INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITE DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

TOWARDS INTEROPERABILITY BETWEEN FUNCTIONAL TAXONOMIES USING AN ONTOLOGY-BASED MAPPING

Masanori Ookubo[1], Yusuke Koji[1], Munehiko Sasajima[1], Yoshinobu Kitamura[1] and Riichiro Mizoguchi[1]

[1]The Institute of Scientific and Industrial Research, Osaka University

ABSTRACT
A functional model of an artifact shows an important part of designer's intention. A taxonomy of generic functions plays a crucial role in describing such functional models. Improving interoperability among functional models of different functional taxonomies facilitates sharing functional models among designers in organizations. The mapping between the different functional taxonomies and the conversion of functional models based on this mapping are one of the ways to support such interoperability. Among the different functional taxonomies, however, there are not only terminological differences but also ontological differences implicitly. By ontological differences we mean the conceptual differences between taxonomies and the general structural differences between models. In this article, firstly we propose a framework for mapping between functional taxonomies using a *reference ontology of function* in order to bridge the conceptual differences. The reference ontology contributes to specify the ontological differences between the functional taxonomies. Second, in order to reduce the structural differences between the functional models, we discuss how to capture the modeling world. These results enable us to convert the functional models among different taxonomies. As examples of the functional taxonomies, we focus on the reconciled functional basis proposed by Hirtz et al. and our functional concept ontology based on Function & Behavior Representation Language.

Keywords: Knowledge management/sharing, Functional analysis, Ontology

1 INTRODUCTION
Functionality plays a crucial role in the conceptual design of engineering products. The knowledge on functionality of a product (called *functional model*) shows an important part of designer's intention and thus its sharing among engineers in organizations promotes the collaborative design. Much research on functionality has been conducted in areas such as functional representation (e.g., [1]-[5]), engineering design (e.g., [6]-[18]), and value engineering [19], though there is no common definition of functions [2][4][11][16][18] . In practical situations, engineers tend to describe functional models in an ad hoc way. Functions can be captured in different ways in different domains. Thus, the reusability and interoperability of functional models in organization(s) are required. Thus, this paper focuses on sharing of design knowledge of product in organizations in society, which is directly related to the theme of this conference.
One of the approaches to this is to use taxonomy of functions, whereby functions of components in functional models are described in terms of a taxonomy. In fact, some taxonomies of generic functions have been proposed in [9][13][15][20]-[23]. When different taxonomies are used in functional models, the interoperability among them becomes problematic. Such an interoperability problem can be solved by *mappings* terms in the different taxonomies. Between the taxonomies, however, there are not only terminological differences but also implicit ontological differences. For example, as terminological mappings between the reconciled functional basis (hereafter FB) [9] and our functional concept ontology [21] (hereafter FBRL), the term "couple" in FB can be mapped to "combine" in FBRL. On the other hand, the term "link" in FB implies not only "to couple flows together" [9] as the change at input and output but also "by means of an intermediary flow" [9] as how to realize it. Thus, it cannot be fully mapped to "combine" in FBRL which implies "to bring two operands into an operand" as the

change at input and output, which corresponds to only the former part of the meaning of “link”. This is not a terminological but an ontological difference, because “the change in the target object” and “how to realize the change” are ontologically different.
In order to reveal such ontological differences, our approach is to use a *reference ontology of function*, which consists of several upper classes (categories) of functions. A function term in a functional taxonomy is (ideally) categorized into an upper class in the reference ontology. The term can be mapped to other term(s) in other functional taxonomy according to the categories in the reference ontology which the terms are categorized to. As a reference ontology of function, we use a modified version of our “descriptive categories of function” [24][25]. It has been developed based on ontological and empirical investigation on various functions in engineering practice [25].
This article proposes a generic framework of mapping between functional taxonomies on the basis of a reference ontology of function. We discuss some general relationships among the reference ontology, target functional taxonomies and functional models. We categorize types of mappings between functional terms in functional taxonomies according to their categorized classes in the reference ontology. Such mappings of terms can be used for conversion from a functional model based on a functional taxonomy into a model based on the other taxonomy. The conversion of a functional model consists of translation of functional terms using the mappings of terms and transformation of model structures.
In this article, as an example of the target taxonomies to be mapped, we use the reconciled functional basis (FB) [9] and our functional concept ontology based on Function & Behavior Representation Language (FBRL) [21][26]. We show some examples of mapping knowledge between them. The complete mapping knowledge and algorithm of the model conversion is under investigation.
Section 2 discusses related work. Section 3 proposes a framework of the reference ontology-based mapping between functional taxonomies. We give an overview of the elements of the framework. Section 4 gives an overview of the two taxonomies; the reconciled functional basis and the functional concept ontology. Then, in Section 5, we discuss some patterns of the ontology-based mapping between the taxonomies with examples of the mappings between FB and FBRL. Section 6 discusses structural differences between the functional models. Section 7 shows an example of possible conversion.

2 RELATED WORK AND OUR AIMS

As mentioned in Inroduction, there are many definitions (see [2][4][11][16][18]) and taxonomies of functions of artifacts [9][13][15][20]-[23]. Among others, the reconciled functional basis is a result of merging two existing taxonomies aiming at a ‘standardized taxonomy’ [9]. We aim at mapping rather than merging, in order to allow the diversity of conceptualization of functions. Thus, the reference ontology of function to be used in this paper provides not a super-set (logical sum) of the existing taxonomies but generic upper categories of functions. Especially, as the application in the semantic web, treatment of interoperability among diverse definitions is important. For example, information integration of product data in Semantic Web is required and realized by ontology mapping [27]. A functional annotation schema for the Semantic Web proposed in [28] based on the functional basis as taxonomy, however, does not cope with such interoperability. We have proposed an annotation system about functions in technical documents [29] based on our ontology. The framework we propose in this paper is intended to be used as a basis for extension of the annotation system as well.
The framework we propose in this paper adopts layered structures of ontologies, which are commonly used in Ontological Engineering such as PhysSys [30]. The FBSO ontology [31] also adopts similar layered structure such as generic, application and domain layers. We focus on the mapping of ontologies rather than a unified set of ontologies. An ontology at the top-level layer is called “top-level ontology” or “upper ontology”, which includes meta, abstract, and philosophical concepts such as DOLCE [32] and SUMO [33]. PSL also treats general (discrete) “process” such as manufacturing process [34]. Our reference ontology of function in the framework proposed in this paper is at the middle-level lower than those upper-levels and includes several function concepts that are general in engineering domain [25]. An analysis of some upper-level ontologies has been done in order to determine which is most appropriate for the manufacturing domain (As a result, Cyc was selected) [35]. A methodology for creating engineering ontologies reported in [36] includes reuse of existing taxonomies, test for application, and refinement of the integrated taxonomy.

The ontology-based integration and interoperability among design knowledge have been investigated from early 1990's such as PACT [37] and KIEF [38]. They mainly focus on generic interoperable mechanism among agents and/or engineering tools. Product data exchange based on a generic ontology has been proposed in [39]. It aims mainly at exchange of product data such as geometry between software systems rather than conceptual knowledge of functional knowledge in our research. We aim at generic and richer ontology of function and clear conceptualization of related types. We investigate ontology mapping for integration between functional knowledge and knowledge about faults as well [40].

In the research field of ontological engineering, much research has been carried out on automatic 'mapping discovery' which is to find similarities between two ontologies and determine which concepts (and properties) represent similar notions [41]. Such techniques use lexical information based on natural language processing techniques, semantics of relationships in (structural features of) ontologies, and/or a set of shared instances of classes. It is pointed out that such (semi-) automatic finding can be facilitated by a common grounding such as a shared upper ontology [41]. Our reference ontology, we think, can have the same effect on the automatic mapping discovery for functional taxonomies(cf. Section 3).

3 OVERVIEW OF THE MAPPING FRAMEWORK

Figure 1 shows the overview of the proposed framework for interoperability between functional taxonomies. A functional taxonomy defines general types of functions as classes (which we call *function classes* or *functional terms*). A function class is a conceptualization of a specific set of functions of products or components. Typically, the classes are categorized in a hierarchical way. We assume that the hierarchy represents logical subsumption relations. Based on a functional taxonomy fx_1, a functional model fm_1 is described. A function of a component (or a (sub-)system) in fm_1 is an instance of one of the classes defined in fx_1.

A function class in a functional taxonomy fx_1 is categorized into an upper class defined in *a reference ontology of function*. We call the upper classes *function categories* for distinction to the classes in functional taxonomies. The function categories are types of ontological definitions of functions and then are super-types of the function classes in the taxonomy. Thus, a set of the function terms in a functional taxonomy is *not* sub-set of the reference ontology. It is not our goal to build such an ontology that subsumes all the functions in the existing taxonomies. By a reference ontology, we here mean that the ontology is referred to for categorizing existing definitions of function and mapping them (in comparison with "reference for system design" such as the ISO's OSI network reference model). We cannot claim the completeness of the reference ontology in nature. Section 5.1 discusses the some function categories defined in our 'prescriptive categories of function' as a reference

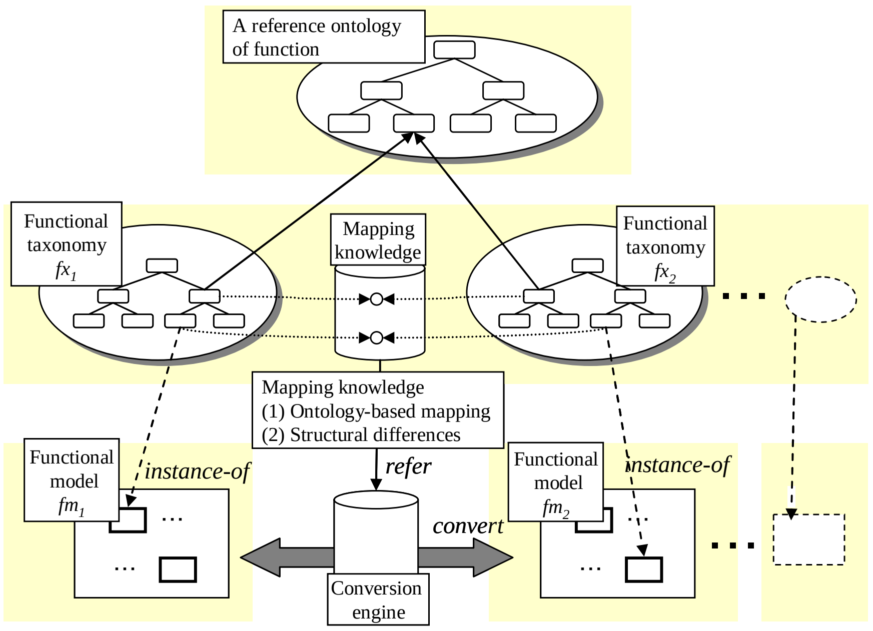

A reference ontology of function
Functional taxonomy fx_1
Mapping knowledge
Functional taxonomy fx_2
Mapping knowledge
(1) Ontology-based mapping
(2) Structural differences
Functional model fm_1
instance-of
refer
convert
Conversion engine
Functional model fm_2
instance-of

Figure 1. An overview of the framework for mapping between functional taxonomies

ontology.
On the basis of the reference ontology, the mapping knowledge is described for each pair of two functional taxonomies. The mapping knowledge consists of (1) mappings between the function classes (terms) in the functional taxonomies and (2) structural differences between the functional models based on the taxonomies. The former is based on the reference ontology in order to make ontological differences clear. The functions categorized into the same class in the reference ontology can be translated into each other directly. This is a terminological difference. On the other hand, if functions are categorized into the different classes in the reference ontology, the mapping becomes complex for bridging the ontological gap. The reference ontology of function aims at making ontological difference explicit and thus we can realize the mapping without loss of information. Sections 5.2 and 5.3 discuss such mappings.
The latter mapping knowledge describes structural difference between the functional models, which is a result of the difference of how function or object is captured by the users (or developers) of the functional taxonomies. For example, in a model based on FB, a material flow and an energy flow are described independently, while the energy-operand in a FBRL model exists with a medium of the material class.
One of the usages of such mapping knowledge is conversion of the functional models. A functional model fm_1 based on a functional taxonomy fx_1 can be converted to a functional model fm_2 based on other taxonomy fx_2. Firstly, using the mappings between the function classes of fx_1 and fx_2, a function in fm_1 can be translated to the mapped function based on fx_2 in fm_2. As a result, we get a functional model fm_2 that consists of function instances of the classes of fx_2, though the structure of fm_2 is the same as that of fm_1. Then, structural modification of the term-translated model fm_2 is made according to the mapping knowledge about structural difference. Note that some structural differences cannot be automatically modified due to the lack of information in the original model.
In this framework, a mapping knowledge is prepared for each pair of functional taxonomies (fx_1 and fx_2). The categorization of functional terms in a functional taxonomy fx_1 is, however, independent of the target taxonomy fx_2. Such categorization based on the reference ontology makes ontological difference clearer and makes the mapping easier. Furthermore, if the target functional terms are categorized into the same category, we can safely use natural language processing techniques based on lexical terms, though such a lexical issue is beyond the scope of this paper.

4 TARGET FUNCTIONAL TAXONOMIES

This section gives overviews of the two taxonomies as examples of targets of mapping.

4.1 Reconciled Functional Basis (FB)

The reconciled functional basis (FB) has been proposed by Hirtz et al.[9], which is a result of reconciliation of two taxonomies; the previous Functional Basis [15] and the NIST taxonomy [23]. It is a result of empirical generalization of product functions and founded on a great number of empirical studies. It is based on the notion of function in [13] as general input-output relationship of a black-box. In [15] a function is defined as “a description of an operation to be performed by a device or artifact”. A function is expressed as an active verb. The recipient of the function’s operation is called ‘flow’ and expressed as the object of the active verb. FB consists of a taxonomy of flow (‘flow set’) and a taxonomy of function (‘function set’). Each of functional concepts is defined in natural language with examples and correspondents. For example, ‘separate’ is defined as “to isolate a flow (material, energy, signal) into distinct components. The separated components are distinct from the original flow, as well as each other”. A functional model based on FB is hierarchical as function structure of Pahl & Beitz.

4.2 Functional Concept Ontology based on FBRL (FBRL)

The authors have developed a modeling framework for functional models [21][24], which includes an ontology of generic types of functions (called the functional concept ontology). The definitions of functions are based on a device-centered viewpoint. The *behavior* of a device is defined as the objective (without designer’s intention) interpretation of its input-output relation as a black box. A device is connected to another device through its input or output ports. A device as an agent changes states of things input (called *operands*) such as substance like fluid, energy, motion, force and information. The input-output relation of the behavior is, to be exact, the difference between the states of the operand at the input port and that at the output port.

The authors defines a (base-)function as a role played by such behavior in a specific context of use [24][42] . The context of use of a product depends on intentions of users or designers. The context of use of a component depends on the system that the component embedded in. The functional concept ontology defines generic types of the base-functions (called functional concepts). It is organized in four *is-a* hierarchies [21]. It has been restructured into single *is-a* hierarchy and redefined recently. A functional concept (a class of function) is defined ontologically using constraints on the cardinality of operands, relationships among them and/or designer's intention to change (focus of intention). For example, a function, "to *divide* an operand", is defined as that a focused operand as input is separated into some sub-pieces which are made from the input. Its ontological definition consists of (1)the cardinality of the input focused operand must be 1, (2)the cardinality of the output focused operands must be grater than 1, (3)there must be material-product relationship between the input operand and the output operands and (4)all the output operands are equally focused. Furthermore, the *dividing* function has three sub-classes, '*split'*, *'decompose'* and *'detach'*. The output operands of the *splitting* function are the same kind, while those of *decomposing* or *detaching* are different kinds. *Detaching* implies spatial separation (mechanical detachment) of sub-portions, while *decomposing* implies chemical decomposition. In the former case we can recognize the sub-portions to be outputted at the input, while we cannot in the latter case. The examples are "scissors *split* a paper", "a distiller *decomposes* oil" and "a washer *detaches* dirt from cloth". Note that we never claim the appropriateness of labels of these concepts and our framework copes with *usual functional words* for easy understanding by engineers [42], which are similar to 'correspondents' in FB.

In addition to base-functions as roles of behaviors for operand(s), the functional concept ontology includes the *meta-functions*, which are collaborative roles played by base-functions for other base-functions such as ToDrive and ToProvide [21]. For example, ToDrive is defined as a role played by a base-function in the context that a base-function supplies energy driving another base-function.

On the basis of the ontologies, a *function decomposition tree* is described as a functional model of a concrete artifact, in which all functions are instances of the generic function classes defined in the functional concept ontology. This is based on the German-style function decomposition in [13]. In our framework, the modeling of "way of function achievement" has been introduced as conceptualization of background knowledge of functional decomposition such as physical principles and theories. Physical principles as a basis of function achievement is also captured as "means" in [6][12], we discuss generic knowledge and its organization based on ontological principles with a clear relationship with other relations such as *is-a* relations of functions [42] .

This framework has been successfully deployed in a manufacturing company in Japan since 2001 for sharing engineers' knowledge about manufacturing devices and manufacturing processes [42]. Its empirical evaluation by the engineers is very positive. They said that this framework enable them to explicate the implicit knowledge possessed by each designer and to share it among colleagues.

In comparison with FB, our functional concept ontology shares a device-oriented viewpoint with FB. The 'operands' in FBRL roughly corresponds to 'flows' in FB. We try to define such viewpoint explicitly as an ontology for consistent modeling [42]. We distinguish a function from a behavior and from a way of function achievement. The distinction from behavior is based on designer's (or user's) intention. Although the definition of a function of a product in FB is based on intention of designers (i.e., purpose), in the definition of the taxonomy such intentions are implicit. Garbacz points out some problems of the classification of FB such as ambiguities of concepts, lack of principle of categorization and non-exhaustiveness from logical and ontological viewpoint [22]. Our definition of function based on intention is similar to those in [5][10][17]. We try to give clear ontological definitions in an *is-a* hierarchy based on the distinctions with other similar concepts such as the way of function achievement and clear definitions of relationships among functions [42].

5 ONTOLOGY-BASED MAPPING BETWEEN THE FUNCTIONAL TAXONOMIES

5.1 A reference ontology of function

The upper-level categories of functions [25] used as a reference ontology of function in this paper defines generic (upper) classes of various kinds of concrete function performed by artifacts. The followings explain its overview and some of major classes shown in Figure 2. Please refer to [25] for other detailed definitions.

An *effect function* is the main category of this ontology and is based on an *effect* which represents temporal changes in attributes of a target thing. The target thing is affected by other thing (an *agent*) and then the *effect* is caused. Such an effect plays a role in a specific teleological context as a function.
The *effect function* has three sub-types; a *device function*, an *environmental function* and *a system interface function*. The *device function* implies changes in entities within the system boundary, while the *environmental function* includes changes outside of the system boundary, especially, those related to users. For example, an electric fan performs moving-air function as a *device function* and cooling function for the human body as an *environmental function* at the same time. Since the cool-down effect by wind appears on the human body, it is out-side of the system boundary thus “environmental”. This cooling function implies physical changes (called *physical environmental function*), while an *interpretative function* sets up one of the necessary conditions of human’s cognitive interpretation. The examples of the latter kind are “to make a man comfortable” function of the electric fan and “to inform time” function of a clock. In the literature, there are similar concepts such as “environment function” [2] and purpose [3][5][10][11]. The *system interface function* represents effects only at the system boundary such as ‘import’ in FB as discussed in Section 5.3.2.
The *device function* is further categorized into an *effect-on-state function* and an *effect-on-process function*. The *effect-on-state function* refers to changes in physical attributes of a physical entity as a target thing. It has a sub-type, a *flowing-object* function, whereby the physical thing flows through a device that causes the changes. The *flows* in FB and *operands* in FBRL can be regarded as the *flowing-objects*. Thus, many functions in FB and all base-functions in FBRL belong to the *flowing-object function* class. On the other hand, the *effect-on-process function* is based on an effect on a process or change. It subsumes the meta-functions in FBRL (mentioned in Section 4.2), each of which represents a role of a base-function for other base-functions that contribute to the same whole function. It subsumes the primary and secondary functions [13], and the assisting functions [10][11] as well.
Moreover, we recognize the some kinds of *quasi-functions*. For example, a *function-with-way-of-achievement* implies a specific way of function achievement as well as a function. For example, “to weld metals” as a function of a welding machine 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. In our terminology as discussed in Section 4.2, the former is a *base-function*, while the latter is a *way of function achievement*. Because meaning of this type of function is impure, we regard this quasi-function. Other examples include washing, shearing and adhering (e.g., glue adheres A to B).
In the rest of this section, we discuss mapping between the functional taxonomies based on the reference ontology above. This mapping can be categorized into two types: “mapping within the same category of function” and “mapping between different categories of functions”. We discuss them in order.

5.2 Mapping within the same category of function

Consider the mapping between a function term V-F1 in a functional taxonomy F1 and a functional term V-F2 in other taxonomy F2. If V-F1 and V-F2 are categorized into the same category of function in the reference ontology, they can be mapped into each other directly. For example, all base-functions of FBRL and many terms of FB are categorized into the same category “*flowing-object function*” in the reference ontology. Thus, in case of mapping among many terms of FB and those of FBRL, we use the mapping within the same category of function. Mapping within the same category of function can be categorized into two types of mapping as follows: “one to one mapping” and “one to N mapping”.

5.2.1 One to one mapping

If the functional term V-F1 corresponds to V-F2 only and vise versa, we call this mapping “one to one mapping”. For example, “distribute” in FB is defined as “to cause a flow to break up. The individual bits are similar to each other and the undistributed flow.” In FBRL, as mentioned before, “split” is defined as the case that the separated entities are the same kind. Thus, “distribute” in FB corresponds to “split” in FBRL (Figure 2(a)). Moreover, no other term corresponds to these terms. Thus, this mapping is called “one to one mapping”.

In this case, when a functional model M-F1 based on F1 is translated into a model based on F2, each function instance of the class V-F1 can be translated into an instance of the class V-F2.

5.2.2 One to N mapping

When V-F1 corresponds to more than two functional terms of another functional taxonomy F2, we call this “one to N mapping”. Generally speaking, “one to N mapping” is used when the principles (or criterion) for classification in taxonomies are different each other.

In FB, “branch” is classified into “separate” and “distribute” by the principle for classification of “whether the separated components are distinct from the original flow or not”. On the other hand, in FBRL, “separate” is classified into “divide” and “take out” by the principle for classification of “whether one of the separated entities is focused or all of them are focused” as discussed in Section 4.2. Thus, the principles for classification of “separate” in FB and FBRL are different from each other. Furthermore, “divide” in FBRL is classified into “split”, ”detach” and “decompose” by the principle for classification of “whether the separated entities are the same kind or not”. According to such definitions, the ”separate” in FB possibly is mapped to three functional terms in FBRL: “take out”, “detach”, and “decompose” (Figure 2(b)). This one-to-N mapping is asymmetric in the hierarchies of functional terms due to the differences of the principles of classifications. As mentioned in 5.2.1, “distribute” in FB corresponds to “split” in FBRL.

Another example is “transfer” in FB. “Transfer” in FB is defined as “To shift, or convey, a flow from one place to another.” Moreover, the sub-classes of “transfer” in FB are classified by the kind of the flow. On the other hand, in FBRL, there are three functional terms about changes of a position of an entity: “move”, “change distance” and “change medium” The “move” implies only the movement of one entity. The “change distance” focuses on change in the distance between the moving entity and other entity. The “change medium” focuses on the medium of the moving entity and then implies the transfer of the entity from a medium to other medium. No term in FBRL is classified by the kind of the operands. Thus, “transfer” in FB corresponds to those three functional terms in FBRL.

In this case, when a functional model based on F1 is translated into a model based on F2, it is necessary to choose appropriate one of the N functional terms in F2 which correspond to V-F1. For this choice, we consider the “context of the usage” about functional term V-F1 first. For example, when “separate” in a functional model of FB is translated into a functional model of FBRL, it is

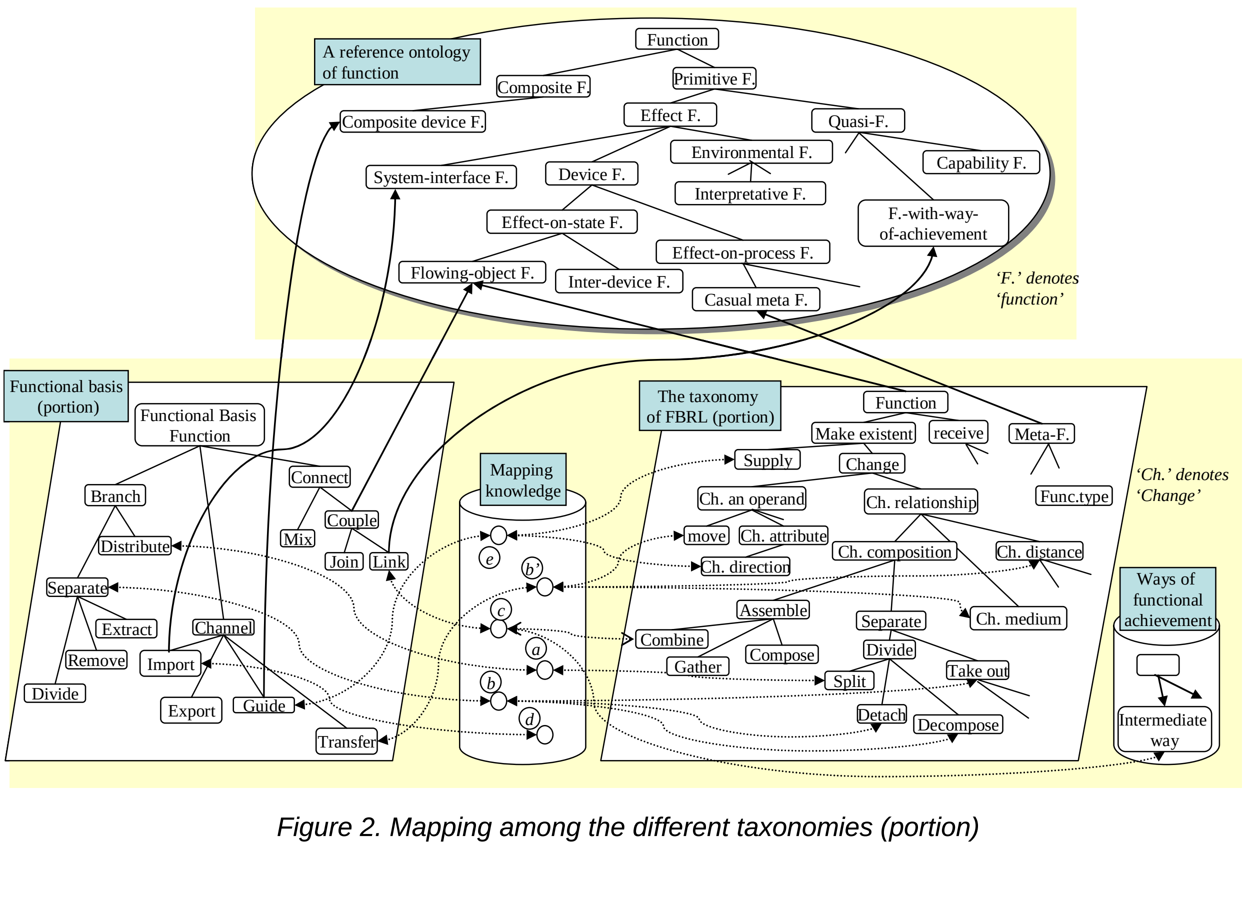


Figure 2. Mapping among the different taxonomies (portion)

important to consider the context of “separate” in the model of FB in order to choose appropriate one of three functional terms, i.e. “detach”, “decompose” or “take out”.

5.3 Mapping between different categories of functions

If V-F1 and V-F2 are categorized into different categories in the reference ontology of function, the mapping between V-F1 and V-F2 becomes more complex. We call this mapping “mapping between the different categories of function”. For example, all functional terms in FBRL are categorized into the *device function* category in the reference ontology. On the other hand, some terms of FB are categorized into the *device function* category, and others are categorized into the following categories: the *function-with-way-of-achievement* category, the *system-interface function* category and the *composite device function* category. In this section, we describe the mapping with examples of the mapping between different categories of FBRL and FB.

5.3.1 Function-with-way-of-achievement

Each of the functional terms categorized into the *function-with-way-of-achievement* category implies “way of function achievement” as mentioned in 4.2. A functional term in FB categorized into the *function-with-way-of-achievement* corresponds to a functional term in FBRL plus a way of function achievement. For example, the functional term “link” in FB implies not only “to couple flows together” [9] as the change at input and output but also “by means of an intermediary flow” [9] as how to realize it. An example of the usage is “a turnbuckle links two ends of a steering cable together” [9], which means that the intermediary flow is used to couple the target ends as the turnbuckle itself is an agent. Thus, according to the definition, the “link” in FB seems to be categorized into the *function-with-way-of-achievement* category. On the other hand, the term “combine” in FBRL implies “to bring two operands into an operand” as the change at input and output, which corresponds to the former part of the meaning of “link”. The latter of the meaning correspond to not the “combine” function but the “intermediate-object” way to achieve the combining function. Thus, “link” in FB is translated into “combine” in FBRL plus the “intermediate-object” way for achievement of combining (Figure 2(c)).
Moreover, “join” in FB is translated into “combine” in FBRL plus “the non-intermediate-object” way for achievement of combining. Generally speaking, when we consider mapping of a functional term categorized into the *function-with-way-of-achievement* category, it is efficient to remove the way of function achievement from it firstly.

5.3.2 System-interface function

The s*ystem-interface function* category covers the terms that represent the changes of behaviours on the system boundary. Both “import” and “export” in FB are categorized into this category. On the other hand, there is no term in FBRL categorized into this category. Thus, “import” in FB is not translated into the functional term in FBRL but into an input operand from external area of the device in the functional model of FBRL (Figure 2(d)). In the same way, “export” in FB is translated into an output operand from internal area of the devices of FBRL to the external area.
Conversely, in the translation from the functional model based on FBRL into a model based on FB, every input operand from the external area in the functional model of FBRL is translated into “import” in FB. As for an output operand, if the output operand is focused by the functional term at the end of the modeling world defined from the device-centered viewpoint, it is translated into “export” in FB.

5.3.3 Composite device function

The *composite device function* category indicates a function term including the definitions of more than two functional terms categorized into the *device function* category in the reference ontology. For example, “guide” in FB is defined as “to direct the course of a flow (material, energy, signal) along a specific path”. It is a sub-type of “channel” as “to cause a flow to move from one location to another location”. Thus, it seems to imply the both meanings. The examples given in [9] support this observation. Thus it corresponds to two functional terms in FBRL: “supply motion[1]” plus “change direction of motion”. Moreover, some of the functional terms categorized into the *composite device function* category include meta-function(s) which coordinates two functions in order to work together.

[1] By “motion” in FBRL we mean conceptualization of states of an object which is moving as a kind of operands, and roughly corresponds to the mechanical energy in FB.

Conversely, the translation of “supply motion” and “change direction of motion” in FBRL into functional terms in FB is asymmetric. There are three cases. First, in case that there is a set of “supply motion” and “change direction of motion” in the functional model based on FBRL, they are simply gathered and translated into “guide” in FB. Second, in case there is “supply motion” without “change direction of motion” in the model, it is translated into “channel” in FB as “one to one mapping”. Finally, in case there is “change direction of motion” without “supply motion” in the model, we translate “change direction of motion” into “guide” in FB. Although to complement the original meaning with the additional meaning on the process of translation is undesirable basically, we do it since the main part of “guide” is to change the direction of mechanical energy in order to maintain the direction within the appropriate range.

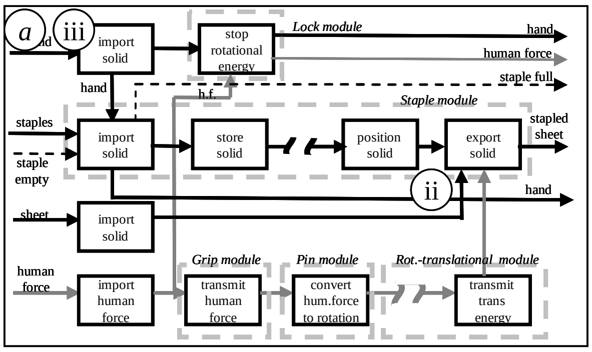
6 STRUCTURAL DIFFERENCES BETWEEN THE FUNCTIONAL MODELS

There are general structural differences among the functional models based on the different functional taxonomies. One of the causes of the structural difference is the differences of how to capture the functions or the objects in each of the functional taxonomies. Thus, the mapping between the functional terms without taking the structural differences into consideration may result in the loss of information of the terms. Therefore, we need to reveal the hidden differences of how to capture the modeling world ontologically which the functional terms and taxonomies have. After that, we can support the improvement of interoperability based on the structural differences revealed.
Figure 3 shows the functional models of a stapler described with FB (hereafter FB model (Figure 3(a))) [43] and FBRL (hereafter FBRL model (Figure 3(b))). Comparing those two functional models as examples, we specified the following six generic structural differences between FBRL and FB models.

(1) The difference in the relation of achievement among the functions
The FBRL model shows the whole function (“combine the sheets” (Figure 3(b)(i))) and the result of decomposing the whole function into sub-functions hierarchically. Thus, a FBRL model shows what kinds of the sub-functions achieve the whole function explicitly. On the other hand, a FB model shows the function of the components of the stapler explicitly, but achievement relation among component’s functions, module’s functions and the whole function is implicit (cf. Figure 3(a)).

(2) The difference in representation of connection among functions
In a FB model a graph of flows shows explicitly how functions are connected in terms of flows. On the other hand, the FBRL model shows implicitly such function chains. In the FBRL modeling schema, a behavioural model associated with a functional model shows the connection (temporal) relations among devices as agents of functions in terms of flow of operands.

(3) The difference in the way of achievement
In a FBRL model, the way of function achievement is expressed separately from the function. On the other hand, in a FB model the way of function achievement is not separated from the function explicitly.



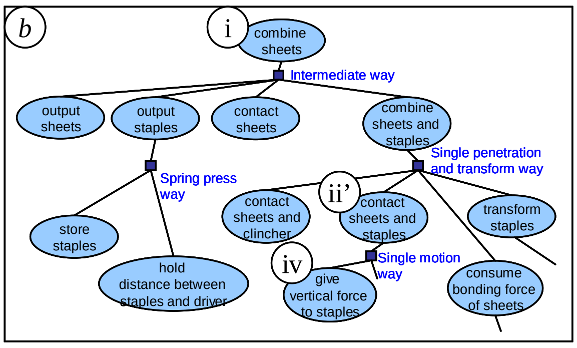

b
i
combine sheets
Intermediate way
output sheets
output staples
contact sheets
combine sheets and staples
Spring press way
Single penetration and transform way
store staples
hold distance between staples and driver
contact sheets and clincher
ii’
contact sheets and staples
transform staples
iv
Single motion way
give vertical force to staples
consume bonding force of sheets

Figure 3. The functional models of a stapler (portion): (a) a FB model, (b) a FBRL model.

(4) The difference of how to capture the change of the distance between the objects
The arrow in the FB model expresses the kind of flow and leads the flow to the next function. Moreover, by joining the arrows, as shown in Figure 3(a)(ii), it seems to imply the change of the distance between the flows (staples and sheet) implicitly. On the other hand, the function in the FBRL model, as the example of "contact the staples and the sheets" (Figure 3(b)(ii')), expresses the change of the distance between the objects explicitly.

(5) The difference of how to capture the user of the device
The FB model treats a part of the user as the flow (e.g., "hand" in Figure 3(a)(iii)). On the other hand, the FBRL model does not treat the user as the operand basically. The reason for this is that in FBRL the user exists external to the device and the FBRL modeling method does not model it as well as other external things.

(6) The difference of how to capture the material and the energy
In a FB model, a material flow and an energy flow are described independently. On the other hand, in a FBRL model the energy-operand (e.g., the "vertical force" in Figure 3(b)(iv)) exists with an instance of the material class as a medium (e.g., "staples" in Figure 3(b)(iv)).

7 CONVERSION OF THE FUNCTIONAL MODELS

In this section we discuss the conversion of functional models using a stapler as an example. Here we shows possible conversion from a stapler model based on FB [43] shown Figure 3a (FB model) into a model based on FBRL shown Figure 4 (the converted model). The conversion of functional models between two taxonomies needs both the translation of the functional terms between the taxonomies and the transformation of the structural differences between the models. The translation of the functional terms can be done by replacing the functional terms in the FB model with the functional terms in the FBRL taxonomy by using the ontology-based mapping knowledge, as mentioned in Section 5.
In the transformation of the structural differences we consider the differences (4) and (5) in Section 6. To convert over the structural difference (4), the function "contact the flows" is inserted into the converted model at the point of the joint of the arrows in the original FB model. In Figure 3(a)(ii), corresponding to a joint of arrows which denote staples and sheets, the function "contact staples and sheets" is inserted as the result of the transformation (Figure 4(i)). To go over the structural difference (5), the function whose input or output is a part of the user as flow is removed as a result of the transformation. In Figure 3(a)(iii), the output of the function "import solid (hand)" is a part of the user as a flow. Thus, the "import solid (hand)" is removed in the converted model (Figure 4(ii)).
The structural differences (1) and (3) cannot be automatically transformed, because the way of function achievement in FBRL is implicit in the FB model. The transformations for the structural differences (2) and (6) in Section 6 are under investigation.

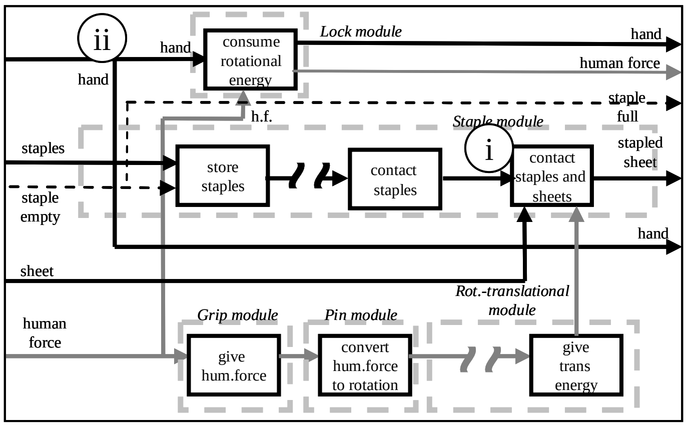


Figure 4. The stapler FBRL model converted from the FB model shown in Fig. 3a (portion)

8 CONCLUSION

In this article, we aimed at improving interoperability between different functional taxonomies and proposed a framework for conversion of functional models using the ontology-based mapping. The important parts of this framework are ontology-based mapping between the functional taxonomies and transformation of structural differences between the functional models. The reference ontology in this framework helps mapping-authors make ontological difference explicit and then makes the mappings between taxonomies easier. As the result, it contributes towards reducing the loss of information in the mapping between the functional taxonomies. Furthermore our framework makes it easy for the designer to understand the functional model described in unfamiliar taxonomy by conversion of the model. As mentioned in Section 7, we need to investigate how to perform the transformation of the structural differences (2) and (6) in Section 6. As future work, we plan to apply this framework to other various functional taxonomies. Moreover, we plan to implement this framework and make the evaluation about interoperability using this implementation.

REFERENCES

[1] Chandrasekaran, B., Goel, A.K., & Iwasaki, Y. Functional representation as design rationale. *Computer*, 1993, 48-56.

[2] Chandrasekaran, B. & Josephson, J.R. Function in device representation. *Engineering with Computers 16(3/4)*, 2000, 162-177.

[3] 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, 23(6),* 2000, 1718-1751.

[4] Chittaro, L. & Kumar, A. N. Reasoning about function and its applications to engineering. *Artificial Intelligence in Engineering, 12*, 2000, 331-336.

[5] Lind, M. Modeling goals and functions of complex industrial plants. *Applied artificial intelligence, 8*, 1994, 259-283.

[6] Bracewell, R.H. & Wallace, K.M. Designing a representation to support function-means based synthesis of mechanical design solutions. In *Proc of ICED 01*, 2001.

[7] Gero, J.S. Design prototypes: a knowledge representation schema for design. *AI Magazine, 11(4)*, 1990, 26-36.

[8] Gero, J.S. & Kannengiesser, U. The situated function-behaviour-structure framework. In *Proc. of Artificial Intelligence in Design '02*, 2002, 89-104.

[9] Hirtz, J., Stone, R.B., McAdams, D.A., Szykman, S. & Wood, K.L. A functional basis for engineering design: reconciling and evolving previous efforts. *Research in Engineering Design, 13*, 2002, 65-82.

[10] Hubka, V. & Eder, W.E. *Theory of technical systems*. Berlin: Springer-Verlag, 1988.

[11] Hubka, V. & Eder, W.E. Functions revisited. In *Proc. of ICED 01*, 2001.

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

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

[14] Rosenman, M.A. & Gero, J. S. Purpose and function in design: from the socio-cultural to the techno-physical. *Design Studies, 19*, 1998, 161-186.

[15] Stone, R.B., Wood, K. L. Development of a Functional Basis For Design. *Journal of Mechanical Design*, 122(4):359-370, 2000.

[16] Stone, R. B. & Chakrabarti, A. (Eds.) Special Issues: Engineering applications of representations of function, *AI EDAM, 19(2/3)*, 2005.

[17] 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, 10*, 1996, 275-288.

[18] Umeda, Y., & Tomiyama, T. Functional reasoning in design, *IEEE Expert,* March/April, 1997, 42-48.

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

[20] Sushkov, V.V., Mars, N.J.I. & Wognum, P.M. Introduction to TIPS: a Theory for Creative Design. *Artificial Intelligence in Engineerin*g, *9*, 1995, 177-189.

[21] Kitamura, Y., Sano, T., Namba, K., & Mizoguchi, R. A functional concept ontology and its application to automatic identification of functional structures. *Advanced Engineering*

Informatics, 16(2), 2002, 145-163.

[22] Garbacz, P. Towards a standard taxonomy of artifact functions. In *Proc. of FOMI 2005*, 2005.

[23] Szykman, S., J. W. Racz and R. D. Sriram. The Representation of Function in Computer-based Design. In. *Proc. of the 1999 ASME Design Engineering Technical Conferences, DETC99/DTM-8742*, 1999.

[24] Kitamura, Y., Washio, N., Koji, Y., and Mizoguchi, R. Towards Ontologies of Functionality and Semantic Annotation for Technical Knowledge Management. *New Frontiers in Artificial Intelligence*, Lecture Notes in Computer Science 4012, Springer, pp. 17-28, June, 2006.

[25] Kitamura, Y., Takafuji, S., Mizoguchi, R., Towards a Reference Ontology for Functional Knowledge Interoperability, In *Proc. of ASME IDETC/CIE 2007, DETC2007-35473, to appear.*

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

[27] Maier, A., Schnurr, H. P., Sure, Y.: Ontology-based Information Integration in the Automotive Industry, In *Proc. of ISWC 2003*, 2003, pp. 897-912.

[28] Kopena, J. B., Regli, W. C.: Functional Modeling of Engineering Designs for the Semantic Web, IEEE Data Engineering Bulletin, *IEEE Computer Society*, 26(4), 2003, 55-62.

[29] Kitamura, Y., Washio, N., Koji, Y., Sasajima, M., Takafuji, S., & Mizoguchi, R. An ontology-based annotation framework for representing the functionality of engineering devices. In *Proc. of ASME IDETC/CIE 2006*, DETC2006-99131, 2006.

[30] Borst, P. Akkermans, H., and Top, J. Engineering Ontologies. *Human-Computer Studies*, 46(2/3), 1997, pp. 365-406.

[31] Li, Z., Anderson, D. C., Ramani, K. Ontology-based Design Knowledge Modeling for Product Retrieval. In *Proc. of ICED 2005*, 2005, 462.1.

[32] Guarino N. Some Ontological Principles for A Unified Top-Level Ontology. *Proc. of AAAI Spring Symposium on Ontological Engineering*, 1997.

[33] Suggested Upper Merged Ontology, http://ontology.teknowledge.com/

[34] ISO TC184/SC4/JWG8. Process Specification Language, http://www.tc184-sc4.org/SC4_Open/SC4_Work_Products_Documents/PSL_(18629)/, 2003.

[35] Schlenoff, C., Denno, P., Ivester, R., Libes, D., Szykman, S. An Analysis and Approach to Using Existing Ontological Systems for Applications in Manufacturing. *AI EDAM*, 14, 2000, pp. 257-270.

[36] Ahmed, S., Kim, S., Wallace, K. M. A Methodology for Creating Ontologies for Engineering Design. In *Proc. of ASME 2005 DTM*, 2005, DETC2005-84729.

[37] Cutkosky, M.R., et al. PACT: An experiment in integrating concurrent engineering systems. *Computer, January*, 1993, 28-37.

[38] Yoshioka, M., Umeda, Y., Takeda, H., Shimomura, Y., Nomaguchi, Y. & Tomiyama, T. Physical concept ontology for the knowledge intensive engineering framework. *Advanced Engineering Informatics, 18(2)*, 2004, 95-113.

[39] Dartigues , C., and Ghodous. P. Product Data Exchange Using Ontologies. In *Proc. of Artificial Intelligence in Design* (AID2002), Cambridge, UK, 2002, 617-637.

[40] Koji, Y., Kitamura, Y. & Mizoguchi, R. Ontology-based transformation from an extended functional model to FMEA. In *Proc. of the 15th International Conference on Engineering Design (ICED '05*), CD-ROM, 2005, 264.81.

[41] Noy, N. Semantic integration: A survey of ontology-based approaches. *ACM SIGMOD Record archive*, Vol. 33, Issue 4, 2004.

[42] Kitamura, Y. Koji, Y., and Mizoguchi, R. An ontological model of device function: industrial deployment and lessons learned. *Applied Ontology*, Vol. 1, No. 3-4, 2006, pp. 237-262.

[43] Stone, R., McAdams, D. and Kayyalethekkel, V. A Product Architecture-Based Conceptual DFA Technique. *Design Studies*, 23(3), 2004, 301-325.

Contact: Masanori Ookubo
The Institute of Scientific Industrial Research, Osaka University
8-1, Mihogaoka, Ibaraki, Osaka, 567-0047 Japan
Tel: +81-6-6879-8416 Fax: +81-6-6879-2123
E-mail: mokubo@ei.sanken.osaka-u.ac.jp
URL: http://www.ei.sanken.osaka-u.ac.jp/