Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2013), *27*, 259–269.

doi:10.1017/S0890060413000267

RESPONSE PAPER

Ontological characterization of functions: Perspectives for capturing functions and modeling guidelines

YOSHINOBU KITAMURA[1] AND RIICHIRO MIZOGUCHI[2]
[1]Institute of Scientific and Industrial Research, Osaka University, Osaka, Japan
[2]Japan Advanced Institute of Science and Technology, Nomi, Ishikawa, Japan

(RECEIVED March 1, 2012; ACCEPTED March 1, 2013)

Abstract

The authors have been involved in ontological modeling of function for over 15 years. As an instance of the revisionary approach discussed in Vermaas's position paper, we have proposed an ontological definition of function and a modeling framework based on it, which has been deployed in industry. In addition, as an instance of the overarching approach, we have proposed a reference ontology of function that explains some kinds, definitions, and practical expressions of functions. In this paper, we explain our methodology in an overarching approach based on *perspectives for capturing functions*. When one captures a function of an artifact, one focuses on a specific aspect of the artifact from a specific perspective. In this paper, we conceptualize such perspectives behind the reference ontology. In addition, based on our experiences in deployment in an industrial setting, we report some solutions, such as ontological modeling guidelines, for overcoming some of the difficulties faced in the practical functional modeling approach described in Eckert's position paper. Our findings suggest that such solutions will help engineers to describe consistent functional models compliant with a single definition of function.

Keywords: Engineering Practice; Function; Knowledge Modeling and Sharing; Ontology

1. INTRODUCTION

As pointed out in Vermaas's position paper in this Special Issue and in survey papers (e.g., Perlman, 2004; Wouters, 2005; Erden et al., 2008), although much research has been carried out on the definition of function in several research areas, such as engineering design (Hubka & Eder, 1988; Pahl & Beitz, 1996; Umeda et al., 1996; Hirtz et al., 2002), artificial intelligence (Goel, 1992; Lind, 1994; Sasajima et al., 1995; Chandrasekaran & Josephson, 2000), philosophy (Cummins, 1975; Johansson et al., 2005; Dipert, 2006; Vermaas & Houkes, 2006), and ontology research (Kitamura et al., 2002, 2006; Arp & Smith, 2008; Borgo et al., 2011), there is no common definition of function, and the relationships among the proposed definitions remain unclear.

The authors have been involved in ontological modeling of function for over 15 years. We have proposed an ontological definition of function (Sasajima et al., 1995; Kitamura et al., 2002, 2006) as an instance of the revisionary approach discussed in Vermaas's (2013) position paper. We have also established a taxonomy of function as a hierarchy of functional concepts in the form of transitive verbs, called a functional concept ontology (Kitamura et al., 2002).

Since then, its modeling framework has been deployed in industry (Kitamura et al., 2004, 2006). Although the practical deployment has been successful, achieving real benefits in daily engineering activities, we have been faced with some of the same difficulties and practical expressions of *function* reported in Eckert's (2013) position paper. In practice, engineers tend to describe functions of artifacts without a clear definition of function in an ad hoc manner. As a result, the function of a certain artifact is described differently by different engineers. For example, the function of an electric fan can be variously represented as "to move air," "to cool the human body," or "to make people comfortable." Such inconsistency makes it difficult to share and reuse functional models.

This practical experience was the motivation for us to propose a reference ontology of function (Kitamura et al., 2007; Mizoguchi & Kitamura, 2009), which categorizes other existing definitions of functions and practical expressions, including our definition of function. Its aim is to harmonize different definitions and practical expressions of functions with clear relationships. It can be regarded as an instance of Vermaas's (2013) overarching approach. With a clear understanding of the relationships among the definitions and practical ex-

Reprint requests to: Yoshinobu Kitamura, 8-1 Mihogaoka, Ibaraki, Osaka, 567-0047, Japan. E-mail: kita@ei.sanken.osaka-u.ac.jp

pressions, we can find the commonalities and differences among them. This will contribute toward providing engineers with some clearly differentiated definitions and expressions for consistent functional modeling and will contribute to interoperability as well. In addition, we have investigated different kinds of function in a phase-oriented model along the product life cycle (Kitamura & Mizoguchi, 2009, 2010).

In this paper, as a response to the position papers mentioned above (Eckert, 2013; Vermaas, 2013), we explain our methodology for the reference ontology and for the phase-oriented model for accommodating and differentiating the existing definitions and practical expressions of function, especially those used in engineering practice.

The core idea of our methodology is that one of the causes of the differences among them is the difference of *perspectives for capturing function*. When one captures the function of a certain entity, one focuses on a specific aspect of the entity from a specific perspective. For example, the three functions of an electric fan mentioned above can be explained in terms of the differences of the *causal scopes* on which the engineers focus for capturing the function, as discussed later. Such causal scope is one of the kinds of perspectives that we want to identify here. We call such a perspective the *capturing perspective* in this paper.

In this paper, we discuss various kinds of perspectives for capturing functions, illustrated with examples. We show that such perspectives can explain some of the examples reported in Eckert (2013). The reference ontology and the phase-oriented model are intuitively based on such perspectives but do not explicate them. Thus, in this paper, we try to conceptualize the rationale behind them, which is implicit in our previous papers, as kinds of perspectives for capturing functions.

In addition, as a response to Eckert (2013), based on our experiences gained in practical deployment (Kitamura et al., 2004, 2006), we report some solutions, such as modeling guidelines, for overcoming some of the difficulties faced in practical functional modeling pointed out by Eckert (2013). We have shown that such solutions help engineers describe functional models based on a single ontological definition of function and will be beneficial for functional modeling in industry.

This paper is organized as follows. Section 2 discusses various kinds of perspectives for capturing functions. In Section 3, we explain the examples of the functions reported in Eckert (2013), using the kinds of perspectives identified in Section 2. Then, we suggest some appropriate perspectives for capturing functions from the viewpoint of composability. Section 4 reports some practical solutions found in our deployment, which contribute to consistent functional modeling based on a stable perspective. Section 5 gives some concluding remarks.

2. PERSPECTIVES FOR CAPTURING FUNCTIONS

2.1. Causal scope

When an entity performs a function, there are changes of values of physical (or mental) attributes. The changes form so-called causal chains. In the example of an electric fan used in the usual way (i.e., a fan supplied with electricity and pointing toward a person), when the fan is working properly, there is the following causal chain:

1. to rotate the fan blades,
2. to move air in front of the fan,
3. to move air around the person,
4. to promote evaporation of water at his/her skin,
5. to cool the skin, and
6. to make him/her comfortable.

The *causal scope*, as a kind of *capturing perspective*, represents which change in a causal chain like the above is focused on. When one focuses on a specific change in a causal chain for capturing the function of an entity, it implies that one focuses on the scope of the location of that change, which extends from the location of the entity to the location of that change.

In the following subsections, we discuss some kinds of *causal scope perspectives*. The relationships among them are shown in Figure 1. We give an ID code (*C#*) to each kind, where *C* represents a causal scope perspective and # represents an ID number based on the extent of the scope, which does not correspond to one of the causal point numbers above.

The causal scope perspectives are categorized into *C1* to *C5*. We can make a distinction between *C1* and *C2*, on the one hand, and *C3* to *C5*, on the other hand, based on the notion of a "system boundary," which is the interface between the whole system and an intentional agent (called an *end user* or just a *user*). By the whole system, here we mean the system that is composed of subsystems and/or components (based on the so-called device ontology; Mizoguchi & Kitamura, 2009), that is the largest and outermost device that has an interface to an end user, and that is directly operated by the end user for his/hers purpose. The effect achieved by the whole system on the user is categorized into one of *C3* to *C5*, as explained in Section 2.1.1, and is called an *external function*, as discussed in Section 2.1.2. The effect achieved by components within the whole system is categorized as *C1* (*C1-1* and *C1-2*) or *C2* and is called the *component function*, which is discussed in Section 2.1.3. In the following subsection, we first explain *C3* to *C5* based on the notion of the system boundary.

2.1.1. System boundary

Based on the notion of the system boundary discussed above, we can identify the following three kinds of *causal scope perspectives:*

C3. The system's input and output: The example is "to move air" in the case of the electric fan. This focuses on the causal change (2) and thus focuses on the airflow at the fan's input and output as locations. Those who capture this function as a function of the fan regard the fan as a black box system and focus on the system's di-

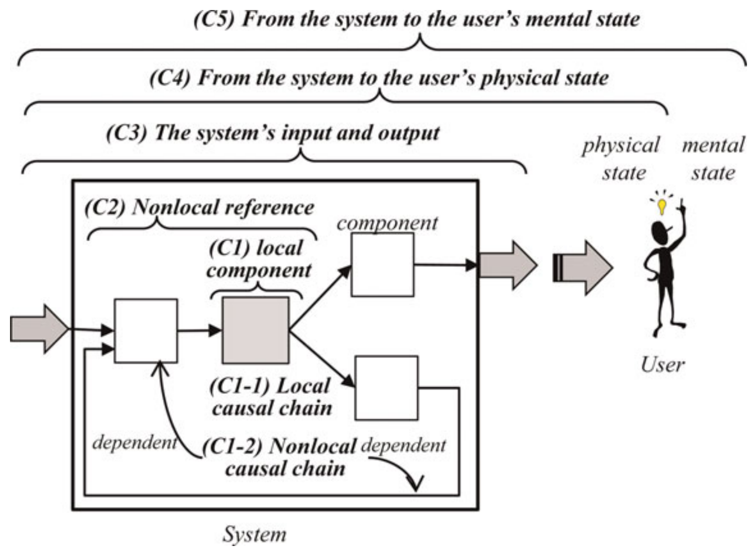

Fig. 1. The kinds of causal scope perspectives. [A color version of this figure can be viewed online at http://journals.cambridge.org/aie]

rect input and output only within the system boundary in terms of location.

C4. From the system to the user's physical state: Consider the example "to cool the human body." This focuses on the causal change (5), which is not a direct result of the fan but involves other phenomena around the user. In other words, the focal causal scope extends to the user, outside of the system boundary.

C5. From the system to the user's mental state: Consider the example "to make him/her comfortable." This focuses on the causal point (6) and refers to a person's cognitive mental state (i.e., to feel comfortable). The causal scope extends to the user's mental state, outside of the system boundary.

In the reference ontology, these functions are categorized as a device function, a physical environmental function, and an interpretative function, respectively, though these kinds of perspectives were left implicit (Kitamura et al., 2007).

Some researchers have pointed out similar distinctions. The environment function (Chandrasekaran & Josephson, 2000), which is an effect on the environment, is similar to ours from the *(C4)* perspective. Goel (2013) points out the necessity for "multiple perspectives on a system" for this distinction as Principle 11. Some researchers distinguish purpose from function (e.g., Hubka & Eder, 1988; Lind, 1994; Gero & Kannengiesser, 2002), where the purpose represents a human-intended goal in a similar sense to perspective (*C4*) or (*C5*). Dipert (2006) points out such goal dependence with respect to some agent and a level of description as well.

2.1.2. Component and external system

From the perspective of the *causal scopes* within the system, we can distinguish between a *component function* and an *external function* (Kitamura & Mizoguchi, 2010) based on the notion of the system boundary discussed in Section 2.1. The causal point (1) in the causal chain of the fan represents a component function of a motor, which is a component embedded in the fan and contributes to the functioning of the fan as the whole system. In short, the component function performed by a component contributes to the achievement of the goal determined by the whole (or the smallest larger sub-) system, whereas the external function performed by the whole system contributes to the achievement of the goal directly determined by the user's intention.

Therefore, the kinds of causal scope perspectives include (*C1*) *the local component* for the component function as well. The external function can be explained by the kind (*C3*) *the (whole) system's input and output* discussed above. We will use these kinds of function in Section 2.5.

2.1.3. Locality of component function

The perspective (*C1*) *local component* should be investigated further. For example, let us consider a two-port fluid valve called a "flow-control valve." When the input signal to the valve is changed by a controller (or a user), the fluid flow rate (i.e., the amount of fluid that flows in a given time) is changed. Therefore, many people would describe its component functions as "*to change the fluid flow rate.*" However, this is not truly "local" but is dependent on components in the fluid system, such as a pump. We believe a truly local function is, "*to change the difference between the pressure at the input port and that at the output port.*" The reason for this is that the causal chain for changing the flow rate is as follows:

1. When one changes the control signal to the valve for decreasing the flow rate, the area where fluid flows is narrowed by a disc.
2. The difference between the pressure at the input port and that at the output port of the valve increases.

3. If the pump in the system is a variable-flow type (a so-called variable displacement pump) with a controller that regulates the flow rate according to the pressure of the total load, the pump decreases the flow rate at its outlet.
4. The flow rate through the valve decreases.

The *changing the flow rate* function refers to change (4) and is thus dependent on the type (and/or on the control) of the pump at change (3).

In the case of a positive (fixed) displacement pump, in contrast, the pump outputs a fluid at a fixed flow rate; hence, the flow-control valve itself cannot reduce the flow rate of the fluid[1] and so just the flow speed at the valve increases. Anyway, the *changing the flow rate* function is dependent on the other components.

In addition, the same valve is sometimes called a "level-control valve." It is also dependent on other components, such as a tank, as well as the system configuration. A couple of assumptions are needed, such as the tank is located upstream of the valve (without any branch) and the input flow rate to the tank is stable to some extent, and hence the level of fluid contained in the tank is controlled by the output flow rate of the tank. Thus, this function is not local either.

Based on the observation above, we can distinguish the following three additional kinds of causal scope perspectives with the example of a flow-control valve:

C1-1. Local causal chain: The entire causal chain is within the component. It depends only on conditions regarding the local inputs and refers to the change at the output port. Example: "to change the pressure difference."

C1-2. Nonlocal causal chain: The expression itself is local (i.e., a subtype of *C1*). Part of the causal chain, however, includes the change at another component in the system. Therefore, it depends on another component and/or the system configuration. Example: "to change the flow rate."

C2. Nonlocal reference: The expression itself refers to another component (e.g., "to change the level"). This is not from the (*C1*) *local component* perspective but is from a perspective inside the system, unlike (*C3*) *the system's input and output* perspective.

Note that the (*C1-1*) *local causal chain* perspective has local conditions (assumption or dependence) for working properly (e.g., the existence of fluid at the input port). We discuss here the causal locality of such conditions.[2] This causal locality is very important for composability of the component models, as discussed in Section 3.2.

2.2. Temporal aspect

A (discrete) change of an attribute of an entity can be captured as a difference of its values from time point t_1 to time point t_2. In other words, here we are concerned with the time interval from t_1 to t_2. The same type of changes can imply different meanings for different phases/stages of the time intervals. For example, when one describes the function of a heater as "a heater increase the temperature," it can imply:

- *"A heater increases the temperature of the air at a specific location in a room compared with a previous point in time.":* This refers to the change in the air temperature at a specific absolute location in real time, which is the same sense as in computer-based numerical simulation.
- *"A heater increases the temperature of the air at the output port to a higher temperature than at the input port.":* This refers to the change in the air temperature while the air flows from the input port to the output port of the heater. Precisely, this change represents the difference in temperature during the time interval from t_1 to t_2, where a specific amount of air flows at the input port at time point t_1 and the same amount of air flows at the output port at time point t_2. These time points are specified relatively with respect to the operating device.
- *"In comparison with the original design, the redesign of the heater increases the highest temperature of the air.":* This refers to the change in the highest temperature of the air between time point t_1 at the original design and time point t_2 after redesigning the heater.

Thus, there are three kinds of *temporal aspect perspective: absolute functioning time* (*T1*), *flowing-object functioning time* (*T2*), and *designing time* (*T3*).[3] Note that many engineers use just "a heater increases the temperature," unaware of these differences. As discussed in Section 2.5, from the viewpoint of the product life cycle phase, (*T1*) and (*T2*) correspond to (*P5*), whereas (*T3*) refers to a different product life cycle phase where t_1 and t_2 correspond to (*P1*) and (*P2*), respectively. Nevertheless, in our industrial deployment, we found many such expressions that can be categorized into (*T3*). For instance, consider the function of diamond powder in a cutting machine. The function can be expressed as the following:

---

[1] If the system were to contain, for example, a *pressure*-control (relief) valve, which causes the fluid to flow out according to a threshold pressure value, the flow rate of the fluid in the system would decrease.

[2] A component has the possibility of performing multiple functions. When such a component is embedded in a system, one of the functions is selected for achieving the system's goal, as discussed in Section 2.1.2. In this sense, the component function is dependent on the system ("how the component is used"). Here we discuss the causal scopes for capturing a function of the component after the system configuration has been fixed and the usage has been selected. Even after that, one can capture different functions according to the causal scopes, as discussed here.

[3] The effect of function at design time determines the "capacity" of the device after manufacturing based on the redesign. The valve's function "to change the difference between the pressure at the input port and that at the output port" discussed in Section 2.1.3 comes from a combination of (*T1*) and (*T2*) temporal perspectives.

- *"The diamond powder increases the frictional coefficient of the cutting blade":* When the cutting machine is functioning, the frictional coefficient is kept high and does not increase while working. The increase refers not to a change in functioning time but to a comparison between two design cases with and without the diamond powder. We will revisit this example in Section 4.2.

This illustrates the need for a *temporal aspect* as a kind of perspective for capturing function.

2.3. Material–energy–information level

A physical change can be captured at the material level or at the energy level, and some attributes can be interpreted as information. Therefore, we can identify three levels as kinds of perspectives: (*L1*) *material level*, (*L2*) *energy level*, and (*L3*) *information level*. For example, the function of an electric motor can be captured as either *"to rotate a shaft"* from the (*L1*) *material level* perspective or *"to convert electrical energy to mechanical (rotational) energy"* from the (*L2*) *energy level* perspective.

Similarly, multilevel flow modeling (Lind, 1994) treats functions of energy, mass, action, and information, and functional basis (Hirtz et al., 2002) has taxonomies of energy and material.

2.4. Intentional aspect

The function of an artifact is usually regarded as one tightly related to the intention of the designer or user. When the artifact is working, there might be other phenomena (causal chains) that are unintended by the designer and/or user, such as a so-called side effect. In addition, one can imagine possible unintended phenomena such as faults and abnormal behaviors. To avoid the negative influence of such unintended phenomena in the normal functioning processes of the system, there might be a function either to prevent such unintended phenomena or to stop (or compensate for) the influences of the unintended phenomena. In our extended theory (Kitamura et al., 2011), we can capture unintended phenomena based on the same modeling methodology. Therefore, from the intentional aspect, we can identify the following kinds of perspectives: (*N1*) *functioning process*, (*N2*) *unintended phenomenon*, (*N3*) *preventing process*, and (*N4*) *compensating process*.

For example, one possible causal chain in an electrical circuit from the (*N2*) *unintended phenomenon* perspective is "a $device_1$ generates heat," "the heat propagates to another $device_2$," "$device_2$'s temperature exceeds its allowable limit," and then "$device_2$ malfunctions due to overheating." To prevent this, the circuit might include a current-controller for $device_1$ that controls the current according to the temperature of $device_2$. From the (*N3*) *preventing process* perspective, its function can be captured as "to prevent the generation of excessive heat by $device_1$." The circuit might also include an electric fan for cooling $device_2$. From the (*N4*) *compensating process* perspective, its function can be captured as "to compensate for the heat" to prevent it influencing $device_2$.

Note that there are many preventing and compensating processes in practical machines, and it is impossible to enumerate all such processes in advance. In practical modeling, only the processes focused on by the designer are described. In addition, by "unintended phenomena," we mean phenomena that are not intended by the designer, although other designers or users might intend such phenomena.[4] Furthermore, "unintended phenomena" and "malfunctioning" are different notions. All malfunctions are unintended phenomena but not vice versa. Many unintended phenomena have no relationship with function (i.e, not harmful to the functioning but merely side effects).

The practical needs for clearly modeling such processes are pointed out in Section 5 in Eckert (2013) as well. Together with colleagues, we have developed an extended modeling framework that can represent all these phenomena and their relationships depicted in three-dimensional space (Kitamura et al., 2011).

2.5. Product life cycle phases

An engineering product has phases in its product life cycle in which transitions are caused by engineering acts such as design, manufacturing, and use. Function is captured in different ways according to the phases focused on. We have explained different kinds of function in such product life cycle phases (Kitamura & Mizoguchi, 2009, 2010). Here, we briefly describe some of the results. In the phases related to use and manufacturing of a product (an entity), we can distinguish the following two functions:

- During the actual use phase (working phase): When the entity is actually used by an agent (a user), *actual functions* are performed. Precisely speaking, in our theory (Kitamura et al., 2006), an actual function is a role played by a behavior as a process under a specific context (of use).[5] Therefore, it occurs in the use phase, exists outside the entity, and is dependent on context (of use). This perspective is called (*P5*) *actual use phase.*[6]
- After-manufacturing phase (come-into-existence phase): As a result of manufacturing the entity, the entity has a capacity to realize specific behaviors based on its physi-

[4] Some of the unintended phenomena would be used to realize functions by users. We call such functions accidental function, as discussed in Kitamura and Mizoguchi (2010).

[5] *Role* here is a technical term in ontological engineering. By role, we mean something that is played by an entity under a context and cannot be defined without referring to the context (Mizoguchi et al., 2007).

[6] There are other kinds of perspectives based on the product life cycle: (*P1*) *before-design*, (*P2*) *after-design*, (*P4*) *intending to use*, and (*P6*) *malfunctioning*. Therefore, the ID numbers of these perspectives are not sequential. Note that (*P4*) represents a situation where a user has an intention to realize a specific use by using an artifact. See Kitamura and Mizoguchi (2009, 2010) for more details.

cal makeup. One of the behaviors could play an actual function as a role according to the context of use. We call the capacity of a behavior to become a function a *capacity function*. It represents a *capacity to perform an actual function*, which is potential and hidden. This perspective is called (*P3*) *after-manufacturing phase.*

These two kinds of function can explain the difference between many definitions in engineering and those in philosophy. In many definitions in engineering (e.g., Lind, 1994; Sasajima et al., 1995; Pahl & Beitz, 1996; Umeda et al., 1996; Hirtz et al., 2002; Kitamura et al. 2006), a function is directly related to a process performed by an artifact when the artifact is used. For example, Umeda et al. (1996) define a function as "a description of behavior abstracted by a human through recognition of the behavior in order to utilize it." Therefore, it corresponds to the actual function.

In contrast, in many definitions in philosophy, a function is a special feature of an artifact (Perlman, 2004) and what the artifact has or what is ascribed to the artifact. For example, in causal-role function analysis (Cummins, 1975) and intention/cause/evolution theory (Vermaas & Houkes, 2006), a technical (artifact) function is regarded as a special kind of capacity to be ascribed to an artifact. Thus, it corresponds to a capacity function.

Arp and Smith (2008) propose a sophisticated definition of artifact function under a generic definition of functions, including biological functions. This definition regards function as a *realizable entity* ("realizable dependent continuant"), which is similar to a capacity function and includes the component function only. The reader is referred to Kitamura and Mizoguchi (2010) for details and more explanation.

2.6. Summary

Figure 2 shows a summary of the perspectives discussed thus far. These perspectives are, in principle, orthogonal to each other. However, there are some exceptions: (*T1*) and (*T2*) are related to (*P5*), and (*T3*) is related to (*P1*) and (*P2*).

3. DISCUSSION

3.1. Explanation of the Eckert examples

In this section, we try to explain the examples of the functions of a hydraulic pump shown by Eckert (2013). The pump is an axial-piston pump "used in off-road vehicles for lifting or moving heavy parts" (Eckert, 2013). It has a rotational shaft that is connected to an external (electric) motor. It has an inlet port and an outlet port for oil. According to Eckert (2013), "All subjects agreed that the main function of the pump was pumping the oil." However, "Two subjects mentioned that the pump could also be a motor that 'transforms the flow of oil into a rotation.' " This is very true, as discussed below.

1. *"Pumping the oil":* As mentioned above, precisely speaking, this is dependent on the usage of this device whereby the shaft torque is its input and the outlet oil is its intended output. When we call this device a "pump," this usage is implied. If we can ignore this dependence, this function is basically from the following perspectives: (*C1-1*) *local causal chain*, (*T2*) *flowing-object functioning time*, (*L1*) *material level*, (*N1*) *functioning process*, and (*P5*) *actual use phase.* This, however, does not mention the pressure of the oil. The relationship among the flow rate, the pressure at the output port, and the power (torque) is dependent on the type of pump and/or the controller, similarly to the flow-control valve discussed in Section 2.1.3. In the case of a so-called positive displacement pump, it outputs fluid at a fixed flow rate. Because the value of the flow rate of the output fluid from the pump is locally determined by the pump, we can say that this is from the (*C1-1*) *local causal chain* perspective. In the case of a variable displacement pump with flow-control, however, the flow rate of the output fluid is dependent on the total pressure loss and/or the load of the fluid system. In this case, the function can be regard as coming from the (*C1-2*) *nonlocal causal chain* perspective.

C. Causal Scope (causal point and its location of changes focused on)		
	(C1) Local component	(C1-1) Local causal chain
		(C1-2) Nonlocal causal chain
	(C2) Nonlocal reference	
	(C3) The system's input and output	
	(C4) From the system to the user's physical state	
	(C5) From the system to the user's mental state	
T. Temporal Aspect (phases/stages of time interval of change focused on)		
	(T1) Absolute functioning time	
	(T2) Flowing-object functioning time	
	(T3) Designing time	
L. Material–Energy–Information Level (level of change focused on)		
	(L1) Material level	
	(L2) Energy level	
	(L3) Information level	

N. Intentional Aspect (intentionality of a designer or a user)	
	(N1) Functioning process
	(N2) Unintended phenomenon
	(N3) Preventing process
	(N4) Compensating process
P. Product life-cycle phases (phases focused on for capturing functions)	
	(P1) Before-design phase
	(P2) After-design phase
	(P3) After-manufacturing phase
	(P4) Intending to use phase
	(P5) Actual use phase
	(P6) Malfunctioning

Fig. 2. Summary of the kinds of capturing perspectives discussed in Section 2.

2. *"Transforms the flow of oil into a rotation":* The rotational torque here is the intended output, and thus this device is a so-called hydraulic motor or actuator. This function is captured from the (*C1-1*) *local causal chain* perspective, and thus is truly local. Other perspectives are the same as the above.

In the following, we assume that the hydraulic pump is used not as a motor but as a pump. Based on this assumption, all of the functional expressions shown in Eckert's (2013) table 2 can be explained as follows[7]:

3. *"Convey fluid/oil":* This might imply that the pump conveys the oil to other components. Therefore, this is from the (*C2*) *nonlocal reference* perspective. Alternatively, it might imply that the pump conveys the oil from the outlet of the pump to the inlet through the whole fluid circulation system, which could be expressed as *"to circulate oil."* This function is from the (*C1-2*) *nonlocal causal chain* perspective.
4. *"Provide pressure/oil pressure":* This function can be interpreted as either of the following more precise expressions: "to give a certain pressure to the output oil" or "to make a pressure difference between the input and output oil." The former is not local and is from the (*C1-2*) *nonlocal causal chain* perspective, because the value of the output pressure is determined according to the balance of the total pressure loss of the fluid system and the controller. The latter is truly local and can be regarded as one captured from the (*C1-1*) *local causal chain* perspective.
5. *"Provide oil pressure for the brake system of a lorry":* This is from the (*C2*) *nonlocal reference* perspective.
6. *"Acceleration of fluid, providing pressure":* This is a combination of expressions (1) and (4).
7. *"Convert mechanical into hydraulic energy":* This is from the (*L2*) *energy-level* perspective. This is also truly local, from the (*C1-1*) *local causal chain* perspective.
8. *"Convert mechanical rotation into hydraulic pressure and flow":* This refers to changes in the kind of medium of the power and the kinds of attributes. This is from the (*C1-1*) *local causal chain* and the (*L1*) *material-level* perspectives.
9. *"Convert power":* This is from the (*L2*) *energy-level* perspective as well.
10. *"Provide a volume flow from A to B":* This is also from the (*C2*) *nonlocal reference* perspective.
11. *"Providing oil pressure/flow → convert mechanical into hydraulic energy":* This is a combination of expressions (7) and (10).

In summary, the differences among all the expressions reported in Eckert (2013) can be explained in terms of the kinds of perspectives for capturing functions proposed in this paper. This demonstrates the richness of those kinds of perspectives, though we do not claim that they are exhaustive.

Such explanations of the functions in Eckert (2013) clarify the reason why each function is described as such, the differences among them, and the reason for the variety of functional expressions for the same pump.

In addition, such differentiation shows an important characteristic, namely, the "locality" of these expressions. Among the expressions above, the truly local ones are (1), (4), (7), (8), and (9), though they are also based on an assumption that the pump is used as a pump. Expressions (7) and (9) are at the energy level. All other expressions depend on the system configuration in some sense.

3.2. Locality and composability

The locality of functional expression is a very important characteristic for knowledge modeling of artifacts as components. It ensures that the model using the expression is dependent only on the component itself and is independent of the system configuration. This enables us to reuse the functional model of the component in any system and is called the "composability" of the model. De Kleer and Brown (1984) note it as the no-function-in-structure principle, which has been regarded as a fundamental principle in the field (Erden et al., 2008). It is implicitly assumed in the function-oriented case-based design (Goel, 1992) and the repository of component models (e.g., Hirtz et al., 2002) as well.

On the basis of this observation, at least from the viewpoint of compositional modeling for engineering, the functional expression might be better captured from a single combination of the following perspectives: (*C1-1*) *local causal chain*, (*T2*) *flowing-object functioning time*, (*N1*) *functioning process*, and (*P5*) *actual use phase.*[8] Our definitions of an artifact function and its behavior are captured from these perspectives (Kitamura & Mizoguchi, 2004; Kitamura et al., 2006).

As reported in Eckert (2013), it is not an easy task to enforce the use of a definition of function on engineers in practice. We believe, however, that it is not impossible and have found some solutions in our industrial deployment. We briefly report them in the next section.

4. ONTOLOGY-BASED PRACTICAL SOLUTIONS

4.1. A functional modeling framework, its deployment, and practical issues

The authors have established a suite of ontologies as a functional modeling framework of artifacts, which includes a definition of function, behavior, and devices; a taxonomy of function (called a functional concept ontology); and types of functional knowledge (Sasajima et al., 1995; Kitamura et al., 2002; Kitamura & Mizoguchi, 2004; Kitamura et al., 2006). A function is

[7] Here we show only the perspectives that differ from those of the "pumping the oil" function.

[8] The (*C1-1*) local causal chain seems to exclude the environmental situation. The environmental situation is, however, represented by a function context for use. The function itself represents a local effect for composability.

defined as "a result of teleological interpretation of behavior under a goal" (Sasajima et al., 1995), and "a role played by a behavior in a teleological context" (Kitamura et al., 2006) is defined as an actual function, as discussed before. We have clearly distinguished between a function expressing "what to achieve" and *a way of function achievement* expressing "how to achieve." A functional model of an artifact is described as a *function decomposition tree*, which is a hierarchy consisting of a macro (goal) function and a sequence of micro (method) functions, which achieves the goal function (called an "is-achieved-by" relation) with a way of function achievement as the principle of its achievement.

As mentioned above, our definition of function is a combination of the following perspectives explained in Section 2: (*C1-1*) *local causal chain*, (*T2*) *flowing-object functioning time*, (*N1*) *functioning process*, and (*P5*) *actual use phase*. In other words, a function in the model represents a local effect in a component as a difference of states of a flowing object at input and output ports, which is intended by a designer and is supposed to be realized in an actual use phase when appropriate inputs are given under a use context envisioned by the designer. These perspectives are described as part of the guidelines discussed in Section 4.2. Note that "the way of function achievement" above is a different aspect, and this notion is independent of these perspectives.

The modeling framework has been deployed in industry (Kitamura et al., 2004, 2006). Together with collaborators, we developed software named SOFAST based on the ontological framework. It was successfully deployed in daily work related to production processes. The real benefits include fast and efficient design review, fast patent writing and enhancement of patent claims, improvements in manufacturing machines, and solutions to quality problems (Kitamura et al., 2004, 2006). The productivity of the production-engineering team increased to 166% in the first year after introducing the methodology and to 211% in the second year.

We faced some practical issues in the deployment (Kitamura et al., 2006). We mention two of them here. The first issue is how engineers can describe functional models that are compliant with functional ontologies including the definition of function. It is similar to the problem pointed out in Eckert (2013). The second issue is that it was not easy for engineers to select appropriate functional terms (verbs) from the complex functional concept ontology.

To overcome these difficulties, we have established two extensions to our framework (Kitamura et al., 2006). One is the establishment of ontological guidelines in order to make ontological commitment easier. The other concerns the treatment of lexical terms for representing functional concepts in order to reduce the difficulty in selecting functional concepts. They are summarized below.

4.2. Ontological guidelines

In order to help engineers describe consistent functional models more easily based on ontologies, the authors have established guidelines about functional models (Kitamura et al., 2006). Part of the guidelines is shown in Figure 3. The guidelines show ontological constraints on the contents of models in the form of a checklist written in a natural language. A model author can check a function decomposition tree using the guidelines and can modify it if necessary. The guidelines are categorized into three groups: α, β, and γ for functions, the relations among subfunctions, and the way of function achievement, respectively. The perspectives discussed in Section 2 are related with "function" and thus correspond to the guidelines in the α group. Some relationships are shown in Figure 3.

Figure 4 shows an example of a modification process based on the guidelines of the function decomposition tree of a wire-saw, which is a manufacturing machine used to slice semiconductor ingots. Figure 4a shows the initial model and 4b the revised one. In the example, the whole function "to slice" in Figure 4a is modified into "to split" in 4b, because "to slice" implies "how to split," which is a way of function achievement (a composite way as discussed below). Thus, "to slice" does not follow the guidelines α1 and α4 in Figure 3, and therefore it is modified into "to split apart the ingot," which enables the designer to select ways other than "slicing."

In Figure 4a, "to move table to wire" and "to move wire" are described as sub (method)-functions. However, violating γ2 in Figure 3, the reason why these functions can perform the whole (goal) function is not clear. Moreover, violating β1 in Figure 3, it is unclear which things (called operands) flow between these

α. About functions and behaviors

- **α 1.** A function represents "what to achieve" only and does not imply "how to achieve".
  - **α 1-1.** A device is a black-box. The inside is not shown at any level. (*(C1-1)* and *(T2)*)
- **α 2.** A function represents (a teleological interpretation of) changes in physical things within the system boundary. *(C1-1)*
  - **α 2-1.** Do not describe the designer's activities. (Not *(T3)*)
  - **α 2-2.**, **α 2-3.** …
- **α 3.** The agent of a function should be a "device" in the physical world.
  - **α 3-1**. **α 3-2**. **α 3-3**. …
  - **α 3-4.** A device can be virtual and/or dynamic.
- **α 4.** Decompose functions which imply kinds of operands and/or degrees of results for functions. Such implications are represented as attributes of *ways of function achievement*

β: About relations between sub-functions

- **β 1.** Identify states of operands that flow through sub-functions.
- **β 2.** Time passes basically from a left sub-function to the next-right sub-function along the operand-flows.
- **β 3.** …

γ: About "is-achieved-by" relations and ways of function achievement

- **γ 1.** The "is-achieve-by" relation represents a kind of aggregation
  - **γ 1-1.** The total changes in sub-functions should correspond to changes in the whole function.
  - **γ 1-2**, **γ 1-3**. …
- **γ 2.** A sub-function should explicitly contribute to a macro-function.
- **γ 3.** The *way of function achievement* represents a single principle.
  - **γ 3-1.** Decompose compound principles.
  - **γ 3-2.** Distinguish a *way* from other *ways* at the principle level.
  - **γ 3-3.** If possible, conceptualize neither a tool nor a kind of an operand but a principle.

Fig. 3. A portion of the guidelines for function decomposition trees. Adapted from Kitamura et al. (2006) with permission.

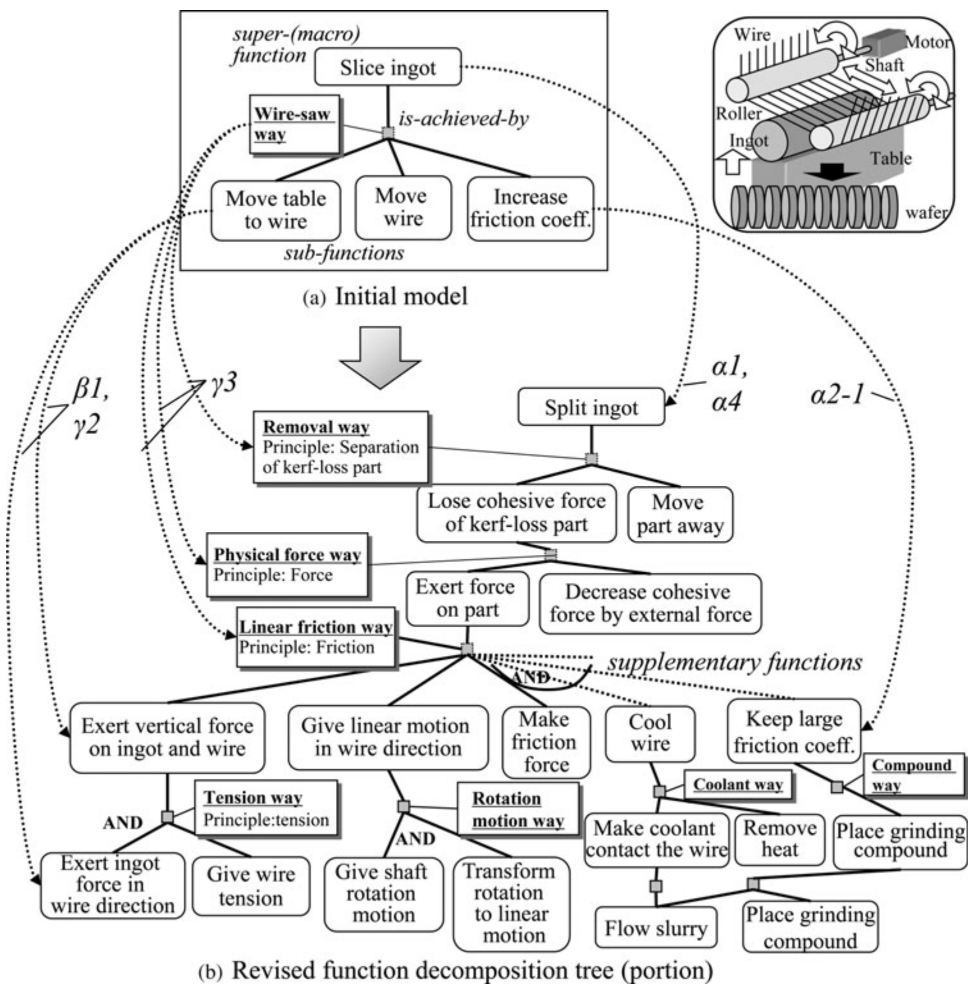

Fig. 4. A modification of a function decomposition tree of a wire-saw portion. Adapted from Kitamura et al. (2006) with permission.

functions. One reason is that there is a missing function, "to exert vertical force on ingot and wire." The "to move table to wire" in reality contributes to this function. Therefore, the wire-saw way in Figure 4a should be decomposed into the three ways shown in Figure 4b following γ3.

In addition, the function "to increase frictional coefficient" shown in Figure 4a violates α2-1 from the (*T3*) *designing time* perspective. Thus, it is revised to the "to keep large friction coefficient" function in (*T2*) or (*T1*), as shown in Figure 4b.

Currently, the guidelines are only for human comprehension. Engineers themselves have to check their models according to the guidelines. Automatic checking for compliance of the functional models with the ontological guidelines using the implemented ontologies is a difficult task and remains the topic of future work.

4.3. Lexical layer

Our experience gained in practical deployment shows the importance of flexible lexical terms for representing functions. Thus, a new version of the software supports four layers of functional terms: an ontology layer, a concept layer, a lexical layer, and a domain-specific layer. The ontology layer is for the full functional concept ontology. The concept layer is a simplified three-level hierarchical version of that ontology for essential functional concepts. The lexical layer includes usual functional words in daily work. The domain-specific layer has vocabulary sets specific to domains or companies. The usual functional words are associated with essential functional concepts. A usual functional word can represent some essential functional concepts and vice versa. For example, the usual functional word "to put together" is associated with the functional concepts "to join" and "to bring into contact." The function "to bring into contact" can be represented as "to place" as well.

Users can select a function in functional models using either essential functional concepts or usual functional words. When a user selects a usual functional word associated with multiple essential functional concepts, the user has to select one from these candidates to reflect the intended meaning. Thus, a function in models is associated with both a usual functional word and an essential functional concept. Therefore, both the user and the computer can search using essential functional concepts independently of a lexical representation of functions.

5. CONCLUSION

In this paper, as a methodology in the overarching approach for accommodating and explaining the differences among existing definitions and practical expressions, we report our methodology based on *perspectives for capturing functions*. We proposed several kinds of such perspectives, and in terms of these perspectives, we explained some existing definitions and many practical expressions, including those reported in Eckert (2013).

Such an overarching approach contributes to interoperability of functional models. We established mappings between our functional ontology and reconciled functional basis (Hirtz et al., 2002) based on a reference ontology and realized interoperable searching of documents using functional terms in either of these taxonomies (Kitamura et al., 2008).

Our main goal here is to enumerate fine-grained perspectives for in-depth understanding of function. We cannot claim their completeness or necessity in nature, however. We demonstrated the usefulness of our framework by showing existing notions of function and practical expressions, including those reported in Eckert (2013). We note that there might be typical combinations of these perspectives, and an investigation of this remains a topic of future work.

Nevertheless, from the viewpoint of the composability of functional models, we could suggest an appropriate set of perspectives for correctly capturing functions, which our functional ontology is based on.

In addition, we summarized some practical solutions for helping engineers describe functional models that are compliant with a single ontology. On the basis of our experience gained from industrial deployment (Kitamura et al., 2004, 2006), we can claim that our ontology-based functional modeling framework with these solutions can work well in industry, and engineers can get real benefits from adopting it.

REFERENCES

Arp, R., & Smith, B. (2008). Function, role, and disposition in basic formal ontology. *Proc. Bio-Ontologies Workshop (ISMB 2008)*, pp. 45–48, Toronto, July 20.

Borgo, S., Mizoguchi, R., & Smith, B. (Eds.) (2011). On the ontology of functions [special issue]. *Applied Ontology 6(2)*.

Chandrasekaran, B., & Josephson, J.R. (2000). Function in device representation. *Engineering With Computers 16(3/4)*, 162–177.

Cummins, R. (1975). Functional analysis. *Journal of Philosophy 72(20)*, 741–765.

De Kleer, J., & Brown, J.S. (1984). A qualitative physics based on confluences. *Artificial Intelligence 24*, 7–83.

Dipert, R.R. (2006). The metaphysical grammar of "function" and the unification of artifactual and natural function. *Proc. 15th Altenberg Workshop in Theoretical Biology: Comparative Philosophy of Technical Artifacts and Biological Organisms*. Altenberg, Austria: Konrad Lorenz Institute for Evolution and Cognition Research.

Eckert, C. (2013). That which is not form: the practical challenges in using functional concepts in design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 27*(3)*, 217–232 [this issue].

Erden, M.S., Komoto, H., Van Beek, T.J., D'Amelio, V., Echavarria, E., & Tomiyama, T. (2008). A review of function modeling: approaches and applications. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 22*, 147–169.

Gero, J.S., & Kannengiesser, U. (2002). The situated function–behaviour–structure framework. *Proc. of Artificial Intelligence in Design '02*, pp. 89–104. Dordecht: Kluwer.

Goel, A.K. (1992). Representation of design functions in experience-based design. *Intelligent Computer Aided Design* (Brown, D., Waldron, M., & Yoshikawa, H., Eds.), pp. 283–308. Amsterdam: North-Holland.

Goel, A.K. (2013). One 30-year case study and 15 principles: implications of an artificial intelligence methodology for functional modeling. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 27(3)*, 203–215 [this issue].

Hirtz, J., Stone, R., McAdams, D., Szykman, S., & Wood, K. (2002). A functional basis for engineering design: reconciling and evolving previous efforts. *Research in Engineering Design 13(2)*, 65–82.

Hubka, V., & Eder, W.E. (1988). *Theory of Technical Systems*. Berlin: Springer–Verlag.

Johansson, I., Smith, B., Munn, K., Tsikolia, N., Elsner, K., Ernst, D., & Siebert, D. (2005). Functional anatomy: a taxonomic proposal. *Acta Biotheoretica 53(3)*, 153–166.

Kitamura, Y., Kashiwase, M., Fuse, M., & Mizoguchi, R. (2004). Deployment of an ontological framework of functional design knowledge. *Advanced Engineering Informatics 18(2)*, 115–127.

Kitamura, Y., Koji, Y., & Mizoguchi, R. (2006). An ontological model of device function: industrial deployment and lessons learned. *Journal of Applied Ontology 1(3/4)*, 237–262.

Kitamura, Y., Koji, Y., & Mizoguchi, R. (2011). FOREST: an ontological modeling framework for product-related processes. *Proc. 7th Int. Conf. Engineering Design in Integrated Product Development (EDIProD 2011)*, pp. 39–49, Wroclaw, Poland, June 30–July 1.

Kitamura, Y., & Mizoguchi, R. (2004). Ontology-based systematization of functional knowledge. *Journal of Engineering Design 15(4)*, 327–351.

Kitamura, Y., & Mizoguchi, R. (2009). Some ontological distinctions of function based on the role concept. *Proc. ASME 2009 Int. Design Engineering Technical Conf. & Computers and Information in Engineering Conf. (IDETC/CIE 2009)*, Paper No. DETC2009-87168, San Diego, CA, August 30–September 2.

Kitamura, Y., & Mizoguchi, R. (2010). Characterizing functions based on ontological models from an engineering point of view. *Proc. 6th Int. Conf. Formal Ontology in Information Systems (FOIS 2010)*, pp. 301–314. Amsterdam: IOS Press.

Kitamura, Y., Sano, T., Namba, K., & Mizoguchi, R. (2002). A functional concept ontology and its application to automatic identification of functional structures. *Advanced Engineering Informatics 16(2)*, 145–163.

Kitamura, Y., Segawa, S., Sasajima, M., Tarumi, S., & Mizoguchi, R. (2008). Deep semantic mapping between functional taxonomies for interoperable semantic search. *Proc. 3rd Asian Semantic Web Conference (ASWC 2008)*, LNCS, Vol. 5367, pp. 137–151. Berlin: Springer–Verlag.

Kitamura, Y., Takafuji, S., & Mizoguchi, R. (2007). Toward a reference ontology for functional knowledge interoperability. *Proc. ASME 2007 Int. Design Engineering Technical Conf. Computers and Information in Engineering Conf., IDETC/CIE 2007*, Paper No. DETC2007-35373, Las Vegas, NV, September 4–7.

Lind, M. (1994). Modeling goals and functions of complex industrial plants. *Applied Artificial Intelligence 8*, 259–283.

Mizoguchi, R., & Kitamura, Y. (2009). A functional ontology of artifacts. *Monist 92(3)*, 387–402.

Mizoguchi, R., Sunagawa, E., Kozaki, K., & Kitamura, Y. (2007). The model of roles within an ontology development tool: Hozo. *Applied Ontology 2(2)*, 159–179.

Pahl, G., & Beitz, W. (1996). *Engineering Design—A Systematic Approach*. Berlin: Springer–Verlag.

Perlman, M. (2004). The modern philosophical resurrection of teleology. *Monist 87(1)*, 3–51.

Sasajima, M., Kitamura, Y., Ikeda, M., & Mizoguchi, R. (1995). FBRL: a function and behavior representation language. *Proc. 14th Int. Joint Conf. on Artificial Intelligence*, Vol. 2, pp. 1830–1836. San Francisco, CA: Morgan Kaufmann.

Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y., & Tomiyama, T. (1996). Supporting conceptual design based on the function–behavior–state modeler. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 10*, 275–288.

Vermaas, P.E. (2013). The coexistence of engineering meanings of function: four responses and their methodological implications. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 27(3)*, 191–202 [this issue].

Vermaas, P.E., & Houkes, W. (2006). Technical functions: a drawbridge between the intentional and structural natures of technical artefacts. *Studies in History and Philosophy of Science 37*, 5–18.

Wouters, A. (2005). The function debate in philosophy. *Acta Biotheoretica 53*, 123–151.

Yoshinobu Kitamura is an Associate Professor of the Institute of Scientific and Industrial Research, Osaka University. He received his bachelor's, master's, and doctoral degrees of engineering in 1991, 1993, and 2000, respectively, from Osaka University. His research interests include ontological engineering, ontology of function and artifacts, ontology-based engineering knowledge management, and conceptual design support.

Riichiro Mizoguchi was a Professor at the Institute of Scientific and Industrial Research, Osaka University, from 1990 to 2012 and is currently a Professor at the Research Center for Service Science, Japan Advanced Institute of Science and Technology. His research interests include knowledge-based systems, theories of role and function, ontological engineering, ontology development, and intelligent learning support systems. Dr. Mizoguchi was president of the International Artificial Intelligence in Education Society and the Asia-Pacific Society for Computers in Education, Vice President of the Semantic Web Science Association, and President of the Japanese Society for Artificial Intelligence. He is currently Associate Editor of *IEEE Transactions on Learning Technologies* and *ACM Transactions on Interactive Intelligent Systems*.