated with the viewpoint. According to the values of such attributes, the server classifies suitable ways, then shows the designer A an attribute-tree shown in the left part of Figure 8. If the product designer would select the "fusion welding" way (shown as (1) in Figure 8), a functional decomposition tree

and a manufacturing activity decomposition tree are constructed (the latter is shown in the upper part of Figure 8). A manufacturing designer can decompose the manufacturing-activity by selecting a way of fusion, say, the arc way (2).

Next, suppose that other designer B from the viewpoint of recycling checks the connection. The server derives the attribute tree shown in the right part of Figure 8 from the viewpoint of disassembly using the same knowledge of ways. If the designer B would submit an alternative plan "to hook-fit clasp" (4) in consideration of necessity of disruption of the connection by the fusion welding way (3), the product designer A would recognize the characteristics of "hook-fit clasp" on seal (5) and thus add a sealant in order to seal the coolant (6).

The server provides a wide range of ways in different domains and then facilitates innovative design. Moreover, the server can show viewpoint-dependent organization of the functional knowledge by reorganizing the viewpoint-independent knowledge. It helps us avoid the third cause of inappropriate functional knowledge discussed in Introduction. Because the server can track effects of decisions made by both the product designer and the manufacturing designer, it facilitates negotiation in a concurrent design team.

9. Related Work and Discussion

Ontology of artifacts

The importance of explicit specification of conceptualization (i.e., ontology) in artifact modeling is widely recognized in literature such as [Abu-Hanna 94; Borst 97; Horváth 98; Salustri 98; Chandrasekaran 00]. As pointed out these papers, ontological statements

² This notation is based on that of [Allen 84].

(commitment) can be used as simplifying assumptions to improve the robustness, complexity, and computability of knowledge representation of artifact models, and to avoid misinterpretation of them.

Borst et al. propose a hierarchy of ontology for designing an artifact which shares a lot of with our work [Borst 97]. The major differences include that Borst's ontologies do not include one for function which is our main issue and that his device ontology is weak in that it does not have enough concepts for understanding the roles all the participants play comparing to our extended device ontology.

Bond graph is a theory for describing a system domain-independently in the field of system dynamics [Rosenberg 83]. It introduces the concept of "flow" that represents amount of something which flows in the device and the concept of "effort" which has capability to cause the flow. Although it is elaborated well, it still lacks the expressive vocabulary and device ontology to specify viewpoint in modeling an artifact. In the flow-based functional modeling approaches [Chittaro 93; Lind 94], primitive functions are proposed based on such generalized flow concept. They are included in our functional concept ontology based on the extended device ontology.

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 [Horváth 98]. For example, in the structural view, the connected entities are specified by positional, morphological, kinematical, and functional descriptors. We concentrate on functionality and then categorize connections among devices according to their functions, that is, kinds of transmitting force or motion in the mechanical domain.

Chandrasekaran and Josephson try to clarify many meanings of the concept "function" and discuss relationship between environment-centric functions and device-centric functions based on ontological consideration [Chandrasekaran 00]. Although we share the distinction between the two major views of functions and the attitude towards the ontological analysis with them, we concentrate only on the device-centric viewpoint in this paper. We define meaning of fundamental concepts such as "device" and "object" as the roles played by the entities behind the device-centric viewpoint.

Definition of functional concepts

Although there are quite amount of research on functional representation, many of researchers focus on modeling framework of an individual (instance) component or system. The conventional functional knowledge uses "vocabulary" representing functionality [Sembugamoorthy 86; Umeda 96]. Nevertheless, the meaning of terms in such vocabulary is implicit or specific to an individual (instance) component and hence, the contents of such knowledge depend on the modeler and thus tend to be ad hoc.

In our ontological approach for functional concepts, all

functions in functional models of target systems and functional knowledge are instances of generic functional concepts in the functional ontology. They are carefully articulated on the basis of the extended device ontology, defined explicitly as discussed below, and organized in *is-a* hierarchies. The ontology provides functional knowledge with rigid functional concepts from a consistent viewpoint.

The features of our definitions of functional concepts can be summarized as follows;

- (1) Use of mapping primitives with behavior,
- (2) Discrimination between the teleological interpretation and the macro-micro relation, and
- (3) Discrimination from ways of achievement

By the first point, 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. Value engineering research [Tejima 81] lacks such operability. FTs represent designers' intention including focus on objects and necessity of objects, and then enable us to define intention-rich functional concepts. De Kleer defines function as a causal pattern between variables in his early work on teleological analysis [de Kleer 84b].

Concerning the second point, we discriminate between the macro-micro hierarchy (*is-achieved-by* relation) and the teleological interpretation (the behavior-function relation). Although this distinction is shared with some researches [e.g., Lind 94; Umeda 96], our definitions are independent of difference of realization of function. On the other hand, FR framework [Sembugamoorthy 86] defines functions as a kind of hierarchical abstraction of behavior and treats the difference of behavior and function as relative relation. Because a function can be achieved by some different sets of micro-functions in different ways, such discrimination is important to keep freedom of conceptual design at the functional level. Moreover, dependency of definition of a functional concept on its achievement reduces generality of the definition. Chandrasekaran and Josephson also point out the importance of implementation-independent functional models and then propose representation of function as effect [Chandrasekaran 96].

The last point helps us keep the functional concepts representing "what is achieved". For example, as discussed in Section 4, "to weld steel" is not just a function but a function implying a specific way of achievement. Such concepts should be decomposed.

Knowledge of function achievement (decomposition)

In the literature on design, general knowledge for functional decomposition or functional synthesis is proposed (e.g., [Bradshaw 91; Umeda 96]). The major differences between our ways of achievement and it include explicit description of "way", organization in *is-a* hierarchies based on principles of ways, and use of the functional concept ontology as follows.

Firstly, our ways of function achievement explicitly represent the feature of achievement such as theory and phenomena. They enable designers to facilitate the smooth interaction between the structural level and the functional level. The designer can check the feasibility of functional decomposition using the features represented as the ways as behavioral constraint. Such functional decomposition is compliant with the observations found in the research on design processes [Takeda 90] in which they say functional decomposition is not done solely in the functional space but done by going back and forth between the functional, behavioral and structural spaces during which some portions of the artifact are determined in each of the spaces simultaneously.

Secondly, we organized such general knowledge as an *is-a* hierarchy. Although the feature knowledge is also captured in [Malmqvist 97], it corresponds to the function decomposition tree of a specific product in our categorization and there is no organization of general knowledge. The crucial issue of organization as an *is-a* hierarchy is to capture inherent properties in order to avoid improper categorization as discussed in Section 5. The concept of way enables us to capture the principles of achievement of a function (i.e., why the function can be achieved) as the inherent properties and thus to realize consistent categorization.

Lastly, our functional knowledge is based on the functional concept ontology [Kitamura 00]. As mentioned in Section 6 and the above, we have identified generic functional concepts with clear operational definitions in *is-a* hierarchies. Use of such functional concepts as vocabulary of description of knowledge facilitates reuse of the knowledge in different domains.

Such systematization of functional knowledge based on the concept of way and the functional concepts enables the design support system to provide a wide range of alternative ways in different domains. Such knowledge in different domain can facilitate innovative design many of which are based on techniques well-known in different domains [Sushkov 95]. TechOptimizer [IMC 99] is a software product based on a theory for innovative design [Sushkov 95], which contains generic principles of invention. It, however, just searches highly abstract principles for given criteria. It is not adaptive for the designers' viewpoints, while our functional way server reorganizes the general knowledge according to the designer's viewpoint.

Limitation of our ontologies and application domain

We cannot claim completeness of the concepts in the functional concept ontology. Note that we defined not domain-specific functions but general functions common in many domains. In design research [Pahl 88], very few (from 3 to 24) generic functions are defined. 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 of function achievement. In fact, in Value Engineering research [Tejima 81], 158 verbs are proposed as a standard

general set for representing functions of artifact. Although it includes functions for human sense also, we concentrate on functions changing physical attributes.

The ontologies have been applied to modeling of a power plant, an oil refinery plant, a chemical plant, a washing machine, a printing device, and manufacturing processes. The device ontology is very appropriate for fluid-related plants like the first three which focus on thermal energy, flow rate, and ingredients of fluid. 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. We extended the device ontology to cope with force and motion and treats force and motion as a kind of objects. In the last application domain, 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.

In summary, on the basis of the extended device ontology, our functional concept ontology 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.

10. Summary

The contribution of this research can be summarized as a framework of systematization of design knowledge about functional decomposition. We pointed out three causes of inconsistent knowledge representation of such functional knowledge, that is, different frameworks for conceptualization, lack of common vocabulary for functional concepts, and viewpoint-dependency. For these causes, we proposed the device ontology in Section 2 and 7, the functional concept ontology in Section 3 and 7, and a categorization of functional knowledge based on the concept of "way of function achievement" in Section 4 and 5, respectively. The benefits of such functional knowledge were shown in Section 6 and 8.

We showed 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.

Acknowledgments

The authors would like to thank Toshinobu Kasai, Kouji Kozaki, Mariko Yoshikawa, Tomonobu Takahashi, Masaru Takahashi, Toshio Ueda and Kouki Higashide for their contributions. Special thanks go to Dr. Masayoshi Fuse, Sumitomo Electric Industries Ltd., for his cooperation for application to manufacturing processes. This research is

supported in part by the Japan Society for the Promotion of Science (JSPS-RFTF97P00701). Part of this research was part of GNOSIS project within the Intelligent Manufacturing Systems (IMS) research program.

References

- [Abu-Hanna 94] Abu-Hanna, A., and Janswijer, W., Modeling Domain Knowledge Using Explicit Conceptualization, *IEEE Expert*, 9(5), 53-63, 1994.
- [Allen 84] Allen, J.F., Towards a general theory of action and time, *Artificial Intelligence*, 23, 123-154, 1984.
- [Borst 97] Borst, P., Akkermans, H., and Top, J., Engineering Ontologies, *Int'l Journal of Human-Computer Studies*, 46(2/3), 365-406, 1997.
- [Bradshaw 91] Bradshaw, J. A., and Young, R. M., Evaluating Design using Knowledge of Purpose and Knowledge of Structure, *IEEE Expert*, 6(2):33-40, 1991.
- [Chandrasekaran 93] Chandrasekaran, B.; Goel, A. K.; and Iwasaki, Y., Functional Representation as Design Rationale, *Computer*, 48-56, 1993.
- [Chandrasekaran 96] Chandrasekaran B., and Josephson, J. R., Representing Function as Effect: Assigning Functions to Objects in Context and Out, In *Proc. of AAAI-96 Workshop on Modeling and Reasoning with Function*, 1996.
- [Chandrasekaran 00] Chandrasekaran, B., and Josephson, J. R., Function in Device Representation, *Engineering with Computers*, 16(3/4), 162-177, 2000.
- [Chittaro 93] Chittaro, L., Guida, G., Tasso, C., and Toppano, E., Functional and Teleological Knowledge in the Multimodeling Approach for Reasoning about Physical Systems: A case Study in Diagnosis, *IEEE Transactions on Systems, Man, and Cybernetics*, 23(6), 1718-1751, 1993.
- [de Kleer 84a] de Kleer, J. and Brown, J. S., A Qualitative Physics Based on Confluences, *Artificial Intelligence*, 24, 7-83, 1984.
- [de Kleer 84b] de Kleer J., How Circuits Work., *Artificial Intelligence*, 24, 205-280, 1984.
- [Forbus 84] Forbus, K. D., Qualitative Process Theory, *Artificial Intelligence*, 24, 85-168, 1984.
- [Gruber 93] Gruber, T. R., A Translation Approach to Portable Ontology Specifications, *Knowledge Acquisition*, 5(2):199-220, 1993.
- [Hodges 92] Hodges, J., Naive Mechanics - a Computational Model of Device Use and Function in Design Improvisation, *IEEE Expert*, 7(1):14-27, 1992.
- [Horváth 98] Horváth, I., Kuczogi, G.Y. and Vergeest, J. S. M., Development and Application of Design Concept, Ontologies for Contextual Conceptualization, In *Proc. of 1998 ASME Design Engineering Technical Conferences DETC*, DETC98/CIE-5701, ASME, New York, 1998.
- [IMC 99] Invention Machine Co., TechOptimizer, <http://www.invention-machine.com/products/techoptimizer.cfm>, 1999
- [Keukene 91] Keukene, A.M., Device Representation: the Significance of Functional Knowledge, *IEEE Expert*, 24:22-25, 1991
- [Kitamura 99] Kitamura, Y., and Mizoguchi, R., Meta-functions in Artifacts, *Papers of 13th International Workshop on Qualitative Reasoning (QR-99)*, 136-145, 1999
- [Kitamura 00] Kitamura, Y., Sano, T., Mizoguchi, R., Functional Understanding based on an Ontology of Functional Concepts, In *Proc. of The Sixth Pacific Rim International Conference on Artificial Intelligence (PRICAI 2000)*, pp.723-733, 2000
- [Kitamura 02] Kitamura, Y., Sano, T., Namba, K., and Mizoguchi, R., A Functional Concept Ontology and Its Application to Automatic Identification of Functional Structures, *Advanced Engineering Informatics (Artificial Intelligence in Engineering)*, to appear, 2002.
- [Kozaki 00] Kozaki, K., Kitamura, Y., Ikeda, M., and Mizoguchi, R., Development of an Environment for Building Ontologies which is based on a Fundamental Consideration of "Relationship" and "Role", In *Proc. of The Sixth Pacific Knowledge Acquisition Workshop (PKAW 2000)*, pp.205-221, 2000
- [Lind 94] Lind, M., Modeling Goals and Functions of Complex Industrial Plant, *Applied Artificial Intelligence*, 8:259-283, 1994.
- [Malmqvist 97] Malmqvist, J., Improved Function-means Trees by Inclusion of Design History Information, *Journal of Engineering Design*, 8(2):107-117, 1997.
- [Mizoguchi 00] 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
- [Pahl 88] Pahl, G., and Beitz, W., *Engineering Design - a Systematic Approach*, The Design Council, 1988.
- [Rosenberg 83] Rosenberg R. C., and Karnopp, D. C., *Introduction to Physical System Dynamics*. McGraw-Hill, 1983.
- [Salustri 98] Salustri, F. A., Ontological Commitments in Knowledge-based Design Software: A Progress Report, In *Proc. of The Third IFIP Working Group 5.2 Workshop on Knowledge Intensive CAD*, 31-51, 1998.
- [Sasajima 95] Sasajima, M.; Kitamura, Y.; Ikeda, M.; and Mizoguchi, R. FBRL: A Function and Behavior Representation Language. In *Proc. of IJCAI-95*, 1830-1836, 1995.
- [Sembugamoorthy 86] Sembugamoorth, V., and Chandrasekaran, B., Functional representation of devices and compilation of diagnostic problem-solving systems, In *Experience, memory and Reasoning*, 47-73, 1986.
- [Sushkov 95] 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
- [Takeda 90] Takeda, H., Veerkamp, P., Tomiyama, T. and Yoshikawa, H., Modeling Design Processes, *AI Magazine*, 11(4), 37-48, 1990
- [Tejima 81] Tejima, N. et al. eds., Selection of Functional Terms and Categorization, Report 49, Soc. of Japanese Value Engineering (In Japanese), 1981
- [Umeda 96] Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y., and Tomiyama, T., Supporting conceptual design based on the function-behavior-state modeler, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 10, 275-288, 1996.