Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2009
August 30 - September 2, 2009, San Diego, California, USA

DETC2009-87168

SOME ONTOLOGICAL DISTINCTIONS OF FUNCTION BASED ON THE ROLE CONCEPT

Yoshinobu Kitamura **Riichiro Mizoguchi**

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

ABSTRACT

Function is an important aspect of artifacts in engineering design. Although many definitions of function have been proposed in the extensive research mainly in engineering design and philosophy, the relationship among them remains unclear. Aiming at a contribution to this problem, this paper investigates some ontological issues based on the role concept in ontological engineering. We discuss some ontological distinctions of function such as essentiality and actuality and then propose some fundamental kinds of function such as essential function and capacity function. Based on them, we categorize some existing definitions in the literature and clarify the relationship among them. Then, a model of function in a product life-cycle is proposed. It represents the changes of existence of the individuals of each kind of function, which are caused by designing, manufacturing and use. That model enables us to give answers to some ontological questions such as when and where a function exists and what a function depends on. The consideration on these issues provides engineers with some differentiated viewpoints for capturing functions and thus contributes to consistent functional modeling from a specific viewpoint. The clarified relationships among the kinds of function including the existing definitions in the literature will contribute to interoperability among functional models based on the different kinds and/or definitions.

KEYWORDS

Functional representation, ontology, functional design, design knowledge modeling

1. INTRODUCTION

This paper discusses some ontological issues on functionality, which is one of the crucial concepts in design. Much research has been carried out on the notion of function in several research areas such as engineering design [1]-[6], functional representation in artificial intelligence [7]-[12], value engineering [13] and philosophy [14]-[16]. The problem here is that there are many definitions of function without clear relationship among them [4][6][16]. It is one of the deep reasons of the difficulty of description of *functional knowledge* and its interoperability. By functional knowledge, we here mean mainly a product model from the viewpoint of functionality such as functional decomposition [2], which consists of functions of components, sub-systems and the whole system, and the relations among those functions. Without clear understanding of function, engineers tend to describe functional knowledge based on an implicit perception of function in an ad hoc manner. As a consequence of inconsistency of the implicit perceptions, it is difficult to share and reuse the functional knowledge. Even if functional knowledge is clearly based on one of the definitions in the literature, the interoperability between functional knowledge based on different definitions of function is difficult due to the lack of the clear relationship among those definitions mentioned above. In order to realize consistent and interoperable description of functional knowledge, we need to clarify relationships among different definitions of function in the literature as well.

Ontological engineering in artificial intelligence research [17][18] can contribute to clear understanding of function and then to promotion of the functional knowledge sharing. Recently, the concept of “ontology” attracts much attention in engineering design as well (e.g., [19]). In many of them, however, ontologies have shallow semantics as a shared vocabulary or a data schema (called light-weight ontologies [18]). Rather, the deep ontological consideration as so-called a heavy-weight ontology is needed for clear understanding of function. Such ontological considerations on function (e.g., [10][20]) as well as deep considerations in philosophy have been done to date.

The authors also have investigated the notion of function based on deep ontological considerations and proposed a functional modeling framework based on heavy-weight ontologies [21]-[27]. The framework has been successfully deployed in the daily work in some manufacturing companies [24]. In the deployment, an ontological distinction between function (“what to achieve") and the way of function achievement ("how to achieve") [24] has enabled the engineers to describe functional knowledge consistently and clearly. Such consistent description has promoted the engineer's reflective thinking in (re)designing and problem-solving and the communication among the engineers. Many real benefits of those ontological distinctions have been proven in the deployment [24].

The purpose of this paper is to investigate some fundamental and ontological issues of function for clearer understanding of function and for further consistent modeling. Some of the ontological issues that we concentrate on in this paper are as follows;

- Where does a function exist? : This issue is related to the relationship between a function and an artifact. Does a function exist *inside* or *outside* an artifact? Does a function inhere in an artifact? Does an artifact *have* or *perform* a function?
- On what does the existence of a function depend? : This issue is related to the dependence of function on the intention of an agent (a designer, a user and so on). Does a function depend on the *intention of an agent* or on an *artifact*?
- When does a function come into exist and/or disappear? : This issue asks how the existence of function changes in the product life-cycle by activities such as designing, manufacturing and use.

The answers to these questions are unclear and different according to several definitions of function in the literature. For example, Umeda et al. define a function as "a description of *behavior* abstracted by human through recognition of the behavior in order to utilize it" [3]. In this definition, a function is a special kind of *process* (intuitively, temporal changes of physical attributes in the physical world) in the use phase of an artifact. Such a process caused by an artifact can be considered that it exists outside the artifact. On the other hand, Hubka and Eder define "function is a property of technical systems" [1]. In this definition, function is a *property* of an artifact as a result of a manufacturing process. Thus, according to these two definitions, the answers for the first ontological issue above are "a function exists *outside* an artifact, which *performs* the function" and "a function exists *inside* an artifact, which *has* the function as its property", respectively. This is not a special case as we will discuss in this paper in detail. The perception similar to the former can be found in many definitions (e.g., [2][5][7][8]) in engineering design and artificial intelligence. The perception similar to the latter can be found mainly in philosophy (e.g., [15][16]).

Thus, the investigation on those ontological issues of function in this paper contributes to clarification of the notion of function and the relationship among several definitions in the literature. It also provides some differentiated viewpoints for capturing functions and thus contributes to consistent modeling of function based on a specific viewpoint.

In this paper, we investigate those ontological issues of function based on the *role concept* in ontological engineering [28][29]. A *role* is intuitively defined as "an entity that is played by another entity in a context" [28]. Using the role concept, we define function-related concepts such as *device, behavior, function context, function* [24] and *capacity to perform a function* [30] and clarify the relationship among them. Then we make five ontological distinctions of the function-related concepts and then define some kinds of function based on these distinctions. Using these kinds of function, we categorize some definitions in the literature and try to clarify the relationship among them. We have proposed some upper-level types of function [27]. This paper investigates more fundamental kinds of function.

Then, Section 3 discusses changes of existence of function by designing, manufacturing, and use based on those concepts introduced in Section 2 and then proposes a macroscopic temporal model of function in a product life-cycle. For example, we discuss what kinds of function exist after the design process. On the basis of those considerations, the ontological issues introduced above will be discussed. We will give some answers to those questions using the kinds of function based on the ontological distinctions. Section 4 discusses related work followed by the concluding remarks.

2. ONTOLOGICAL DISTINCTIONS OF FUNCTION

2.1. Context-dependence (Behavior and Function)

One of the crucial characteristics of function is context-dependency. According to the context-dependency, we can distinguish function from *behavior*. As a fundamental viewpoint for capturing an artifact[1] in the physical world, we adopt the device ontology in our research [24]. From its device-oriented viewpoint, we regard an artifact as a system (composition) of *devices*. Devices are connected to other devices via ports and form the whole system. A device itself consists of other devices of smaller grain size, which form a whole-part hierarchy. A device operates on other things we call *operands*, and thus it changes their physical states. The operand is something that flows through the device via ports and is affected by the device. Here we define *behavior of a device* as the changes in the attribute values of the operands from the value at the input port of the device to the value at the output port. Such a behavior of a device is constant with respect to the device's situation and/or its context (i.e., context-independent).

On the other hand, function depends on a teleological context (we call *function context*), which represents how the device is used. A behavior can perform different functions according to contexts. For example, the behavior of a heat exchanger is always "to exchange (transfer) heat", which is described as temporal changes of the temperatures of the fluids. However, its function is "to give heat" when it is used as a heater with a turbine in a power plant, or "to remove heat" when it is used as a radiator with an engine in a car. Thus, we define *function* as "a role played by a (device-oriented) behavior in a teleological (functional) context"[2] [24]. A role concept cannot be defined without a context and the existence of a role concept depends on the existence of the context [28]. Strictly speaking, a function is a composite of a function-role concept and the behavior playing the function-role concept, which is a kind of a thing generally called a role-holder [28].

Based on the definition, we say that "a behavior plays a function role". If a device performs a behavior and the behavior

[1] In this paper, by an artifact, we mean a physical object that exists in the spatial-temporal space, consists of devices as a system, and is designed and manufactured intentionally as engineering. We treat not the artistic aspect but only the physical and functional aspects of an artifact.

[2] This definition is a precise version of our previous definition "a function is a result of teleological interpretation under a goal" [21]. This definition is of the base-function for an operand. We define meta-functions for other function [22] as well. In this paper, we concentrate on the base-function.

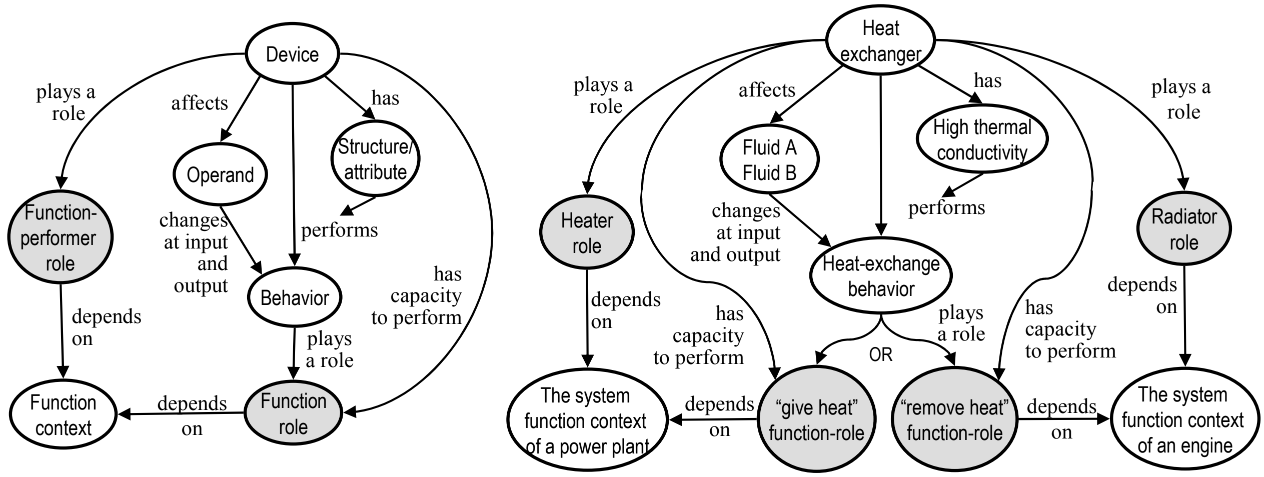

Figure 1. The relationship among device, behavior, function and function context. Left: Generic relationship, Right: An example of a heat exchanger.

plays a function role in a context, then the device plays a *function-performer role* in the context. For example, the heat-exchange behavior plays the heat-removing function role, and then the heat exchanger plays the function-performer role of removing heat as a radiator. Figure 1 shows such relationships among a device, an operand, behavior, function and function context. The left part shows generic relationships. The right part shows those in the example of the heat exchanger.

Our definition of function treats function's intention-relatedness in the literature as the context-dependence. The intention-relatedness is captured in the literature as "aims-means" [1], "means and ends" [8], F-B relationship [3] and in value engineering [10]. We categorize the function context and then identify kinds of function based on the categorizations in the following sections. In some philosophical considerations [15][31][32], function is regarded as a role. According to the categorization of definitions of function in [16], our definition is a kind of 'goal-contribution theory' in 'recent past backward-looking reductionist category'. In ontological engineering, general characteristics of roles have been investigated such as 'externally-founded', 'anti-rigid' and 'dynamic/multiple' [28][29]. We have shown that the characteristics of function satisfy those generic characteristics of roles [24]. On the other hand, in some definitions in the literature [2][5][7], function is not distinguished explicitly from behavior based on the intention-relatedness.

2.2. What Determines the Context: Functional Structure or User's Intention (Component Function and External Function)

The *function context* represents teleological goals to be achieved by the function. It is categorized into the *system function context* and the *external function context* according to what determines the context. The former is the context for a function of a component embedded in a system. The system function context is determined by a functional structure in which the system's function is achieved by a sequence of (relatively finer-grained) components' functions. We call this type of function under a system function context a *component function*. The functions of the heat exchanger discussed above are examples of this type. When the heat-transfer behavior of the heat exchanger is embedded in a power plant, under the system function context which consists of the whole function "to convert heat energy to electricity" and the functions of other components in the system configuration of the power plant, we can recognize "to give heat" as the component function of the heat exchanger. The heat exchanger performs a function-performer role of giving heat and is called "heater" as a role-holder as shown in the right part of Figure 1.

Such a functional structure is called "degree of complexity" [1], function decomposition [2], or function achievement relation [24]. The causal-role function analysis [15] also captures the relationship between the whole function and the component functions as contribution. The constituent function [33] also is similar to the component function.

On the other hand, the latter *external function context* depends directly on the user's intention. For a device as the whole product, its external function context is determined by how it is used by a user externally. A function in an external function context (which we call the *external function*) is directly intended by a user. Some external functions are also intended by the designer, as discussed in the next section. For example, a screwdriver can be used for performing a screwing (exerting rotational force on a screw) function or a hitting (exerting linear force on something) function using the tail part of the screwdriver. The performed function of a screwdriver at a specific use depends on the user's intention. The external function context determines the whole function of an artifact as the whole system. Based on the whole external function as the system function context, the component functions of the components of the system are determined.

This distinction between these two kinds of functions is related to the issue of what a function depends on. The external function depends on a user's intention, while a component function depends on a functional structure of the system. For the component function, although the functional structure of the system depends on the user's intention, the dependence on the user's intention is indirect. In this sense, we could say that the component function depends on the system.

Note that a function depends not only on a context but also on a behavior and a device which performs the function as discussed in Sections 2.4 and 3.4. By the word "external" in the external function, we mean that its *context* is *external* to the device as the whole system. The external function depends both on the external function context and on the capacity of the device as discussed in Section 2.4.

2.3. Essentiality (Essential Function and Accidental Function)

The function context is categorized also according to its essentiality. In the example of the screwdriver discussed above, the screwing function is intended by its designer. The screwdriver is designed and manufactured for performing this function. Such a function context that is intended by a designer in the design phase is called a *designed function context* and the function under this context is called an *essential function*. On the other hand, the hitting function of the screwdriver is intended *not* by the designer *but only* by the user. Such a function context is called an *accidental use function context* and the function under this context is called an *accidental function*. This distinction of function is relative with respect to a device. For example, the screwing function could be performed by a key as its accidental function. The hitting function is an essential function of a hammer.

This distinction can be considered not only among the external functions discussed thus far but also among the component functions. In other words, those distinctions are orthogonal to each other. A component is designed and manufactured for a specific function, which is its essential function. When a component is integrated into a system, a designer of the system can use a component for achieving its essential function or a different function as an accidental function. For example, slurry containing diamond powder is manufactured for improving cutting efficiency. However, in a cutting machine, the slurry is also used for cooling the cutting blade. In this case, from the viewpoint of the intention of the designer of the slurry as a component, the slurry performs a cooling function as an accidental function.

Such accidental use is distinguished from a (proper) function in some philosophical writings (e.g., [14][16]). Such accidental use is called "function-as." We regard an accidental effect as a function if it is recognized in a specific function context. If not, it is a behavior. The first justification for our use of the term function here is the effect-oriented definition from the engineering viewpoint. As actual effects, an essential function and an accidental function can have the same intended effect for users. The second justification is that we discuss mainly functions actually performed by a device, which is different from the function inhering in device in philosophy as discussed in the next section.

2.4. Performance and Capacity (Actual Function and Capacity Function)

A function discussed thus far is a role played by behavior performed by the device. It actually exists in the physical world when the device is used. In this sense, we can call this function *actual function*. On the other hand, a device that performs an actual function is needed to have *capacity to perform the function*, which is based on a set of appropriate properties (e.g., physical attributes, structure, geometry and material) for realization of the behavior that plays the actual function as a role. The capacity to perform a function is possible, potential and inherent in a device.

This distinction between the actual function and the capacity to perform the function is orthogonal to other distinctions discussed in Sections 2.2 and 2.3. For an essential actual function of a device, the physical properties necessary for the capacity to perform that function are intentionally designed by a designer and are manufactured. For an accidental actual function, a device's capacity to perform that function is not intended but potentially inheres in the device. When an appropriate context and appropriate inputs for an actual function are given to the device, the device's hidden capacity to perform that function is induced and is exhibited. Then that function is actually performed. The possible actual function is restricted by the device's capacity to perform the functions. In the example of the screwdriver, the screwdriver has potentially the capacities to perform (at least) the screwing actual function and the hitting actual function based on its shape and hardness. The latter capacity is hidden in usual use and is induced when it is used as a hummer in an accidental function context.

In philosophy, a function is typically a special feature of artifacts [16]. In causal-role function analysis [15] and ICE theory [32], a technical function is regarded as a special kind of capacity (or disposition) to be ascribed to an artifact. In engineering design, as mentioned in Introduction, Hubka and Eder [1] define functions as follows: "The function is a property of the technical system, and describes its ability to fulfill a purpose, namely to convert an input measure into a required output measure under precisely given condition." In their definition, a purpose represents intended effects as output effects, while a function is the *ability* for an internal task of the technical systems.

We call this type of function *capacity function*, which represents capacity to perform an actual function. We call the capacity function to perform an essential actual function of a device as an *essential capacity function*. For an accidental actual function, we call an *accidental capacity function*. Thus, the purpose and the function in Hubka and Eder's definition [1] roughly correspond to the actual function and the capacity function in our terminology, respectively. Boorse [31] makes a similar distinction in terms of a "weak function statement" and a "strong function statement", which roughly correspond to the actual function and the capacity function, respectively. The distinction between *function* and *functioning* in [34] is also similar. In [34], the function of a biological organ is defined as "a disposition to act in a certain way to contribute to the realization of [a ...] larger function" [34]. The functioning can be realized in temporal physical space as a SPAN entity. These roughly correspond to the capacity function and the actual function, respectively. Their definition of function is, however, different from our capacity function in the senses of disposition and reliability. Moreover, our actual function is a role played by a behavior and includes that in an external function context.

There are deep ontological differences between these two types of functions. A capacity function is a thing (property) that a device *has*, *exists inside* the device, and is *dependent* on the device. On the other hand, an actual function is a thing that a device *performs (cannot have)*, exists *outside* the device and is *less dependent* on the device. Thus, when we say "a device has a function", the function represents a capacity function. This capacity function represents a possibility to perform an actual function and thus is potential and hidden in device. It is induced by a user.

We define a function as an actual function as discussed above. Its justification is from the realization-independence of function. A behavior and physical features of an artifact are specific ways of realizing the required function. A function can be realized by different behaviors (and artifacts) in different ways. Thus a function should be independent of its realization. This engineering requirement justifies our definition of functions as an actual function detached from artifacts.

2.5. Performance and Specification (Actual function and Specification of Function)

The actual function introduced in the previous section actually exists in the temporal physical space in the time interval of the actual use by a user. On the other hand, we can consider a *specification of function (SF)*, which specifies an actual function to be made exist in the physical world. Generally, a specification is a proposition that specifies a thing in the real world [18]. In the usual terminology in engineering design, the term "a specification of function" seems to refer to a specification of a device from the viewpoint of functionality. In addition to this meaning, by the term, we here mean a set of constraints on the actual function to be performed by the device. In the same manner, we consider a *specification of behavior*. When we use the term "*a specification of device"*, by the term, we here mean a specification of the physical aspect of a device. Its typical example is produced as the result of the design process and specifies the physical properties of the device to be manufactured. The specification of function that is a result of a conceptual design is called *specification function*. We discuss this topic in detail in Section 3.1. See examples in Sections 2.6 and 3.1. Moreover, the capacity function discussed in Section 2.4 is represented as a specification of function (SF) as discussed in Section 3.2.

By specification, we here mean a *proposition* that gives constraints on an actual function. Thus, a specification of function is not needed to be *described* on a physical thing. Thus, it is not only a *formal* description such as formal design requirements but also a *hidden* proposition as a capacity function as discussed in Section 3.2 or a *mental* proposition as a *required function at the use phase* discussed in Section 3.3.

2.6. Class and Instance of Function (Functional Class and Instance)

We can identify generic types (classes) of function in *is-a* (a-kind-of) hierarchies. In the literature, the hierarchy for the "degree of abstraction" of functions [1], generally-valid function [2], a taxonomy of generic functions called Functional Basis[3] [5] have been proposed. We also have proposed an ontology of generic functional classes called a functional concept ontology [22] and have established the mappings between it and Functional Basis [26]. In the ontology, a functional class consists of constraints on behavior playing the function-role and propositions for teleological interpretation called *functional toppings* [21][22]. For example, a definition of the "to give heat (to medium A)" function includes the behavioral constraints; the existence of two mediums A and B for heat and the existence of a thermal energy flow from B to A. The definition includes also focus on the transferred thermal energy and focus on the heat receiver (the medium A) for teleological interpretation.

Such definitions of the function classes conform to the generic form of definition of the role-concepts. A role-concept is defined mainly with constraints on its *player*, specification of a *context* in which the role is played, and own attributes [28]. In the case of a function-role, the constraints on the player are those on some features of behavior and refer to a limited part of behavior. It enables us to treat the possibility that some behaviors can play the same function as a role. The specification of the context consists of constraints on the functional toppings. The functional toppings represent focuses on local elements in a device (e.g., energy and medium) as a *reflection* of a functional context, which is not internal in either cases of the system function context and the external function context. Thus, each functional class can be defined internally, while the recognition of a concrete actual function as a role played by a behavior is done according to a non-internal function context. After recognition of function, a functional structure of a system consists of instances of the internally-defined generic functional classes and relations among them.

According to the categorization of function discussed thus far, individuals of function also are categorized into instances of the classes of actual functions, those of *specifications of function* or those of capacity functions. We discuss the existence of those instances of functions in Section 3. The following paragraphs give the overview of the relationships using simple examples.

A functional class discussed above defines constraints on individuals of actual functions. When a device performs a function in the use phase, an individual of an actual function is instantiated from a functional class as its instance (called *instance-of* relation) as discussed in Section 3.4. For example, when a heater performs a giving-heat function in a specific function context, an individual of the actual function is instantiated from the "to give heat" class. The instance of a functional class satisfies the constraints defined in the class. The instance of the "giving-heat" function consists of the focus on the transferred thermal energy, the focus on the heat receiver and own attribute values such as a value of the thermal conductance (we call *attribute values* of function). In the same manner, the behavior playing the giving-heat function-role consists of at least two heat-carrying-mediums and a thermal energy flow between them. The behavior includes other changes which an agent (a designer or a user) does not intend and/or care about.

A functional class specifies its instances. In this sense, a functional class is similar to a *specification* discussed in the previous section. In fact, the *instance-of* relation is similar to the relationship between an instance of a *specification of function* and an instance of an actual function (called *realization-of* relation [18]). An instance of a *specification of function* can be regarded as a class-equivalent, though a class is not an entity but an instance of a specification of function is an entity. The classes of *specifications of function* can be defined in the same manner of the classes of actual functions. For example, there is a "to give heat" class of *specifications of function*, which refers to the "to give heat" class of the actual function discussed above. As we discuss in Section 3.1, a design result includes a specification of function to be realized. For example, a result of designing of a heater includes an instance of the "to give heat" class of *specifications of function (SF)*. This instance specifies

[3] The functional terms defined in Functional Basis [5] are categorized into the actual function in our terminology, though the distinction between function and behavior is not so clear and some functional terms have impure meanings [26].

optionally the attribute values of the function such as the thermal conductance mentioned above (called *attribute values of specifications of function*). This instance of SF gives constraints on an instance of the "to give heat" class of the actual function.

2.7. Class of Device and Function

As discussed in Section 2.3, a device is designed and manufactured for performing its essential actual function. When the device loses its capacity to perform the essential function, the device is not regarded as the same device (we will revisit this topic in detail in Section 3.5). Thus, for a device, the capacity to perform its essential actual function is an *essential property* of the device. Generally, the notion of the *essential property* of a thing is used for providing its *identity* in ontological engineering [18]. For example, the capacity to perform the screwing actual function is the essential property of a screwdriver. A screwdriver is designed and manufactured for having this property. When a screwdriver loses this capacity, it is not regarded as a screwdriver anymore. Note that the actual performance of an essential function (i.e., as an actual function) of a device is *not* an essential property of the device. Even if a screwdriver is used for hitting a nail as an accidental function, we can say that it is still a screwdriver. The essential property of a device is *the capacity to perform* its essential actual function.

A definition of a class (type) of a device includes the capacity to perform its essential actual function as a main part. The capacity of its essential function refers to a functional class[4]. Thus, the capacity to perform a function essentially needs to refer to a class of an actual function, which is defined independently of a device. Additionally, a definition of a class of a device includes the capacities to perform its accidental functions which are known as the functions that the device can possibly perform. Such possible accidental functions induced by accidental uses, in principle, cannot be enumerated completely before used in reality.

An instance of a device class has the capacities to perform the functions which are specified in the definition of the class. For example, the screwdriver class refers to both the screwing functional class and the hitting functional class as the capacities to perform functions. An instance of a screwdriver has at least both capacities unless it is broken and malfunctioning.

3. CHANGES OF EXISTENCE OF FUNCTION

In this section, we discuss temporal changes of existence of function in a product life-cycle, which are made by activities such as designing, manufacturing and use. Its aim is to investigate the ontological issues such as where and when a function comes into exist and disappears. Figure 2 shows a part of the model of a chair as an example. The upper part shows the *is-a* (*sucClassOf*) hierarchies of the classes of specification, device, function, and so on. The lower part shows a macroscopic temporal model of the individuals (instances) of those classes where the time flows from left to right. Each of Figure 2(a), (b), (c) ... shown as a gray box represents a *phase* in a time interval or at a time point. For example, Figure 2(a) represents the phase of the starting time point of the designing activity. The transition from a phase to next phase is made by an activity (depicted as a dark-gray arrow in Figure 2). For example, the design activity causes the transition from the phase (a) to the design result phase shown in Figure 2(b). In the following sections, we discuss what kinds of function exist in what phase and the relations among them.

[4] In the terminology of the OWL language (http://www.w3.org/2004/OWL/), a device class has the property named *capacity function* and the range of the property is restricted by a functional class. More precisely speaking, the restriction is described using a class of *a specification of function* corresponding to the functional class of the actual function.

3.1. Designing Activity

For a designing process, a *function context* of anticipated use and an *actual function* to be realized are usually specified as given inputs for the designing problem, though these specifications of the functional context and the function are incomplete in many cases. In Figure 2(a), these design requirements for the designing activity are represented as an instance of a *specification* class of a function context (SFC_1: "a person sits on a device" as an example for a chair) and an instance of a class of a *specification of function* (SF_1: "to support a person being in a sitting position"). The SF_1 is a *specification* (in the sense discussed in Section 2.5) of such a function that should be performed by a device to be designed. It is regarded also as a function-level specification of the device. Thus, this *specification of function* can be regarded as so-called a *required function*.

After the designing process, there are the specifications of functions, behaviors and devices, which are determined in the designing process for satisfying the design requirements. In Figure 2(b), there are a specification of function (SF_2), a specification of a device (SD_1) and a specification of a behavior (SB_1) as the design result. The specification of function as a result of the designing (SF_2) usually satisfies the required function (SF_1). If not, the situation is failure of the designing task. The SF_2 is called a *specification function*, which restricts the actual function that exists in the real world as introduced in Section 2.5. Although we could regard the specification of a chair SD_1 as a sub-class of the chair class, we do no adopt this view because this view implies that each designing process generates a new class of a device.

3.2. Manufacturing Activity

The manufacturing activity is to make a device exist in the real world according to a specification of the device. In the example of the chair in Figure 2, the manufacturing activity makes an instance of the chair class (C_1) that satisfies the conditions of the specification of a chair (SD_1) as shown in Figure 2(c). The chair instance C_1 has an *instance-of* relation with the class of a chair and has a *realization-of* relation with the specification instance SD_1. As discussed in Section 2.6, these relations have the same role, that is, to restrict the chair instance C_1.

In Figure 2(c), the chair instance C_1 has an instance of specification of function (SF_3) that is to support a person as a capacity to perform a function. In other words defined in Section 2.4, the chair instance C_1 *has* a *capacity function* SF_3; to support a person. The existence of SF_3 is based on the *specification function* SF_2 as the result of the designing. Thus, SF_3 is a capacity function of an essential actual function. The propositions of SF_3 are determined by the manufacturing process. Ideally, SF_3 is equal to SF_2. If SF_3 does not satisfy SF_2, it represents the case of failure of the manufacturing activity. In addition, the chair instance C_1 can have specifications of functions other than SF_3 as its capacity functions of accidental functions. These specifications are based on the physical properties of C_1 as the derived (and unintended) results of the designing and the manufacturing for realizing the required function SF_1. In Figure

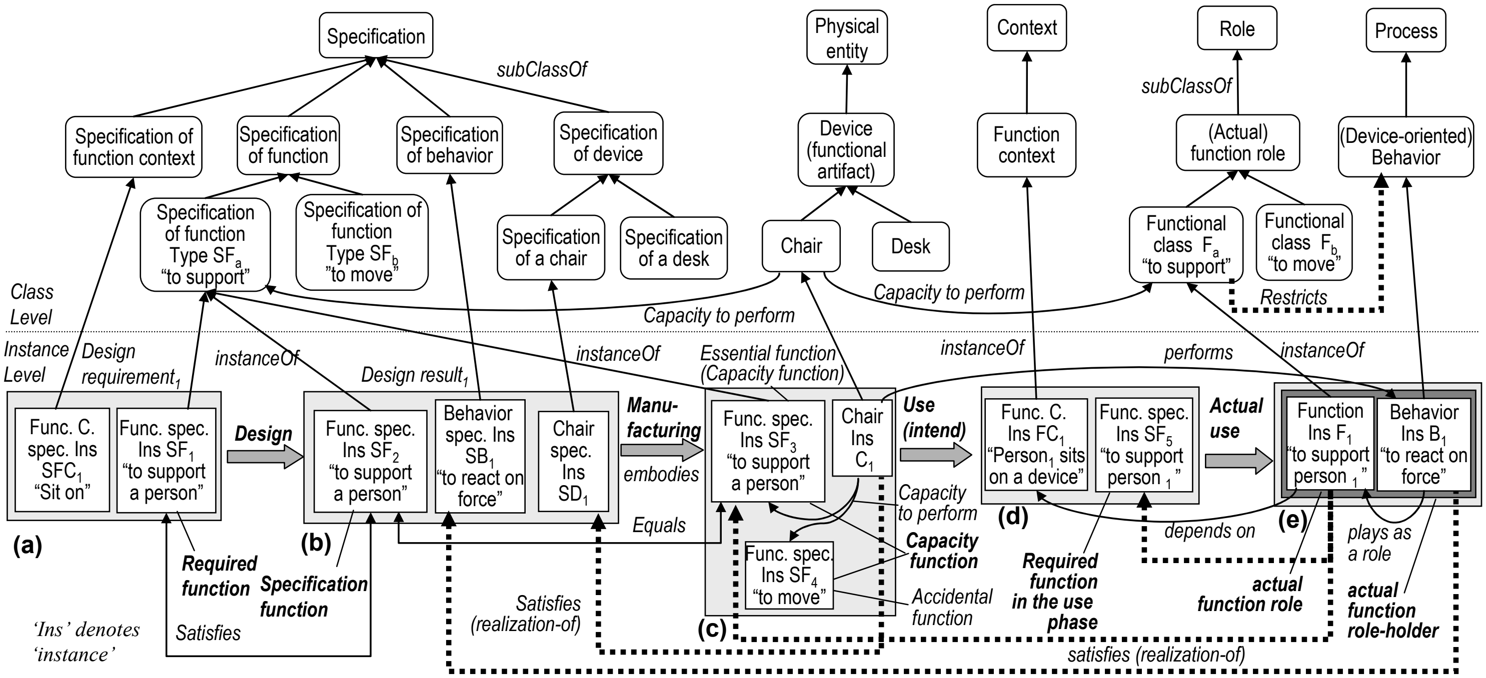

Figure 2. An example of changes of existence of function in a product life-cycle. (a chair)

2(c), the chair instance C_1 has another capacity function SF_4 "to move an entity" as a possible accidental function. As discussed in Section 2.5, those *specifications of function* (SF) of the capacity functions (especially that of an accidental function) are *hidden* propositions.

3.3. Use Activity: Intending to Use

Figure 2(d) shows a situation when a user intends to realize a specific function in a specific *external function context*. There is an instance of the external function context (FC_1: "a specific person ($person_1$) sits on a device") and an instance of the specification of function (SF_5) representing the user's intention ("to support the $person_1$"). The SF_5 is similar to the required function SF_1 before designing and thus can be called a *required function in the use phase*. The required function SF_1 before designing, however, specifies only looser restrictions on instances of actual functions in a generic context, while the required function in the use phase SF_5 specifies tight restrictions on the instance of the actual function in the function context that is fully determined according to the specific context of use. However, the function context FC_1 and the specification of function SF_5 specify neither the instance of the device to perform the function nor its class. Thus, the user does not determine what device he/she wants to use in this phase yet.

3.4. Use Activity: Actual Use

As shown in Figure 2(e), when a device C_1 is used in the function context FC_1 intended by a user and the device C_1 has the capacity to perform a function satisfying the intended specification of function SF_5, the device performs a behavior instance B_1 and the behavior B_1 plays the actual function instance F_1 as a role in the intended function context FC_1.

This is a result of a user's activity to select such a device C_1 that has a capacity to perform the actual function F_1 which can satisfy the required specification of function SF_5. Thus, F_1 satisfies SF_5 (F_1 *is-realization-of* SF_5). The specification SF_5 has a role of specifying the actual function F_1. Thus, those instances have different identities.

In Figure 2(e), because the actual function instance F_1 satisfies the specification function SF_3 (the essential capacity function of the device C_1) associated with the *specification function* SF_2 as the result of the designing process, we can say that an essential function of the device C_1 is actually performed and F_1 is an essential actual function. As a result, F_1 is a *realization-of* SF_2, SF_3 and SF_5.

On the other hand, an actual use of an accidental capacity function of the device C_1 can be represented as a situation that the actual function instance F_1 satisfies the capacity function specification SF_4 ("to move an entity") that is other than SF_3 based on the designed *specification function* SF_2. The capacity that implicitly inheres in the device is induced by the use and then an accidental actual function is performed.

Thus, the performance of an actual function (F_1) can be regarded as *successful* if F_1 satisfies both the required function at the use phase (SF_5) and the capacity function of a device (SF_3 or SF_4). If not, it represents the case of *failure* of the performance of an actual function as discussed in the next section.

In Figure 2(e), F_1 is an instance of the supporting function-role class F_a. The existence of this role-instance primarily depends on the intended function context FC_1. Then, the actual function-role instance F_1 and the behavior B_1 as its player compose a role-holder as a complete actual function. From the temporal point of view, these occurrences are at the same time-point. From the viewpoint of causality, we however can consider that the causal order of these occurrences is the existence of the function-role and then the role-playing by the behavior. In this sense, the existence of the actual function as a role-holder also depends primarily on the function context FC_1 as the user's intention, while it depends on the device's capacity to perform the function (for the existence of the behavior) as well.

Thus, we can consider that the identity of the actual function F_1 depends primarily on the function context FC_1. For example, let us consider a case where a user initially sits on the left part of a sofa and then moves to the right part of the same sofa for better comfortableness. If we view this use-sequence of the sofa being invoked by the same requirement/desire, we can consider that there is only an individual of the actual function "to support the user" and the same function-role is played by different behaviors achieved by the same sofa. In other words, there is only an instance of a function-role but are two instances of the role-holder of the function-role. The individual of the role-holder is instantiated for each time of playing the role.

There are two cases of the accidental use. The first case is that the accidental usage of a device has been known and has been anticipated. The second case is that the specific usage of a device is newly found. The former case is represented as the association between the device and the function that is defined at the class level as discussed in Section 2.7. In the latter case, we can say that the association is *emerged*. Note that the functional class itself is defined previously and then is not emerged.

3.5. Use Activity: Non-Use and Malfunction

In such a situation that no person sits on the chair C_1, we can describe this situation as a model in which the capacity functions SF_3 and SF_4 exist as properties of the chair C_1 and there is no actual function.

Let us consider such a situation where a leg of a chair C_1 is broken and thus a person cannot sit on it when a user intends to do so. If a user is aware of the insufficiency of a device's capacity to perform a required function, that device is not selected usually for the function by the user. If the user is not aware of it and the user sits on the chair once, an instance of actual function exists for a moment then immediately disappears according to the failure of performance of the function.

The problem here is that, at this time point, a chair C_1 loses its essential property, that is, the capacity to perform its essential function; to support a person. In this situation, whether the entity C_1 can be recognized as a chair or not depends on the possibility of repair and the length of the time interval to be considered. If the entity C_1 can be repaired for resuming having the capacity and we consider a longer time interval including the repairing period, we can consider that the C_1 remains a chair. Such recognition is natural, for example, for a car without gas. On the other hand, if the chair cannot be repaired and/or we consider the time point only, the entity C_1 could not be regarded as a chair.

3.6. Summary of Existence of Function

The followings show a short summary of the existence of instances of function discussed thus far.

- Start of designing (shown in Figure 2(a)): An instance of a *required function* (a specification of function) exists as given inputs for designing.
- Result of designing (Figure 2(b)): An instance of a *specification function* (a specification of function) exists as a design result.
- Result of manufacturing (Figure 2(c)): Instances of *capacity functions* (specifications of function) exist inside a device and inhere in the device as its hidden properties.
- A specific use is intended (Figure 2(d)): An instance of a *required function at the use phase* (a specification of function) exists in an instance of a *function context* representing user's intention.
- During actual use (Figure 2(e)): An instance of an *actual function role* exists and is played by a behavior performed by a device. The actual function-role and the behavior compose *an actual function role-holder* as a complete existence of the actual function. Those exist outside the device.

Thus, the answers to the ontological questions introduced in Section 1 such as "when and where a function exists" and "on what the existence of a function depends" are different according to the kinds of function to be considered. For an *actual function* that is our primary definition of function, an instance of that function (1) comes into exist when a device is actually used for the function, (2) exists dependently on a function context for the function as the user's intention, (3) exists dependently on the device's capacity to perform the function, (4) exists dependently on the existence of a behavior as a player and that of the performing device, (5) exists outside the device, and (6) exists completely as a role-holder.

On the other hand, for an *essential capacity function*, which is similar to some definitions in philosophy, an instance of that function (1) comes into exist when a manufacturing process of a device finishes, (2) depends on the designer's intention for the device, (3) exists dependently on the device, (4) exists inside the device as a property, and (5) inheres essentially in the device. The main contribution of this paper is that the ontological distinctions introduced enable us to describe such propositions clearly based on different perceptions of function.

At the instance level of the proposed model, the instance of an essential capacity function is an essential property of a (function-oriented) device and exists during the whole period of the device's existence (except the period of malfunctioning). Thus, its existence is *stable*. On the other hand, the instance of an actual function exists dependently on a specific function context such as a user's intention and its existence is supported by a realized behavior of a device and the device's capacity to perform the function. Thus, its existence is *dynamic*. This dynamism fits in well with the properties of the notion of function discussed in Sections 2.1, 2.2 and 2.4; those are function's dependence on contexts and its independence of behaviors and devices as its realization (i.e., a function can be realized by different behaviors and/or different devices). As discussed in Sections 2.3 and 2.4, the justifications of the actual function as our primary definition of function are this realization-independence and the importance of *actual effects* and values for users from the engineering viewpoint. In engineering, what function a user can obtain is important. It is represented as an instance of a specific type (class) of the actual function (e.g., "to support"). As discussed below, such a class is defined as a sub-type not of the capacity function but of the actual function. Moreover, in engineering process, the instances of the specifications of function play a crucial role as the design requirement and the design result as shown in Figure 2. A specification of function also needs to refer to a specific type of the actual function.

At the class level, the definition of a type (class) of function (e.g., "supporting function" class in Figure 2) refers to a behavior as a player and a function context as a context. Thus, it is defined as a sub-type of the actual function as a role. The sub-types of *specifications of function* (in the left part of Figure 2) refer to those classes of the actual function. The definition of a (function-oriented) device also refers primarily to those classes for the specification of its essential capacity function as its essential property. Thus, at the class level, the sub-classes of the actual function play a main role in the class definitions.

4. RELATED WORK

We discussed definitions of function in the literature related to the ontological distinctions in Section 2. Here we discuss remaining related work. In the contrast with our device-oriented definition of function, Chandrasekaran and Josephson discuss an environment function, as an effect on the environment (the surrounding world of the device) [10]. Some researchers distinguish purpose from function (e.g., [35][36]), whereby the purpose represents a human-intended goal. Although the effect of the actual function in our definition is local within the device performing the function, the environment function and the purpose capture the effects on the environment or human perception, which are consequences in causal or enabling relations of the local effect. Borgo et al. [20] investigate a formal account of the definitions of function and behavior proposed in [10]. They logically define those concepts as sub-types of the upper-level categories of the DOLCE top-level ontology.

The Situated FBS framework [11] proposes a design process model with three kinds of environments called the external, the interpreted and the expected worlds. We do not discuss internal processes of the designing activity. Our *specifications of function* could be categorized into those in these worlds.

Vermaas and Houkes emphasize that functions are features that are ascribed by agents to artifacts relative to use plans and human beliefs [32]. Our external function context depends on such a "use plan," though the functions in the ICE theory are based on an agent's beliefs on capacity and contribution, which are implicit in our definition. Garbacz points out that a function is a state of affairs, which represents a connection between objects and processes [37]. Our definition tries to define the connection in terms of the context from the device-oriented point of view.

Our definition of function is based on the notation of role concept [28]. Much research has been conducted on representation of roles; e.g., [29][39]. Our analysis of function as role discussed in Section 2 follows the role's characteristics discussed in these papers. Masolo et al. [29][39] also introduce a context to define a role on the basis of D&S theory [38]. In this paper, we investigated types of contexts for functions and kinds of function based on distinctions of the function contexts.

The exclusive features of our theory of roles include a clear distinction between instantiating a role and playing the actions of a role and hence between the instance of a role and the existence of the role playing thing (role-holder in our terminology) [28]. The 'qua-individual' in [39] is similar to an instance of a role-holder. We conceptualize both a class of a role concept and a class-equivalent of a role-holder.

Vieu et al. investigate modeling strategies of 'artifact' and 'roles' [40]. In their definition, an artifact is the result of an agent intentionally selecting a physical object and attributing to it some capacities, which characterize function as determined by the agent [40]. Their theory can represent the relationship between an artifact and what constitutes the artifact, and intentional use of natural things (e.g., a pebble used as a paperweight [40]). In this paper, we do not discuss the definition of artifacts and concentrate on the devices that are designed and manufactured intentionally (see the footnote 1). Although the intentional use of natural things seems to be able to be treated as an accidental use in our theory, further investigation on that use remains as future work.

In this paper, we concentrate on a typical functional requirement-driven design process (method). Our intention is to propose *not* a new design methodology *but* a model (representation) of function along a product-life cycle including designing. So, we do not discuss the designing process itself (i.e., how to get the design results from the requirements) in Section 3.1. Nevertheless, it clarifies the relationship among design requirements, design results, manufacturing based on the design results, and using it. To cover other design methodologies, e.g., affordance-based design [41] also remains as future work.

5. CONCLUDING REMARKS

In this paper, aiming at clearer understanding of the notion of function, we proposed some fundamental kinds of functions based on ontological distinctions using the role concept as a key concept. Using the kinds of function such as the actual function and the capacity function, some definitions of function proposed in the literature both in engineering design and philosophy are categorized and the relationships among them are clarified. In many considerations in the literature, such kinds of function have been called just "function" without clear distinctions. Such categorization of the existing definitions and the relationship among them contribute to reconcile the different perceptions behind them and then to clear understanding of the notion of function.

The ontological distinctions led us to propose a model of changes of existence of function in a product life-cycle. The macroscopic temporal model of function enabled us to answer some ontological questions introduced in Section 1 using the kinds of function.

From the viewpoint of design knowledge management, the ontological distinctions and the kinds of functions contribute to consistent modeling of function. The model-authors can describe a functional model consistently based on a differentiated kind of function. The relationships among definitions of function contribute to interoperability of functional models based on different definitions. We aim not at standardization of definition of function but at interoperability of different definitions of function. For this, we had proposed some upper-level types of function as a reference ontology of function [27]. The kinds of function in this paper are more fundamental and will be integrated into the reference ontology.

The ontological consideration on function discussed in this paper is still far from completion. Especially, further investigation on specifications of function and its relationship with actual functions is needed. The temporal model of function discussed in Section 3 treats actual functions in an external function context. An investigation on the model of component functions in system function contexts remains as future work. Furthermore, logical modeling of function like that in [20] is required for clearer semantics of the notion of function.

ACKNOWLEDGMENTS

The authors are most grateful to Professor Barry Smith of University at Buffalo for extensive discussions. The authors thank the anonymous reviewers for their valuable comments.

REFERENCES

[1] Hubka, V. and Eder, W.E., 1988, *Theory of Technical Systems*, Springer-Verlag.

[2] Pahl, G. and Beitz, W., 1996, *Engineering Design - a Systematic Approach*, Springer-Verlag.

[3] Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y., and Tomiyama, T., 1996, "Supporting conceptual design based on the function-behavior-state modeler", *AI EDAM,* **10**, pp. 275-288.

[4] Hubka, V. and Eder, W.E., 2001, "Functions Revisited", In *Proc. of ICED 01*.

[5] Hirtz, J., Stone, R.B., McAdams, D.A., Szykman, S., and Wood, K.L., 2002, "A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts", *Research in Engineering Design*, **13**, pp. 65-82.

[6] Stone, R. B. and Chakrabarti, A. (eds.), 2005, "Special Issues: Engineering applications of representations of function", *AI EDAM*, **19** (2 and 3).

[7] Chandrasekaran, B., Goel, A. K. and Iwasaki, Y, 1993, "Functional representation as design rationale", *Computer*, **26** (1), pp. 48-56.

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

[9] Chittaro, L. and Kumar, A. N., 1998, "Reasoning about Function and its Applications to Engineering", *Artificial Intelligence in Engineering*, **12**, pp. 331-336.

[10] Chandrasekaran, B. and Josephson, J.R., 2000, "Function in Device Representation", *Engineering with Computers*, **16** (3/4), pp. 162-177.

[11] Gero, J.S. and Kannengiesser, U., 2002, "The Situated Function-Behaviour-Structure Framework", In *Proc. of Artificial Intelligence in Design '02*, pp. 89-104.

[12] Goel, A. K., Rugaber, S., and Vattam, S., 2009, "Structure, behavior, and function of complex systems: the structure, behavior, and function modeling language", *AIEDAM*, **23** (1), pp. 23-35.

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

[14] Wright, L., 1973, "Functions", *Philosophical Review*, **82**, 39-168.

[15] Cummins, R., 1975, "Functional Analysis", *The Journal of Philosophy*, **72** (20), pp. 741-765.

[16] Perlman, M., 2004, "The Modern Philosophical Resurrection of Teleology", *The Monist*, **87** (1), pp. 3-51.

[17] S. Staab and R. Studer (eds.), 2003, *Handbook on Ontologies*, Springer.

[18] Mizoguchi, R., 2003, 2004, "Tutorial on ontological engineering - Part 1-3", *New Generation Computing*, **21**(4) and **22**(1-2).

[19] McMahon, C., van Leeuwen, J. (eds.), 2009, "Special issue on Developing and Using Engineering Ontologies", *AIEDAM*, **23**(1).

[20] Borgo, S., Carrara, M., Garbacz, P. and Vermaas, P. E., 2009, "A Formal Ontological Perspective on the Behaviors and Functions of Technical Artifacts". *AIEDAM*, **23**(1), pp. 3-21.

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

[22] Kitamura, Y., Sano, T., Namba, K., and Mizoguchi, R., 2002, "A Functional Concept Ontology and Its Application to Automatic Identification of Functional Structures". *Advanced Engineering Informatics,* **16**(2), pp. 145-163.

[23] Kitamura, Y. and Mizoguchi, R., 2003, "Organizing Knowledge about Functional Decomposition", In *Proc. of the 14th International Conference on Engineering Design (ICED 03)*.

[24] Kitamura, Y., Koji Y., and Mizoguchi, R., 2006, "An ontological model of device function: industrial deployment and lessons learned", *Applied Ontology*, **1**(3-4), pp. 237-262.

[25] Kitamura, Y., Washio, N., Koji, Y., Sasajima, M., Takafuji, S., and Mizoguchi, R., 2006, "An Ontology-Based Annotation Framework for Representing the Functionality of Engineering Devices", In *Proc. of ASME IDETC/CIE 2006*, DETC2006-99131.

[26] Kitamura, Y., Segawa, S., Sasajima, M., Tarumi S., and Mizoguchi, R., 2008, "Deep Semantic Mapping between Functional Taxonomies for Interoperable Semantic Search", In *Proc. of the 3rd Asian Semantic Web Conference (ASWC 2008)*, LNCS 5367, pp. 137-151.

[27] Kitamura, Y., Takafuji, S. and Mizoguchi, R., 2007, "Towards A Reference Ontology for Functional Knowledge Interoperability", In *Proc. of the ASME IDETC/CIE 2007*, DETC2007-35373.

[28] Mizoguchi, R., Sunagawa, E., Kozaki, K. and Kitamura, Y., 2007, "The Model of Roles within an Ontology Development Tool: Hozo", *Applied Ontology*, **2**(2), pp. 159-179.

[29] Masolo, C., Vieu, L., Bottazzi, E., Catenacci, C., Ferrario, R., Gengami, A. and Guarino, N., 2004, "Social Roles and their Descriptions", In *Proc. of the 9th Int'l Conf. on the Principles of Knowledge Representation and Reasoning (KR2004)*, pp. 267–277.

[30] Kitamura, Y. and Mizoguchi, R., 2009, "A Device-Oriented Definition of Functions of Artifacts and Its Perspectives", *Functions in Biological and Artificial Worlds: Comparative Philosophical Perspectives*, MIT Press, pp. 203-221.

[31] Boorse, C., 2002, "A Rebuttal on Functions", *Functions: New Essays in the Philosophy of Psychology and Biology*, Oxford Univ. Press, pp. 63-112.

[32] Vermaas, P.E. and Houkes, W., 2006, "Technical functions: A drawbridge between the intentional and structural natures of technical artefacts", *Studies in History and Philosophy of Science*, **37**, pp. 5-18.

[33] Johansson, I., 2006, "The constituent function analysis of functions", In: *Science—A Challenge to Philosophy?*, Peter Lang , pp. 35–45.

[34] Johansson, I., Smith, B., Munn, K., Tsikolia, N., Elsner, K., Ernst, D., and Siebert, D., 2005, "Functional Anatomy: A Taxonomic Proposal", *Acta Biotheoretica*, **53**(3), pp. 153-66.

[35] Chittaro, L., Guida, G., Tasso, C., and Toppano, E., 1993, "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**, pp. 1718–1751.

[36] Rosenman, M. A. and Gero, J. S., 1998, "Purpose and function in design: From the socio-cultural to the technophysical", *Design Studies*, **19**, pp. 161–186.

[37] Garbacz, P., 2006, "Towards a standard taxonomy of artifact functions". *Applied Ontology*, **1**(3/4), pp. 221-236.

[38] Gangemi, A. and Mika, P., 2003, "Understanding the semantic web through descriptions and situations", In *Proc. of the Int'l Conf. on Ontologies, Databases and Applications of Semantics*.

[39] Masolo, C., Guizzardi, G., Vieu, L., Bottazzi, E, and Ferrario, R., 2005, "Relational roles and qua-individuals", In *Proc. of the AAAI Fall Symposium on Roles, an interdisciplinary perspective*.

[40] Vieu, L., Borgo, S., and Masolo, C., 2008, "Artefacts and Roles: Modelling Strategies in a Multiplicative Ontology", In *Proc. of the 5th International Conf. on Formal Ontology in Information Systems* (FOIS 2008), pp. 121-134.

[41] Maier, J. R. A. and Fadel, G. M., 2003, "Affordance-based methods for design", In *Proc. of DETC 03*, DTM-48673.

