In Proc. of the First Workshop FOMI 2005 - Formal Ontologies Meet Industry, Castelnuovo del Garda (VR), Italy, June 9-10, 2005, CD-ROM (Reformatted version)

An Ontological Model of Device Function and Its Deployment for Engineering Knowledge Sharing

Yoshinobu Kitamura, Yusuke Koji and Riichiro Mizoguchi

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

Abstract. This research aims at promoting sharing of knowledge about functionality of engineering artifacts among engineers. Such functional knowledge shows an important part of designer's intention, so-called design rationale. Sharing design rationale plays a crucial role in team-activities in engineering practice such as designing, trouble-shooting, and maintenance. Nevertheless, in the current practice in industry, engineers have suffered from the difficulty of reusing technical documents of such functional knowledge, since the documents tend to be written in ad hoc manner using each engineer's vocabulary and are specific to products or domains. For resolving these difficulties, we have developed an ontological framework of functional knowledge, which includes an ontology of device and function as conceptual viewpoint and a functional concept ontology as a controlled vocabulary. These ontologies play a role as guidelines or constraints to avoid ad hoc modeling. This framework was successfully deployed in a manufacturing company in Japan in daily activities such as design review, equipment improvement and patent application. This paper firstly discusses some ontological issues of functionality of artifacts. Secondly, we show a definition of the concept of function as role and operational definitions of generic functions. Thirdly, we summarize the authors' experiences of the deployment, and discuss the success factors, difficulties, and their solutions including future work. Lastly, in order to place our definition of function in the related concepts in the literature, other types of function are discussed.

Keywords: Engineering domain, Functionality, Knowledge sharing, Design, Role

1. Introduction

The recent situation in engineering industry requires effective sharing of product knowledge among engineers for engineering activities such as designing and manufacturing during the product life-cycle. It is important to share not only *product data* but also *designer's intention* so-called *design rationale* (DR) of the product [1-2]. It represents justification of the current design including reasons of existence of a component or a sub-system in the system, reasons of design decisions, and design alternatives. Explicit representation of DR plays a crucial role in engineering activities such as design review, product improvement, and facility maintenance. For example, in design review to double check an original design by a team of designers, an explicit description of the original designer's intentions helps other people understand the original design more effectively. In facility maintenance, in order to adjust a working parameter of the facility, a maintenance engineer should understand the reason of the current value which is a result of the designer's decisions.

One of the key concepts to capture such designer's intention is *functionality* of artifacts [1-6]. Intuitively, a function of a product explains what users can get using it (effects or utility of the artifact). A function of a component embedded in a system explains how it contributes to achieving the system's whole-function (so-called "how things work"). It is the reason why the component exists in the system, which is a part of DR.

Nevertheless, sharing of such knowledge about function is difficult in practice, from our experience of collaborative research with a production company. Few CAD/CAM/PDM systems treat such subjective knowledge. Thus, engineers have to rely on documents in natural language. Engineers have been regularly writing various kinds of technical reports/documents and have stored much of those in databases. Unfortunately, however, few such technical documents have been efficiently reused.

One of the reasons of this difficulty is lack of *semantic constraint* for functional knowledge. Without guideline or restriction, functional representation tends to be ad hoc, specific to the target product, and hence not reusable. Although much research has been conducted on the representation of functionality in Artificial Intelligence [2-12], engineering design [13-19] and Value Engineering [20], there is no common definition of the concept of function itself [21]-[23] and semantic constraints are not enough for deriving effective guidelines. For example, one might describe "to weld metals" as a function of a welding machine in a "verb+noun" style in Value Engineering [20]. However, "to weld metals" implies both the metals are joined and their parts are fused. From the viewpoint of functionality in manufacturing, joining is only the goal the designer intends to attain ("what to achieve"), while the fusion can be regarded as a characteristic of "how to achieve that goal". In fact, the same goal, say, "to join", can be achieved in different ways (e.g., using nuts & bolts) without the fusion. If a function of the welding machine is described as "to join", the commonality between two facilities can be found. This issue is not a terminological but *ontological* in order to distinguish "what to achieve" from "how to achieve".

This observation suggests the necessity of an ontological schema for functional knowledge. An ontological schema specifies not only the data structure but also a *conceptual viewpoint* for capturing the target world and a *controlled vocabulary* to describe the knowledge at an appropriate level of abstraction. The *conceptual viewpoint* provides guidelines or constraints on modeling, which helps knowledge authors to describe knowledge consistently, and especially to distinguish "what to achieve" from "how to achieve". On the other hand, the *controlled vocabulary* provides a systematized set of generic verbs representing functionality of devices.

What we need is an ontology about artifacts including functionality, which is a kind of *domain ontology for engineering*. Much work on engineering domain ontologies has been done (e.g., [24-29]). Many of them aim at improvement of interpretability for communication among agents or tools (e.g., [25], [28]). Rather, we aim at "ontology as meta knowledge" as discussed above. One remarkable example of meta-knowledge type is the PhysSys ontology [27]. It, however, has no ontology for functions from the teleological viewpoint.

The authors have been involved in ontology-based modeling of physical systems for many years and have established an ontological framework for functional knowledge [30], [31]. This framework includes an ontology of device and function as conceptual viewpoint and a functional concept ontology as a controlled vocabulary. This framework has been successfully deployed in a manufacturing company in Japan for sharing functional knowledge [32].

This paper discusses ontological issues of functionality of artifacts and summaries the authors' experiences in the ontology deployment. Section 2 discusses fundamental issues as requirements of an ontology about function. Section 3 presents our ontologies about functions. Section 4 discusses the experience in the deployment, success factors and difficulties. Section 5 presents discussion on other types of function and limitations. Then, related work is discussed followed by some concluding remarks.

2. Ontological Issues of Function

This section discusses ontological issues as requirements for reusable and consistent functional knowledge. Our goal here is to define the concept of function, generic functions, and relationships between functions clearly and operationally.

2.1 Definition of Function with Behavior

For clear definitions of functionality, the relationship with "behavior" plays a crucial role. The distinction between function and behavior originates from the qualitative reasoning (QR) research (e.g., [34]). *Behavior* here represents temporal changes of physical quantities of a physical entity. It is objective and independent of the *context* which includes designer's intention, user's aims and the system in which the entity is embedded. In QR, in order to realize reusability (composability) of the component model, context-dependent information is carefully excluded (called No-Function-In-Structure principle [34])[1].

In comparison with behavior, *function* is related to intention of a designer or a user (i.e., teleological) and context-dependent. A behavior can perform different functions according to the context. For example, a heat exchanger can be used as a heater or a radiator. The behavior is the same in any context, that is, a heat flow between two fluid flows. The functions of the heater and the radiator can be "to give heat" and "to remove heat", respectively. This difference of functions is dependent on the embedded system.

Thus, the first issue is to clarify this *teleological interpretation relation* between behavior and function and to define functions based on this relationship. In the literature, this relationship is defined as "means and ends" [8], F-B relationship [9] or "aims-means" [15] (This includes design requirements as well) and causal patterns [3]. On the other hand, the well-established standard taxonomy, called *functional basis*, for functions by the NIST Design Repository Project [19] lacks clear relationship with objective behaviors based on such teleological relationship.

In order to clarify the relationship and to define functions operationally, our approach is to describe a function as "behavior plus information for teleological interpretation" in terms of a set of primitives (called *functional toppings*) as discussed in Section 3.1. Moreover, we define generic functions as constraints on the information as discussed in Section 3.2.

2.2 Function as a Role

The second issue related to the definition of function is the "*role*" concept in ontological engineering research. Intuitively, a *role* is something that can be *played by* an entity in a context. Precisely, in [35], a role is the *secondness* concept which is dependent on a pattern of relationship. In [36], a role is *anti-rigid* (i.e., contingent (non-essential) property for identity), *dynamic* (temporary and multiple), and *founded* (i.e., extrinsic property defined with external concept).

1 Although a behavioral model depends on modeler's assumptions and viewpoint for capturing the target world as the same as all information models do, the concept of "behavior" itself represents context-independent changes.

Similar to these definitions, by role we mean here such a concept that an entity plays in a specific *context* and cannot be defined without mentioning external concepts [37]-[39]. We distinguish *role* (something to be played) from *role-holder* (something playing (holding) a specific role). This distinction is emphasized as *role-taker* by Breuker as well [40]. For example, a man (class constraint for role) can play “husband role” (role concept) in a “marriage” relation (role context), who is called “husband” (role holder). Our ontology editor of an environment for building/using ontology named Hozo has a capability to treat roles [38],[39].

Firstly, a function (and a behavior as its basis) is *founded* [36], since a function of an artifact affects an entity other than the artifact itself (we call *operand*) and causes temporal changes of the operand (we call *behavior of the device* in the sense discussed in Section 2.1). For example, a radiator decreases temperature of the warmer fluid and the definition of the removing-heat function refers to the change of the warmer fluid’s temperatures as input and output. Thus, definition of a function of an artifact requires an operand as an external entity.

Secondly, a function is *anti-rigid* [36] and context-dependent (dynamic), because a function of a specific device can be changed without losing the device’s identity according to a user’s goal (e.g., a chair can be used as a ladder or a hummer) or the system in which the device is embedded as a component (e.g., the heat exchanger mentioned in Section 2.1). Moreover, a function can be performed by different components. On the other hand, a component can perform multiple functions simultaneously. For example, in the cascade configuration of controllers, a valve performs two functions, i.e., to control level of working fluid in a tank and to control flow rate of the fluid from the tank according to operator’s focus [37].

In the literature, similar concepts are discussed. Chandrasekaran and Josephson use the concept of *role* as *natural* (without human’s intention) effects on environment (e.g., the role of cloud is to give rain) and define *function* as “role + intention” [11]. In EPISTLE Framework, the concept of *facility* is defined as a functional thing, capability to perform a function and a service [41]. Fan et al. define *purpose* of an artifact as *default role* which is expected to be done by the artifact [42]. Breuker pointed out function as role and discussed *mental roles* in law [40].

In the relationship between function and behavior discussed in Section 2.1, we say that “a behavior can play a function role”. If a device performs a behavior and the behavior 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 removing-heat *function role* and then a heat exchanger plays the *function-performer role* of removing-heat as a radiator.

In summary, the second issue is to define the concept of functionality as a role (holder) of behavior (and device). Such definition requires a description of the context for functional interpretation. The *functional topping* is a localized representation of the context. Our definition of function as role of behavior with functional context is discussed in Section 3.1.

2.3 Device Ontology and Entity’s Roles

For consistent representation of functions, consistent representation of behavior and system is a very important issue as well. In the design literature, German systematic design approach [13] has provided us with a basic viewpoint to capture functions, in which functions are regarded as the input-output relations of a black-box. The black-boxes can be connected and aggregated. Such a device-centered viewpoint originating from systems dynamics theory is called *device ontology* (e.g., [11],[26],[27]). A device ontology is suitable as a basis for establishing ontologies of functions, since functions are usually considered as what components or devices achieve.

The definition of the device ontology also requires representation of “role”. Let us consider a manufacturing machine called a wire-saw, which is designed to slice a semiconductor ingot into wafers using a wire moved by rollers. To assign a role to this wire is not trivial. It exerts force on the ingots as a *device*, and is moved by rollers as *operand*. In order to assign roles to entities in a consistent manner, we have to maintain the relationship among entity, role and behavior. We will revisit this example in Section 4.2.

Thus, the third issue is to establish a *role assignment system* from the functional viewpoint in order to capture the target world consistently. As discussed in Section 3.1, we extend the conventional device ontology by redefining the concepts of behavior, *conduit* (a subtype of *device*) and *medium* (a subtype of *operand*) for a richer role assignment system.

2.4 “is-a” and “part-of” Relations of Function

The German systematic design approach [13] provides a fundamental relationship among function as well, so-called *function decomposition*. It represents how a function is achieved by a series of sub-functions which are finer-grained functions (we call “*is-achieved-by*” relation), which is a kind of so-called “part-of” relation. Usually (but not always), it corresponds to the whole-part (*aggregation*) hierarchy of physical structures of devices, that is, the whole system, sub-systems and components. In the literature, this relation has been captured as whole-part relation [8], and “degree of complexity” [15] as well as the function decomposition [13].

On the other hand, we can consider "*is-a*" relation among functions. It represents generic or abstract concepts of functionality. Pahl and Beitz defined highly-abstracted functions (called generally-valid functions) [13] . In Hubka and Eder [15], the hierarchy for the "degree of abstraction" of functions represents the specialization of functions with additional conditions. The conditions, however, may sometimes (not always) include the characteristics of a specific method of achieving a function such as "transportation by sea" [15] which has the same difficulty as "welding" as discussed in Introduction.

It is not easy to distinguish the relationships among functions, that is, *part-of*, *is-a*, and the *teleological interpretation* (discussed in Section 2.1). One of the reason is that a function can achieve more generic function in the *is-a* hierarchy (i.e., super-class of the function). The papers such as [4] do not distinguish *teleological interpretation* from the *part-of* relation and thus define that "behavior" is how to achieve a function where distinction between behavior and function is relative.

In summary, the fourth issue is to clarify the relationships about function, i.e., *part-of*, *is-a*, and the *teleological interpretation*. We introduce the concept of *way of function achievement* as conceptualization of the *part-of* relation. Moreover, we define *is-a* relations among them as discussed in the next Section.

3. Ontologies for functional knowledge

This section explains an ontology of device and function (Section 3.1), a functional concept ontology (Section 3.2), and functional knowledge based on these ontologies (Sections 3.3 and 3.4).

3.1 An Ontology of Device and Function

Our ontologies have been defined in an ontology editor of an environment for building/using ontology named Hozo [38],[39]. Figure 1 shows an example and Figure 2 shows a portion of an ontology of device and function. The ontology editor basically supports frame-based representation with slots. Concepts are represented as frames (denoted by nodes in Fig. 1 and 2) with slots (right-angled link) and the *is-a* relations among concepts (straight link with "is-a"). Concepts are categorized into the wholeness concepts composed of part concepts and the relation concepts between the concepts. A wholeness concept has slots of part concepts (part-of relation denoted by right-angled link with "p/o") and slots of attributes ("a/o"). A relation concept has slots of participant concepts (participate-in relation. denoted by "p/i") and the attribute-slots. Figure 1 shows an example of definition of the husband role discussed in Section 2.2 (i.e., a *husband* (role holder) is defined as a *man* (class constraint) playing a *husband role*) in the "marriage" relation concept. The upper right part of Fig. 1 shows its definition from a role-centered view. It is defined also in the "married couple" which is a wholeness concept corresponding to the "marriage relation".

The ontology of device and function consists of an extended device ontology and an ontology of function. In the definition of the extended device ontology in Fig. 2, the *device* concept is defined as a role-holder in *behavioral-relation* between two *physical-entities* (Fig. 2(a)). One of them plays the "*agent*" role, which is called *device*. It operates on the other entity (*operand* which is another role-holder) and changes its *physical-attributes*. A *port* of a device is a virtual interface for propagation of parameter values to another device. Each device is connected to each other through its input and output *ports* (Fig. 2(b)).

The *operand* is something flows through the device and is affected by the device. The operand role can be played by fluid, energy, motion, force, or information. It has *IO-States*, which represents values of *physical-attributes* at a *port* of a device. The pairs of IO-States at input ports of a device and those at output ports of the same device are defined as *behavior* (Fig. 2(c)). It represents objective conceptualization of its input-output relation as a black box.

We extended the conventional device ontology by redefining the concepts of behavior, conduit, and medium. We categorized the meanings of behavior into four types (from B0 to B3) [31]. The definition of behavior above (i.e., behavior based on *IO-States* of the flowing *operands*) is called *B1-behaviour* in terms of [31]. A *conduit* (e.g., a pipe and a shaft) is defined as a special device that transmits an operand without any change in an ideal situation. A *medium* (e.g., steam for heat energy) is something that holds an operand and enables it to flow between devices. A medium role and a conduit role can be played by an entity simultaneously, although a device role and an operand role cannot be. Such multiple role-playing is a solution for the issue in Section 2.3. We revisit this issue in Section 4.2.

A (base-)function role is defined as a role concept which is played by a *behavior* under a *function-context* (Fig. 2(d)). In the function-context, there is a *function-performing relation* among two physical entities and a behavior (Fig. 2(e)). In the relation, a

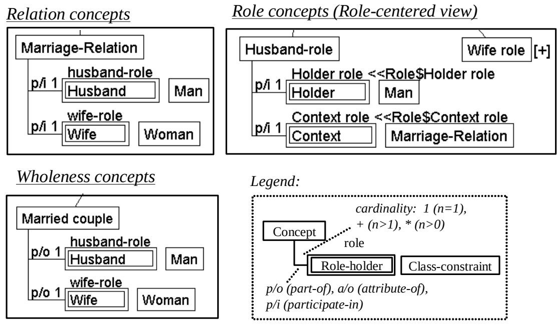


Figure 1. Example of definition of a role concept (husband role) in Hozo [38][39].

behavior plays a base-function role, which is called a base-function role holder (Fig. 2(f)).

The *function-context* represents teleological goals to be achieved by the function. A function-context of a function of a component in a system (called System-Function-Context, Fig. 2(g)) can be determined by a *goal function* and *method* (sibling) functions, which are defined in "*way of function achievement*" relation (Fig. 2(h)). It means that the *goal function* can be achieved by a sequence of functions as *method-functions* (i.e., is-achieved-by relation in Section 2.4) on the basis of a physical principle. The sibling functions are other *method functions*.

These definitions give more detailed definitions of our definition in previous papers [43],[44], that is, a function as a teleological interpretation of a *B1-behavior* under *a goal*. The "goal" is defined as the function-context in detail. The "interpretation" is defined as interpretation of the role of the behavior in the function context. The teleological interpretation of behavior to function can be described using *functional toppings (FTs)*, which are primitives of additional information to behaviors, that is, *Obj-Focus*, *O-Focus*, *P-Focus* and *Necessity*. They represent information about such an *operand* that the designer intends to change (focus of intention). *Obj-Focus* specifies its kind such as substance or energy. *O-Focus* specifies the type of its physical attributes to change (such as temperature and phase). *P-Focus* specifies ports and represents focus on a flow of operand or medium. *Necessity* specifies the necessity of *operands* in the context. Such definition of function resolves the first issue and the second issue discussed in Section 2.

For the example of the heat exchanger mentioned in Section 2.1, its behavior is described as a thermal energy flow between two medium flows. The giving-heat function can be represented as; (1) *O-Focus* on temperature and (2)*P-Focus* on the medium flow which receiving heat (heat destination). On the other hand, the removing-heat function can be represented as (1)the same *O-Focus*, (2) *P-Focus* on another source medium-flow which releases the heat energy and (3) the heat energy is not necessary (*Necessity*). Such functional toppings show the difference between these two functional interpretations.

The values of the functional toppings are determined according to the function-context. The functional toppings (FTs) are localized representation of the surrounding context of the component. This locality contributes to obedience to the No-Function-In-Structure principle discussed in Section 2.1. Moreover, the values of FTs are limited and operational, which are related to behaviors. It limits the space of teleological interpretation of a behavior. In fact, semi-automatic identification of functions can be done by enumerating possible all interpretations from given behavioral model [44].

Moreover, in addition to the base-function, we identified *meta-function*. In comparison to a base-function which is a role of behavior (and device) for operand, a meta-function is a collaborative role played by a *function* for *another function* such as ToDrive and ToProvide. For example, ToDrive means that a function supplies energy driving another function. It is result of teleological interpretation of causal relations among base-functions of different components. Each *base-function* has a specific type of functions (called *function types*) as well, which represents causal patterns of achievement for goals of each base-function of a component such as ToMake and ToMaintain (we redefined the ones in [6]). For more detail on meta-functions, see [44].

3.2 Functional Concept Ontology

We developed an ontology of generic functions (called functional concept ontology) [44], which are sub-classes of the "function" class in the ontology of device and function. Figure 3 shows an overview of our modeling framework and a portion of the functional concept ontology.

It defines about 220 concepts in four kinds of *is-a* hierarchies with such operational definitions. For example, an energy function, "to shift energy", is defined as a behavioral constraint: focused energy moves between two different mediums and semantic constraint on functional toppings as a reflection of function context. It can be defined by the axioms

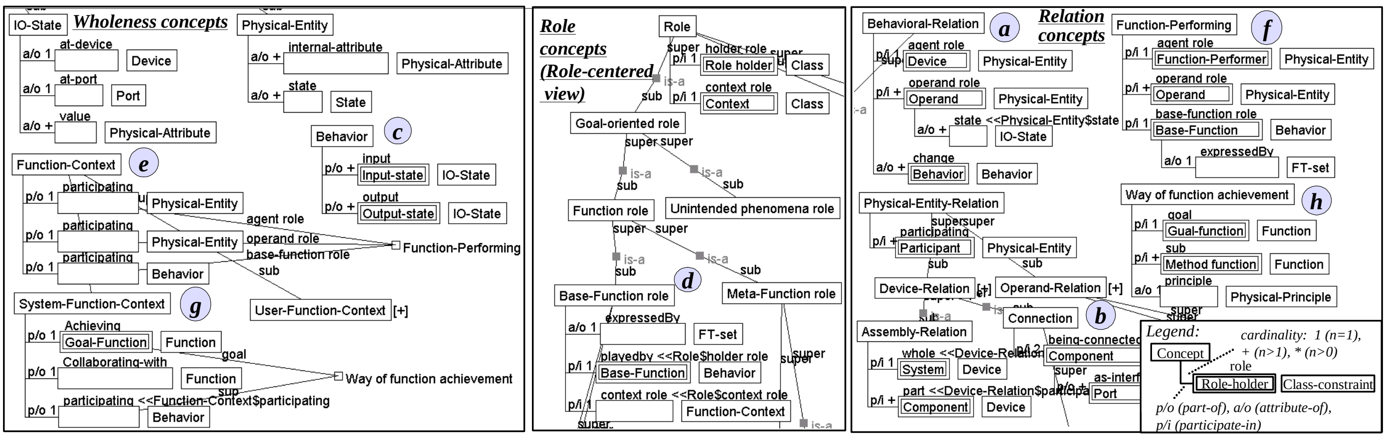


Figure 2. A Portion of an Ontology of Device-centered Behavior and Function in Hozo.

inherited from the super-concept plus the following three axioms; (1) P-Focus on an inlet port and an outlet port, (2) Energy in the focused outlet port is made from energy in the focused inlet port, and (3) Mediums of the focused energies are different. *To take*, a subtype of *to shift* in the *is-a* hierarchy, is defined as FTs of *to shift* with an additional FT, P-Focus on the port of energy provider. Likewise, *to remove* is defined as that of *to take* with an additional FT, the energy taken is *unnecessary* as *Necessity* FT.

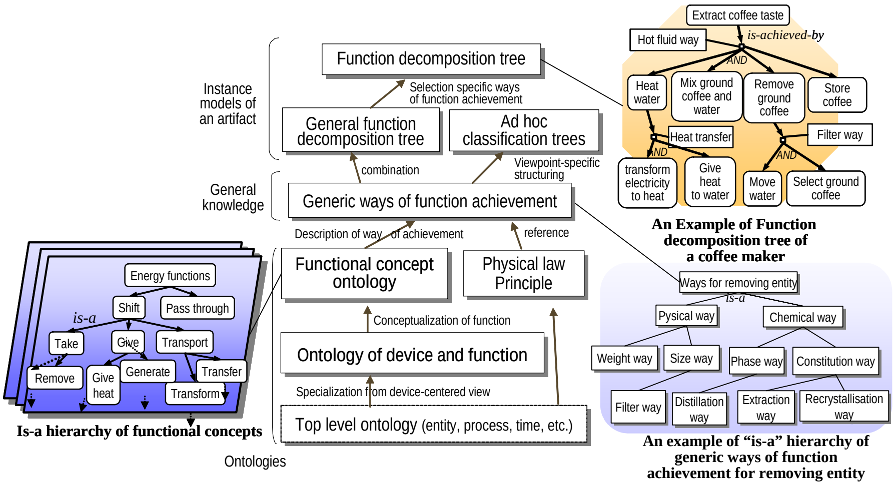


Figure 3. Layered ontologies and functional knowledge.

3.3 Function decomposition tree

On the basis of the ontologies, a *function decomposition tree* is described as a functional model of a concrete artifact. According to the extended device ontology, the knowledge authors assign *roles* to physical entities in the target world. Functions of the function decomposition tree are instances of generic functions defined in the functional concept ontology. In the example of the coffee maker in Figure 3, the goal function of the coffee maker is "to extract coffee taste". The whole function is decomposed into finer functions, that is, "to heat water", "to mix ground coffee and water", "to remove ground coffee" etc. Its way of function achievement is conceptualized as "hot fluid way", which principle is extraction by heat.

A *general function decomposition tree* includes possible alternative ways of function achievement in OR relationship for a specific goal function. It shows *design alternatives* which are possible to be adapted or have been rejected in the previous design. It plays a crucial role in sharing design rationale.

3.4 Generic way of function achievement

The concrete *ways of function achievement* in function decomposition trees can be generalized into a generic *way of function achievement* (called functional way knowledge). Then, *ways* to achieve the same function are organized in *is-a* relations according to their principles (called an *is-a hierarchy of ways of function achievement*). We distinguish the organization as an *is-a* hierarchy from the other derivative organizations depending on viewpoints (called an *ad hoc classification tree*). The concept of "way of function achievement" and this distinction are solutions for the issue in Section 2.4. Figure 3 shows a portion of the "is-a" hierarchy of generic ways of function achievement for removing entity. They are categorized into "physical ways" and "chemical ways", which are categorized into more specialized sub-classes. The filter way used in the coffee maker is sub-class of the "size way" of the "physical way".

4. Deployment in Industry [32]

The ontology and the modeling framework for functional knowledge have been deployed for over four years at the Plant and Production Systems Engineering Division of Sumitomo Electric Industries, Ltd. (hereinafter referred to as SEI). The purpose was to share functional knowledge among engineers about the production facilities used in their daily work.

After a one-year study of our framework, test use was commenced in February 2001. In May, 2001, use on real problems encountered in daily work was started. The targets are manufacturing facilities that are mainly used in semiconductor manufacturing processes including machines to slice semiconductor ingots, machines to polish wafers, a tension control system, and inspection machines.

A knowledge management software named SOFAST® (abbreviation for Sumitomo Osaka-university Function Analysis and Systematization Tool and registered trademark of SEI.) has been deployed since December, 2002. It is designed to support the description of the functional decomposition trees and sharing in an intra-network. Using its client software, a user can describe function decomposition trees through a graphical user-interface and store them in its knowledge repository. Then, all users can search *ways of function achievement* in the repository to achieve the function of interest by specifying a goal function. A users' group of SOFAST software for companies was established in April, 2003. There are currently 13 member companies where test use has been done.

4.1 Successful Usage

One of the use of the function decomposition tree is to clarify functional knowledge, which is implicitly possessed by each engineer, and share it with other engineers. The experiential evaluation by Sumitomo's engineers was unanimously positive. Such sharing was done for design-review where a design team do check a design and explore possible alternatives. As a replacement of a comparative table with a text, the engineers started to use the general function decomposition trees for design-review documents. After adopting our framework, the number of times the design reviews had to be done was successfully reduced to one third. This is because it shows alternative *ways* of achieving functions for each (sub) function, their features in comparison, and reasons for adopting a specific way, or not, in the one figure.

Writing a function decomposition tree according to the methodology gives designers the chance to reflect on good stimuli, which leads them to an in-depth understanding of the equipment. Such a deep understanding of the equipment contributes to redesigning and solving problems with it. For example, an engineer was not able to reduce the time a machine requires to polish semiconductor wafers after four months of investigation by adjusting the known working parameters. He consequently described its function decomposition tree shown in Figure 4(a). He understood the guide ring has only one function, that is, to guide the movement of the rotating disk. The rotating disk freely moves inside the guide ring for polishing the wafer. Thus, he was not aware of another implicit function, "to place diamond power between wafers and the table" (i.e., the function marked with dotted circle was missing). Then, he found it when he referred to that of the wire-saw shown in Figure 4(b). Although these two devices have different main functions and the wire-saw knowledge has been described by another engineer, he found the shared function "to maintain a large friction coefficient" (marked with circles in Figures 4(a) and 4(b)) and its sub-function "to place grind compound" in the wire-saw model. As a result, he became aware of the missing function in the polisher model and its parameters for placing more diamond powder (that is, the width of the guide ring) to obtain a high friction coefficient. Eventually, he reduced the necessary time to 76%, which was better than the initial goal. This improvement was achieved within three weeks. This example shows that functional knowledge tends to be implicit and awareness of such functionality gives engineers good stimuli for improvement. Moreover, reuse of functional knowledge between different facilities contributes to it.

The general function decomposition tree can be used to compare design candidates by explicating different ways to achieve functions. It contributes to patent analysis and patent applications. In communications between engineers and patent attorneys in applying for a new patent, it is difficult to determine the product's originality and to make appropriate claims. When the general function decomposition tree has been adopted as regular document format for patent application, the period was reduced to just one week from three or four weeks. The patent claims were increased and doubled in some cases, since the attorneys found extra differences with other patents by checking at each level of function decomposition. The same benefit was found by another company in the users' group.

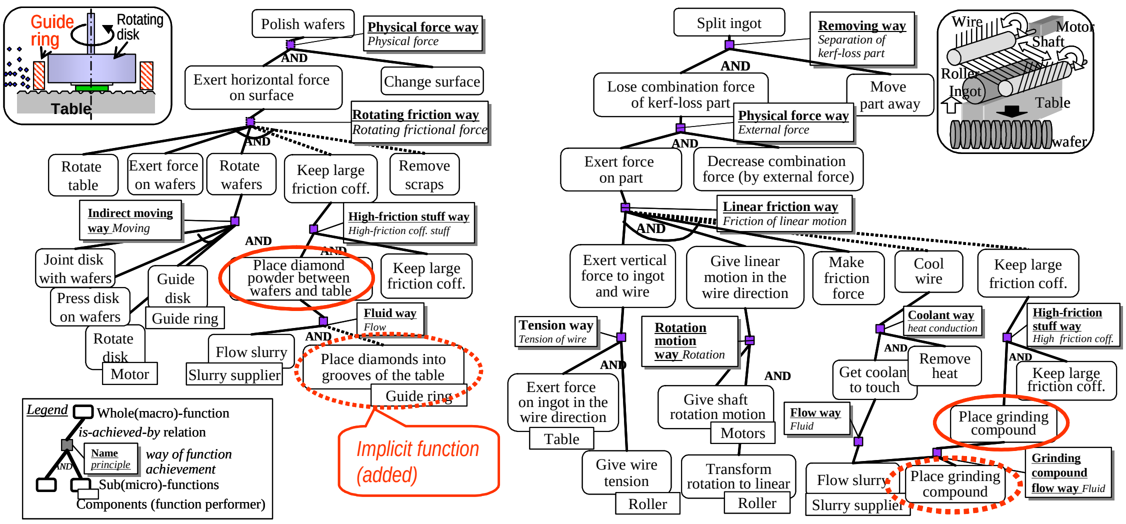
Generic knowledge about ways of function achievement help designers search ways to achieve a function and/or alternatives in an existing product. In the deployment, a novice engineer developed an inspection machine in three days by systematically consulting generic ways of shedding light in the knowledge repository of SOFAST. Such development usually requires experts two weeks.

4.2 Discussion: Success Factors for Deployment

The successful deployment discussed thus far is a kind of knowledge management activity. In general, difficulties with knowledge management activities include:

- Difficulty in explicating implicit knowledge,
- Difficulty in retrieving useful knowledge, and
- Lack of motivation in writing own knowledge.

First, it is rather difficult to explicate one's own implicit knowledge. Functional knowledge is intrinsically subjective rather than objective. Without guideline, novice modelers would be puzzled in



Left: Figure 4(a). Function decomposition tree of a polisher which is described for its improvement
Right: Figure 4(b). Function decomposition tree of a wire-saw which is referred to.

describing functional models. Especially, the extended device ontology provides users with hints on interpreting how a device works consistently. It provides concepts for assigning "roles" for each object in the target world. For the wire-saw example of difficulty discussed in Section 2.3, the wire can be considered as an *agent* (exerting force on ingots), an *operand* (moved by the roller) or a *conduit* (transmitting tension) depending on its location. According to semantic constraints in the extended device ontology, a possible way to consistently assign roles is to decompose the wire into two parts, a working wire playing an *agent role* and a transmitting wire playing both a *medium* role and a *conduit* role.

The concept of "way of function achievement" also helps modelers to eliminate the confusion between "what to achieve" and "how to achieve it". A clear distinction between a general-specific hierarchy (*is-a* relations) and a whole-part hierarchy (*is-achieved-by* relations) helps knowledge authors to have consistent descriptions of function decomposition trees and *is-a* hierarchies of *ways of function achievement*. This avoids the confusion between the two, which has often occurred as discussed in Introduction and Section 2.4. For example, the example of "to weld" in Introduction can be described as *fusion way* of the *joining function*. The fusion way has specific characteristics of the output that the operands are fused and they are hard to be separated. Although a functional concept "to join" loses some amount of information of "to weld", what is loses goes to the characteristics of the fusion way. As a total, functional concepts are successfully made very generic without any loss of information.

Secondly, one of the main reasons of the difficulty in retrieving functional knowledge is that many product data have too much detail to recognize "commonality" between the products. Thus, engineers have difficulty to find related knowledge in different domains. It requires appropriate abstraction of the product knowledge. The concept of function itself is a solution, since a function is intrinsically abstract and independent of its realization (e.g., structure and material). Nevertheless, many functions in practice imply the way of function achievement such as "welding", which is a kind of dependence on realization. It prevents to recognize the commonality between artifacts. In our framework, on the basis of the concept of the way of function achievement, we established an ontology of functional concepts, which enables the computer systems to find more artifacts to match with a specific artifact.

Lastly, in general, knowledge authors have no effective motivation to write their own knowledge and share it with others. In the SEI's deployment, however, engineers themselves say that they obtain benefits from writing functional models of their own equipment, since it gives them the chance to reflect and obtain good stimuli, which leads them to an in-depth understanding of the equipment. This is enabled by our modeling framework operating as a knowledge medium, which externalizes the engineers' understanding, which had been implicit, to an appropriate level of abstraction with consistent guidance. In general, the micro-macro hierarchy of the function decomposition tree enables the designer to systematically explore possible alternatives (for conceptual design) and/or causes of the problem (for problem solving) for each function. For example, fault tree analysis (FTA) for trouble shooting tends to make it difficult to enumerate all possible causes without a clear understanding of the function structure of the target device.

4.3 Difficulties in the Deployment and Solutions

The main point of our framework is that it adopts an ontological approach to controlling the content of functional models. However, we faced some difficulties in the deployment. In this section, we discuss the difficulties and their solutions including future work.

The first one is less freedom of functional representation than ad hoc functional modeling. It became a problem especially in selecting a functional concept for a component. There could be a domain-specific vocabulary and different terms for the same concept. We cannot claim completeness of concepts in our functional concept ontology due to its nature and understand the necessity of extending it.

In order to reduce this difficulty and to make the use of SOFAST easier, the latest version supports multi-level terms which consist of the functional concepts (defined in the ontology), *usual functional words* and domain-specific words. The *usual function words* are verbs for representing functions appearing in daily work, and have been prepared beforehand by collaboration with companies in the users' group. Such words are associated with functional concepts. SOFAST of the latest version thus allows knowledge authors to use terms rather freely.

The second difficulty of our ontological approach is that it needs training for writing functional models that are compliant with functional ontologies. In other words, it is not very easy to impose ontological commitments on knowledge authors. In the deployment, the SOFAST users' group held many practices for writing functional models using real examples submitted from member-companies.

The authors are currently establishing stepwise guidelines for describing functional knowledge to help easier commitment to the ontologies [33]. Moreover, automatic checking of violations in the functional models based on the ontologies is being investigated. Their definitions are structured with slots and constraints and include axioms as a result of deep insights into the behaviors and functions in physical systems. Such formal definitions can be used to automatically check the models.

The third difficulty is the writing cost for the detailed model of the target artifact. In the deployment, the function decomposition is adopted as an official standard format for design review in stead of conventional documents in a table form. Thus the modeling cost is not additional but rather reductive. Nevertheless, there is still difficulty to describe such detailed model.

The authors and colleagues are currently investigating more simplified functional models as meta-data for web documents [45]. In this framework, a knowledge author can choose the level of description, from just a keyword representing the whole function of the device to a full functional model as discussed in this paper. Such functional knowledge associated with web documents in natural language will reduce the cost of modeling according to company's demand.

As a result of these difficulties, some of the functional models described thus far do not follow the functional ontologies completely in the deployment. The current vocabulary used in SOFAST is not fully based on the functional concept ontology. Sharing *ways of function achievement* in SOFAST does not rely on the *is-a* hierarchy of generic *ways* but on searching for specific *ways* of function achievement by specifying the goal function. Nevertheless, discrimination of *is-a* relations from *ways of function achievement* has helped engineers avoid a great deal of confusion.

5. Discussion

5.1. Other kinds of function

Our ontology of device and function discussed in Section 3 defines the concept of functionality strictly from the device-centered viewpoint. Such strict definition is done intended to prescribe guidelines to functional modeling. Other types of function, however, still remain to be investigated. In order to clarify the limitation of discussion in this paper and to place our definition of function in the map of functions, this section discusses rather descriptive definitions of other kinds of function as shown in Figure 5. They represent viewpoints (or context) for human's perception of a function. Thus, a device can achieve some functions in different categories simultaneously. Note that Figure 5 shows an "is-a" hierarchy only for readability, because some distinctions are independent from each other.

Firstly, the function discussed in Section 3 represents changes of entities (behaviors) within the system boundary (here we call *device function*). On the other hand, an *environmental function* includes changes outside of the system boundary, especially, those related to *users* or *user actions*. For example, an electric fan performs moving-air function as a *device function* and cooling function for human body as an *environmental function*, where the cool-down effect by wind is on human body and thus outside of the system boundary. This cooling *environmental function* means physical changes of the system (called *physical environmental function*), while an *interpretational function* sets up one of necessary conditions of human's cognitive interpretation. For example, a clock has "to rotate hands (in the specific and constant rate)" as a *device function* and "to inform time" as *an interpretational function*, which requires human's cognitive interpretation.

Chandrasekaran and Josephson discuss a similar kind of function called *environment function* as effect on environment (the surrounding world of the device) [11]. They conceptualize "mode of deployment", which is relationship (configuration) of device and environment. It is similar to the *function-context* in our ontology, but it mainly represents functional configuration. Some researchers distinguish *purpose* from function [7],[8], where the purpose represents human-intended goal in the similar sense to this *environmental function* or *interpretational function*. Hubuka distinguishes the purpose function as effects from the technical function as internal structure [15]. Rosenman and Gero investigate *purpose* in *socio-cultural environment* [46]. The situated FBS framework treats change of requirements [12]. In our collaborative work with the Delft University of Technology, we are extending our framework to include user actions as well [47]. The affordance-based design has been investigated [49].

Secondly, the *base-function* discussed in Section 3 refers to temporal changes of physical attributes of objects which flow through the device (called *flowing object* function here). It can be generalized into *effect-on-state function* which means temporal changes of physical attributes. The effect-on-state function has another kind, that is, *inter-device* function which refers to changes of another device (called B-3 behavior in [31]). Its example is a rod's function "to push cam". The cam is another device, which is not considered as objects flowing through the rod.

On the other hand, the *effect-on-process function* represents effect on a process or its changes. Behavior as basis of function can be regarded as a kind of a process. Thus, as a subtype of the effect-on-process

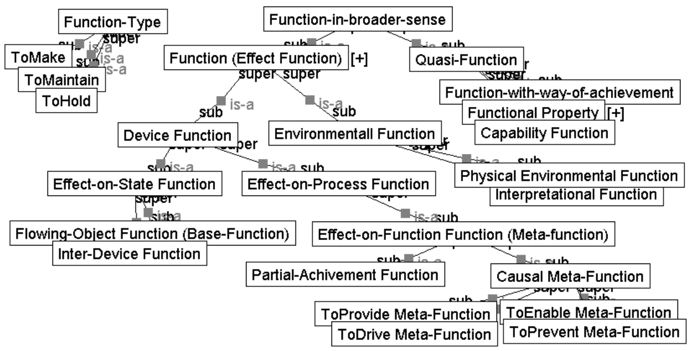


Figure 5. Descriptive Categories of Function

function, *effect-on-function function* (we can call meta-function) represents a role of a function for another function. It includes *partial-achievement function* and *causal-meta function*. The former is performed by a *method function* for a *goal function* in the *is-achieved-by* relation. The latter represents a role for another method function and is called a *meta-function* in Section 3.1 and [44].

Thirdly, the function-types discussed in Section 3.1 are additional descriptors for the functions discussed thus far.

Fourthly, we recognize the following three kinds of *quasi-functions*. Although the authors do not consider them as kinds of function, it is found that a quasi-function is confused with a function. Firstly, a *function-with-way-of-achievement* implies a specific way of function achievement as well as a function. Its examples include washing, shearing, adhering (e.g., glue adheres A to B) as well as welding mentioned in Introduction. Because meaning of this type of function is impure, we regard this quasi-function. Secondly, a *functional property*[2] represents that an artifact (usually material) has a specific attribute-value which directly causes functionality. This is found in material science domain where a material whose function is dependent on its electronic, optical or magnetic property is called functional material [50]. For example, if an electrical conductivity of a material is high (i.e., it has high conductivity property), the material can perform the "to transmit electricity" function. There is direct relationship between the high-conductivity property and the transmitting function. Lastly, a *capability function* represents that an entity can perform an activity which is not effect on others. For example, people say that "a human has walking function".

5.2. Limitation of Application Domains

Besides the production systems and facilities in the deployment, the ontologies have been applied to modelling a power plant [44], an oil refinery plant, a chemical plant, a washing machine, a printing device, and manufacturing processes. The models have taken into account changes in thermal energy, flow rate, and ingredients of fluids, including force and motion of operands. The current functional concept ontology can describe simple mechanical products, although it does not cover static force balancing and complex mechanical phenomena based on the shape of operands. The modelling framework currently cannot cope with the human's mental process, body movements (so-called therblig in Industrial Engineering), business processes, or software processes.

6. Related work

We have discussed definitions of function in the literature in Section 2 and 5.1 and ontological work in Introduction and Section 2. Thus, we here give the remaining notes on related work.

Our domain ontology is different from "task" knowledge of designing or diagnosing, which is activity of human or automated problem-solvers. If one ignores the difference between domain and task, the generic tasks and the generic methods (PSMs) in the task ontology research (e.g.,[51]) are similar to our generic functions and generic ways of function achievement, respectively. We focus on structuring knowledge about how to achieve functions (activities in domain world). We conceptualize the principle behind the sequence of activities (called method in both researches) as the way of function achievement. It helps us organize them in *is-a* hierarchies, though PSMs for a specific task are usually not organized well. Moreover, we distinguish function at the teleological level from behaviors at the objective level.

Behavior of artifacts is a kind of "process" by which we intuitively mean a sequence of state changes over time. We concentrate on physical process which represents temporal changes of physical quantities as we discussed in Section 2 and 5. On generic "process", extensive research has been done such as the process specification language (PSL) [52] for "activity" and its temporal relationships, formal ontologies for processes (e.g., [35]), and the MIT process handbook for business activities [53] which taxonomy for business activities includes "activity-with-way" such as "buy in a store" like "welding" in Introduction.

Similarly to the *way of function achievement*, a feature of function decomposition can also be found as a "means" in [14],[16],[18]. We defined *is-a* relations between conceptualized generic *ways of function achievement*, and investigated how to organize them.

The generic way of function achievement aims at its generality by pointing out partial (and abstract) information on structure and behavior. The similar knowledge in the literature, the design prototypes [5] and the FBS modeling framework [9], include structural decomposition and physical features, respectively. In IDEAL [10], generic teleological mechanisms (GTM) are used (modified) to design different contexts based on analogies. In our approach, based on a limited set of functional concepts, designers can explore explicit *is-a* hierarchies of *ways of function achievement*.

TRIZ (TIPS) theory provides some patterns (or strategies) for inventions based on the contradiction between two physical quantities [54]. We did not

[2] The term "functional" here is intended to represent neither mathematical dependence relation nor attributes of function but function-*oriented* property, though attributes of function include (not only) functional properties. Functional property is used as an antonym of mechanical or structural property.

concentrate on design strategies but on modelling schema. TRIZ theory also concentrates on physical principles (effects), although we established a clear relationship between physical principles and functional structures.

7. Concluding remarks

Ontological consideration on functionality of artifact and a successful deployment of an ontological modeling schema have been discussed. The role of ontologies is to provide semantic constraints to capture the target world consistently and controlled vocabulary for representation. We are currently extending the ontological schema for unintended phenomena [48] and user actions [47] and using it as metadata schema for web-documents [45].

Acknowledgements: The authors would like to thank Kouji Kozaki, Munehiko Sasajima, Eiichi Sunagawa, Sninya Tarumi, Naoya Washio for their contributions to this work. Special thanks go to Dr. Masayoshi Fuse, Mr. Masakazu Kashiwase, Mr. Shuji Shinoki and the engineers in the Plant and Production Systems Engineering Division of Sumitomo Electric Industries, Ltd. for their cooperation with the deployment. The authors are grateful also to an anonymous reviewer for valuable comments.

Reference

[1] Lee, J. Design rationale systems: understanding the issues. IEEE Expert, 12(3):78-85., 1997.

[2] Chandrasekaran B, Goel AK, and Iwasaki Y. Functional representation as design rationale. Computer 1993:48-56.

[3] De Kleer J. How Circuits Work. Artificial Intelligence 1984:24;205-80.

[4] Sembugamoorthy V, Chandrasekaran B. Functional representation of devices and compilation of diagnostic problem-solving systems. In Experience, memory and Reasoning 1986:47-73.

[5] Gero JS. Design Prototypes: A Knowledge Representation Schema for Design. AI Magazine. 1990;11(4):26-36.

[6] Keuneke AM. A. Device Representation: the Significance of Functional Knowledge. IEEE Expert 1991;24:22-25.

[7] Chittaro L, Guida G, Tasso C, Toppano E. Functional and Teleological Knowledge in the Multi-Modeling Approach for Reasoning about Physical Systems: A Case Study in Diagnosis. IEEE Transactions on Systems, Man, and Cybernetics 1993;23(6):1718-1751.

[8] Lind M. Modeling Goals and Functions of Complex Industrial Plants. Applied artificial intelligence 1994;8:259-283.

[9] Umeda Y, Ishii M, Yoshioka M, Shimomura Y, Tomiyama T. Supporting conceptual design based on the function-behavior-state modeler. Artificial Intelligence for Engineering Design, Analysis and Manufacturing 1996;10:275-288.

[10] Bhatta SR, Goel AK. A Functional Theory of Design Patterns. In Proc. of IJCAI-97 1997:294-300.

[11] Chandrasekaran B, Josephson JR. Function in Device Representation, Engineering with Computers 2000;16(3/4):162-177.

[12] Gero JS, Kannengiesser U. The Situated Function-Behaviour-Structure Framework. In Proc. of Artificial Intelligence in Design ’02 2002;89-104.

[13] Pahl G, Beitz W. Engineering design - a systematic approach. The Design Council; 1988.

[14] Malmqvist J. Improved function-means trees by inclusion of design history information. Journal of Engineering Design 1997;8(2):107-117.

[15] Hubka V, Eder WE. Theory of Technical Systems. Berlin: Springer-Verlag; 1998.

[16] Bracewell RH, Wallace KM. Designing a Representation to Support Function-Means based Synthesis of Mechanical Design Solutions. In Proc of ICED 01. 2001.

[17] Rajan JR, Stone RB, Wood KL. Functional Modeling of Control Systems. In Proc. of ICED 03 2003.

[18] Wilhelms S. A Conceptual Design Support System using Principle Solution Elements. Proc. of ICED 03 2003.

[19] Hirtz J, Stone RB, McAdams DA, Szykman S, Wood KL. A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts. Research in Engineering Design 2002;13:65-82.

[20] Miles LD. Techniques of value analysis and engineering. McGraw-hill; 1961.

[21] Umeda Y, Tomiyama T. Functional Reasoning in Design, IEEE Exert 1997:March/April:42-8.

[22] Hubka V, Eder WE. Functions Revisited. In Proc. of ICED 01 2001.

[23] Deng YM. Function and Behavior representation in conceptual mechanical design. Artificial Intelligence for Engineering Design, Analysis and Manufacturing 2002;16:343-362.

[24] Gruber TR, Tenenbaum JM, Weber JC. Toward a Knowledge Medium for Collaborative Product Development. In Proc. of Artificial Intelligence in Design ’92 1992;413-432.

[25] Cutkosky MR, et al. PACT: An Experiment in Integrating Concurrent Engineering Systems. Computer 1993;January:28-37.

[26] Gruber T, Olsen G. Theory Component-Assemblies, Ontology Server. http://www-ksl.standford.edu. 1994.

[27] Borst P, Akkermans H, Top J. Engineering Ontologies. Int’l Journal of Human-Computer Studies 1997;46(2/3):365-406.

[28] Sekiya T, Tsumaya A, Tomiyama T. Classification of Knowledge for Generating Engineering Models - A case study of model generation in finite element analysis -. In; Finger S, Tomiyama T, Mäntylä, editors. Knowledge Intensive Computer Aided Design, Boston: Kluwer Academic Publishers; 1999, p. 73-90.

[29] López, M. F., et al..: Building a Chemical Ontology using Methontology and the Ontology Design Environment, IEEE Intelligent Systems, Jan./Feb., pp.37-46 (1999)

[30] Kitamura Y, Mizoguchi R. Ontology-based description of functional design knowledge and its use in a functional way server. Expert Systems with Application 2003;24(2);153-166.

[31] Kitamura Y, Mizoguchi R. Ontology-based systematization of functional knowledge. Journal of Engineering Design 2004:15(4): 327-351.

[32] Kitamura, Y., Kashiwase, M. Fuse, M., Mizoguchi, R., Deployment of an Ontological Framework of Functional Design Knowledge, Advanced Engineering Informatics, 2004:18(2):115-127.

[33] Kitamura, Y., Mizoguchi, R., Ontology-based Functional-Knowledge Modeling Methodology and its Deployment, Proc. of EKAW 2004, 99-115, 2004.

[34] De Kleer J, Brown JS. A Qualitative Physics based on Confluences. Artificial Intelligence 1984;24:7-83.
[35] Sowa, J. F.: Top-level ontological categories, International Journal of Human-Computer Studies, Vol. 43, No. 5-6, pp. 669–685 (1995)
[36] Guarino Masolo, C., Vieu, L., Bottazzi, E., Catenacci, C., Ferrario, R., Gengami, A., and Guarino, N.: Social Roles and their Descriptions, in Proc. of the 9th Int'l Conf. on the Principles of Knowledge Representation and Reasoning (KR2004), 267–277 (2004)
[37] Mizoguchi, R., et al. Construction and Deployment of a Plant Ontology, Proc. of EKAW 2000, 113-128, 2000.
[38] Kozaki, K., Kitamura, Y., Ikeda, M., Mizoguchi, R., Hozo: An Environment for Building/Using Ontologies Based on a Fundamental Consideration of "Role" and "Relationship", Proc. of EKAW2002, 213-218, 2002
[39] Sunagawa, E., Kozaki, K., Kitamura, Y., Mizoguchi, R., Organizing Role-concepts in Ontology Development Environment: Hozo, AI-TR-04-1, I.S.I.R., Osaka University, 2004.
[40] Breuker, J., Hoekstra, R., Epistemology and ontology in core ontologies: FOLaw and LRI-Core, tow more ontologies for law, EKAW 2004 Workshop on Core Ontologies in Ontology Engineering, pp. 15-27, 2004.
[41] West, M., Some Industrial Experiences in the Development and Use of Ontologies, EKAW 2004 Workshop on Core Ontologies in Ontology Engineering, pp. 1-14, 2004.
[42] Fan, J., Barker, K., Porter, B., and Clark, P.: Representing Roles and Purpose, in Proc. of the international conference on Knowledge capture (K-Cap2001), pp. 38–43 (2001)
[43] Sasajima M, Kitamura Y, Ikeda M, Mizoguchi R. FBRL: A Function and Behavior Representation Language. Proc. of IJCAI-95 1995;1830–6.
[44] Kitamura Y, Sano T, Namba K, Mizoguchi R. A Functional Concept Ontology and Its Application to Automatic Identification of Functional Structures. Advanced Engineering Informatics 2002;16(2):145-163.
[45] Kitamura, Y., Mizoguchi, R., An Ontological Schema for Sharing Conceptual Engineering Knowledge, Proc. of the International Workshop on Semantic Web Foundations and Application Technologies, pp. 25-28, 2003.
[46] Rosenman, M.A., Gero, J. S., Purpose and function in design: from the socio-cultural to the techno-physical, Design Studies, 19, 161-186, 1998.
[47] van der Vegte WF, Kitamura Y, Koji, Y., Mizoguchi R, Coping with Unintended Behavior of Users and Products: Ontological Modeling of Product Functionality and Use, In Proc. of CIE 2004: ASME 2004 Design Engineering Technical Conferences and Computers in Engineering Conference 2004. DETC2004-57720.
[48] Koji Y, Kitamura Y, Mizoguchi R. Towards modeling design rationale of supplementary functions in conceptual design. In Proc. of Fifth International Symposium on Tools and Methods of Competitive Engineering 2003; 117-130.
[49] Maier, J. R. A., Fadel, G. M., Affordance-based methods for Design, Proc. of DETC 03, DTM-48673, 2003.
[50] Engineering and Physical Sciences Research Council, http://www.epsrc.ac.uk/ResearchFunding/Programmes/Materials/ResearchPortfolio/EngineeringFunctionalMaterials.htm
[51] Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., Van de Velde, W. and Wielinga, B.: Knowledge Engineering and Management - The Common-KADS Methodology, The MIT Press, Cambridge, MA (2000)
[52] ISO TC184/SC4/JWG8, Process Specification Language, Part 1, 11 and 12, http://www.tc184-sc4.org/SC4_Open/SC4_Work_Products_Documents/PSL_(18629)/ (2003)
[53] Herman, G. A., Malone, T. W.: What is in the process handbook?, Organizing Business Knowledge: The MIT Process Handbook, MIT Press (2003) 221-258
[54] Sushkov VV, Mars NJI, Wognum PM. Introduction to TIPS: a Theory for Creative Design. Artificial Intelligence in Engineering 1995;9.