Part 3: Advanced course of ontological engineering
Author: Riichiro Mizoguchi

Abstract

This article deals with advanced topics of ontological engineering to convince readers ontology is more than a rule base of terminological problems and is worth to consider a promising methodology in the next generation knowledge processing research. Needless to say, ontology in AI is tightly connected to ontology in philosophy. The first topic here is on philosophical issues which are very important to properly understand what an ontology is. After defining class, instance and *is-a* relation, we point out some typical inappropriate uses of *is-a* relation in existing ontologies and analyze the reasons why. Other topics are basic ontological distinction, part-of relation, and so on. As an advanced example of ontology, an ontology of representation is extensively discussed. To conclude this tutorial, a success story of ontological engineering is presented. It is concerned with a new kind of application of ontology, that is, knowledge systematization. An ontology-based framework for functional knowledge sharing has been deployed into a company for two years and has been a great success. Finally, future of ontological engineering is discussed followed by concluding remarks.

1. Fundamental issues

Building an ontology needs a lot of skill and knowledge about fundamental issues. This section discusses issues which have to be seriously investigated to build a good ontology.

1.1 Background

One of the most critical contributions of an ontology is that it gives the higher level distinction of concepts which help understand the lower level concepts, that is, domain concepts systematically and consistently which is hard to attain without ontological ways of thinking. The main topic here is so-called ontological distinction which is indispensable for designing upper ontology. An ontology

design is a kind of design activity which necessarily has some design rationale that largely influences the resulting ontology. In other words, any ontology cannot be free from some assumption and/or designer's standpoint. The standpoint taken in this article consists of Newtonian world point of view and 3D modeling, that is, the world is considered as being composed of the three-dimensional Euclidean space with the absolute time and both object(continuant) and process (occurrent) exist with equal importance. The latter is discussed in detail below.

One of the difficulties in understanding ontology research is the definition of each concept. How deeply and rigorously should we define a concept? How is an ontology different from a dictionary in word definition? These are common questions about this topic. A dictionary consists of definitions of terms people use for human consumption and basically tries to cover as many terms as possible. Some says an ontology is a computer-understandable dictionary. Although it is not incorrect, it tends to lead people to a misunderstanding about ontology. It is critical to note that in principle, it is impossible for us to define the meaning of each term/concept in a computer as we understand it. Try to define what a human is as you understand it. A human is living. What is "living"? He/she falls in love, laughs, cries, eats, produces an artifact, walks, dies, etc. How can we represent them in the computer? What are music, poem, dog, forest, etc.? How is a horse different from a cow? Etc. etc. Answering such questions is not the job of ontology researchers. Those who answer them are domain experts or lexicographers.

Then, what is the job of ontology researchers? Ontology research in ontological engineering is mainly concerned with something like *meta-questions* such as what is identity?, how is identity inherited?, what is a proper taxonomy?, what is class/instance?, what is a role?, what is a whole(what makes it a whole rather than just a collection of things)?, what are *is-a* and *part-of* links?, how are an object and a process different?, etc. Once again, even when investigating what is an instance, ontology researchers never try to define what a particular instance, say, Mr. A is. It is just like building a taxonomy of animals is not the ontology researcher's but zoologist's job.

1.2 Class, instance and *is-a* relation

Let us share the very fundamental idea of class, instance and *is-a* relation in ontology, though the definition 2 is my own idea.

> Definition 1: Intrinsic property
> The intrinsic property of a thing is a property which is essential to the thing and it looses its identity when the property changes.
>
> Definition 2: The ontological definition of a class
> A thing which is a conceptualization of a set X can be a class if and only if each element x of X belongs to the class X if and only if the "intrinsic property" of x satisfies the intensional[1] condition of X. And, then and only then, < x *instance-of* X> holds.
>
> Definition 3: *is-a* relation
> *is-a* relation holds only between classes. <class A *is-a* class B> holds iff the instance set of A is a subset of the instance set of B.

These definitions shows that *is-a* relation used in object-oriented languages is very different from this ontological one which does not allow people to build a model like <Mr. A *instance-of* teacher> and <teacher *is-a* human>, since there is no person whose intrinsic property is being a teacher. As discussed in 2.2.4 of Part 2, if we model

---

[1] A set can be defined in two ways: extensional and intensional ways. The former definition is made by enumerating all the elements and the latter definition by specifying a necessary and sufficient condition for being its elements.

<Mr. A *instance-of* teacher> and <teacher *is-a* human>,
then we have a difficulty in managing instances when Mr. A quits his job.

As is seen from the above discussion, the rest of this article deals with a very generic examples rather than domain-specific ones to make the discussion as general as possible.

1.2.1 Inappropriate use of *is-a* link

The skeleton of an ontology is an *is-a* hierarchy. Its appropriateness is critical in the success of ontology building. As Guarino's remarks, however, there are quite a few inappropriate use of *is-a* relation in the existing ontologies. The main causes include the lack of proper semantics of *is-a* relation. The following is a list of some typical examples of misuse of *is-a* relation indicated by Guarino[Guarino 98].

(a) <physical object *is-a* amount of matter>
It is true that any physical object is made from matter. However, identity criteria of these are different. Matter of some amount is still the same matter after it is divided into half, while most of physical objects loose their identity by such division. Therefore, physical object cannot be a subclass of amount of matter, since it contradicts with the principle of identity criterion preservation in an *is-a* hierarchy.
(b) <association *is-a* group(of persons)>
The reason is the same as the above. An association can still be the same after some members are changed, while a group is not, since the identity of a group(of persons) is totally based on its members.
(c) <human *is-a* physical object> and <human *is-a* living thing>
This is an example of multiple inheritance which causes a trouble when considering instance extinction. A human loses his/her existence and hence he/she cannot be an instance of human when he/she dies, while his/her body still exists as a physical object. The system needs a special managing function of extinction of an instance for such a case together with a mechanism for representing necessary information on identity inheritance. To avoid such a difficulty, an ontology is required to have a class *human body* under the *physical object* separately from *human,* which has *human body* as its part, under *living thing*.
(d) <organization *is-a* group> and <organization *is-a* social entity>
The reason why this is inappropriate is understood by (b) and (c) because either one has difference identity criterion. The former corresponds to <human *is-a* physical object>.

1.2.2 The causes of such misuse of is-a relation

A problem common to the above misuse is related to how to deal with dependency of parts on its whole. An association is composed by member people. So, there exists a dependency between members and the association. However, the identity of an association comes not from the members but from its being a social entity by which it become beyond just a group of people. It is analogous to the human example. A human is heavily dependent on the body, but its identity comes not from the body but from the mind[2]. Another kind of misuse of *is-a* relation is found in dealing with the role concepts discussed in Part 2. For these reasons, the definition of a class presented in 1.1 plays a very important role because it avoid most of such difficulties by giving a proper semantics of *is-a* relation.

1.2.3 OntoClean[Guarino 02]

OntoClean is a theory and a methodology for polishing up an ontological taxonomy to become ontologically clean. It carefully analyses the subsumption(*is-a*) relation in taxonomies in an ontology and comes up with four meta-properties such as **rigidity, identity, unity, dependency**. By using

---

[2] The important thing here is not what is "the mind" but where identity comes from.

these metaproperties, OntoClean has some constraints among the values to eliminate invalid combination of these properties in the subsumption hierarchy.

In OntoClean, a term "property" is used to mean a predicate to characterize a thing. Examples of property include *Person, Student, Hard, Red*, etc. A property is said to be **rigid** iff it is essential to all its instances. For example, being a *person* is rigid because any person is a person throughout his/her existence in the real world. Rigid property corresponds to the intrinsic property discussed above. A property is said to be **anti-rigid** iff it is non-essential to all its instances. Being a *student* is anti-rigid because there is no one whose essential property is being a student. A property is said to be **non-rigid** iff it is essential to some instances and not to others. Because being *hard* is essential to a hammer, but not to a sponge, "Being *hard*" is non-rigid. Identity plays a critical role in organizing a proper taxonomy. For example, time duration and time interval are different. Two time intervals of 9:00-10:00 of December 14 and 13:00-14:00 of the same day are different, though these two are the same as time duration. Identity criterion(IC) is a condition which provides a necessary condition for identify a thing. Meta-property concerning identity is defined as whether a property carries IC or not and whether a property supplies IC or not. For example, *Person* carries and supplies IC, while *Student* only carries IC. In a similar way, it defines unity condition and dependency to obtain a new meta-property. Following subsumption constraints such as rigid property cannot subsume anti-rigid property, OntoClean can clean up a taxonomic structure. As described in Part 2, ontology building tools, WebODE and OntoEdit, support OntoClean methodology.

1.3 Ontological distinction

(1) Substrate and entity
Space and time are indispensable for things to exist in the world, while these two can exist independently without the others. Such dependency is essential and differentiates from each other. Matters are less basic than the other two, but it still is very substrate-like because every physical individual is made from matter.

(2) Individual and attribute
Any individual cannot exist without an attribute, that is, anything has necessarily at least one attribute(color, mass, size, etc.). At the same time, any attribute cannot exist alone. It necessarily needs an individual to associate it with. Thus, both an individual and an attribute are inherently dependent on each other and cannot be separated. Such a deep mutual independence is an essential structure of being: object vs. process which will be discussed later, matter vs. physical object are examples.

Concerning attribute, it is necessary for us to have a clear understanding of what it means. There is a confusing concept: property. How are *attribute* and *property* different? To avoid a terminological discussion, let us use examples. Our goal is to properly understand concepts related to these two terms. Any individual has color and an individual's color might be red. Then, what are the concepts corresponding to *color* and *being red*? Apparently, each of the both requires its own category name. My proposal here is that the former can be conceptualized as *attribute* and the latter as *property* because the term *attribute* cannot be used for the latter. In short, *property* is an abstraction of an individual's having the value of an attribute.

(3) Physical and abstract
A physical[3] thing is something which needs time or space to exist. Others are abstract. There is nothing which requires space but does not time to exist. Although this definition is simple and effective, it is still controversial, since a concept like "his hope to success" becomes a physical thing which looks abstract following common sense. However resolution of this is not hard because it is a terminological problem. People use the term physical and abstract without definition, that is, without fully knowing what these two terms mean. Based on the idea that an ontology is terminology- independent, a resolution of this difficulty is not to use the terms

[3] By *physical*, we usually mean *concrete*. It is a convention on ontology research. There is no intention to contrast it with "*chemical*".

physical and *abstract*. Instead, two concepts such as time-dependent thing and time-space independent thing are sufficient for us like Cyc[OpenCyc]. Furthermore, the former could be decomposed further to come up with time-space dependent thing which corresponds to *physical* thing in common sense and time-dependent but space-independent thing.

(4) Continuant(Object) vs. Occurrent(Process)[4]

This is one of the most controversial issues and has a long history of discussion [3Dvs4D]. It is sometimes called 3D model vs. 4D model. Common sense is based on 3D model which consists of the 3D Euclidean space with time. People think there are *objects* which exist quite long time in a stable manner and *processes* which are time-dependent things. In ontology research, an object is defined as a thing all of whose parts exist at a time, while process is defined as a thing which has temporal parts such as the initial phase of burning, terminal phase of burning, etc. The problem is that everything in the world is changing. A person is changing. Reflecting this fact, in the 4D model, which consists of 4-dimensional space in which time is dealt with just a fourth dimension, a person is modeled as a process, that is, it has temporal parts like a process. A person at time t1 and the same person at time t2 are different objects in the 4D model. A person is modeled as the trajectory of these temporal parts. As we discussed this issue in Part 2, the key difference between the two models is how to recognize ***identity*** of an object. 3D ontology admits the two categories: object and process and recognizes the identity of an object, while 4D ontology admits only process and identifies an object as its trajectory in the 4D space because it understands everything is changing. The key issue here is how to differentiate the two: Continuant(Object) and Occurrent(Process). More concretely, how to model a person: continuant or occurrent?

My personal solution to this long-lasting debate would be a revolutionary idea that

Continuant is a role in the context of process.

It is apparent we have to accept the fact that everything is changing. There is nothing which exists in the world forever without any change. At the same time, however, it is also true that we see essential differences between objects and processes. To solve this conflicting issue, it is helpful for us to consider what a process is. There are two kinds of processes in this context.

(a) One such as "walking", "tennis", etc. which require *participants* such as person, ground, players, racket, ball, etc.). Such a process is the (interactive) behavior of them.

(b) One which is the change of the participants themselves. The change process is an internal process of the participants.

Processes of type (a) intrinsically need participants which necessarily play the *continuant role*. Those participants cannot play a process role in (a) from the definition. In the processes of type (b), on the other hand, something playing the *continuant role* also intrinsically exists, since the change process is an internal one in the participants which is already recognized as something playing the *continuant role*. Furthermore, the internal change process is composed of processes of type (a) which in turn identifies other participants of finer grain sizes. Therefore, we always can say there exists a thing which intrinsically plays the continuant role, which is almost equivalent to the idea that *continuant* exists as an independent category. For example, a tornado can cause some destructive effects to its existing area as a process of type (a) and it changes as a process of type (b). We do not try to identify if a tornado is an object or a process. We only have to identify the *continuant role* in it in any case. A tornado and a person are in the same category in this context, that is, both are continuants which continuously change as time goes keeping its diachronic identity. The difference between them is not categorical but matter of degree, that is, the former changes faster than the latter. Therefore, it is safe for us to say "a tornado is a continuant" in the above sense(type a). This shows that object(continuant) and process(occurrent) are co-existent like substance and attribute. Neither can exist without the other. As far as I understand, both continuant and occurrent can be regarded as the top-level categories in an upper ontology like substance and attribute.

---

[4] I intentionally neglect the difference between 4D view and perdurantism for simplicity.

(5) Basic concept and role concept
As described in 2.2.4 in part 2, the concept of role is critically important to understand the world correctly. The real world is a full of roles. Husband is a role played by a man, fuel is a role played by gasoline, coal, etc. As we see in the above discussion, it helps us understand the long-lasting debate on 3D vs. 4D. Similar contribution of the concept of roles is also shown in 1.2 and 1.5.

(6) Entity and relation
Relation is usually considered as abstract. But it is not true, though it is something in the higher order than an entity, that is, entities first exist and relations are something found between entities. An example is the marital-relation with Mr. A and Ms. B which is time-dependent and hence cannot be abstract. Although it is intangible, it exists in the time frame of the real world. People sometimes confuse relation as a formalism with relation as an existing thing. Typical examples are *action* and *attribute* which are sometimes formalized as a relation because an action is often formalized as one between an actor and an object and an attribute as one between an object and a value. But, of course, they are not relations. They are intrinsically independent categories included in an ontology. Friendship between persons, marital relation, part-whole relation, etc. exist in the world.

(7) Representation and non-representation
Representation and symbols are usually dealt with in the semiotics rather than in ontology. However, from the real-world modeling point of view, which is the major use of ontological engineering, we need to deal with representation in our ontology, since there apparently exist music, novels, text, symbols and so on in the real world. Representation and non-representation (object, process, relation, attribute, etc.) are very different from each other. For the representation, it is not easy to identify what an instance is. For example, what an instance of a piece of music needs some consideration.

Although these distinctions are not exhaustive, they are very important when to build an ontology. Some of the above issues are discussed in detail below.

1.4 Role attribute

Attribute also needs careful treatment. Many of the attributes people think so are not genuine attribute but ***role attribute***. Let us take an example of height. It is a role attribute whose basic attribute is length. Height measured in the direction along the vertical axis higher than the ground, depth, width and distance are role attributes. It is just like a man is called husband when he has got married. The genuine attribute is called basic attribute here. Examples of basic attributes include length, area, mass, temperature, pressure, volt, etc. Role attribute includes height, depth, input pressure, maximum weight, area of cross section, etc. Table 1 summarizes role attributes.

Table 1. Some role attributes.

Role Attribute 2	Role Attribute 1	Basic attribute	value + unit
Age	years alive	time interval	integer + year
Height of a person	height	length	real + m
(Japanese has this	width	length	real + m
concept)	distance	length	real + m
	area of cross section	area	real + m^2
	input flow rate	flow rate	real + m^3/hour

1.5 *Instance-of* vs. *occurrence-of* relations

When we think of an instance, say, Mr. A, we seldom put him into a position of this 3D space with time. When we put him into the real world, we have to specify at least where he is at what time.

However, we seldom care such information when we talk about an instance. This suggests a serious implication that we might need another relation between a class and an individual existing in the world. That is something like *occurrence-of* in addition to *instance-of* relation. I mean, Mr. A's sitting in a chair in his room at 1:30 am of August 13, 2001, is a process and his occurrence in the real world. Mr. A is an *instance-of* human who can participate in many processes as well as its occurrence. In other words, the idea of instance is different from "an instance's being there" which is a real process. Existence of an instance is thought of in an abstract space which is not the same as the real world. A naïve idea of the existence of an instance is not a process.

1.6 Kinds of a *part-of* link[Winston 87]

Part-of relation is extensively used to represent a thing as a whole which is composed of a few parts and is usually transitive. However, not all *part-of* relations are transitive. For example,

<Thum *part-of* Mr. A>
<Mr. A *part-of* Committee>

<Thum Not *part-of* Committee>

This suggests that *part-of* relation has multiple semantics. In fact, some ontology representation languages have two types of *part-of* relations, transitive and intransitive. However, it is not sufficient to deal with the varieties semantics of *part-of* relations. There are at least five kinds of *part-of* relations as shown below.

(1) Functional *part-of,* that is, contribution of the part to the whole is functional.
E.g., wheel is *part-of* bike
(2) Qualification *part-of*, that is, the thing needs to have a qualification or a role to become a part of the whole. E.g., husband *part-of* married-couple
(3) Spatial/temporal relation *part-of*, that is, the thing needs to satisfy spatial/temporal constraints to become a part of the whole. E.g., tree *part-of* forest, mountain *part-of* mountains
(4) Stuff *part-of*, that is, the whole is stuff. E.g., a piece of pie *part-of* pie
(5) Material *part-of*, that is, the thing is materials of the whole. E.g., glass *part-of* cup

The cause of the misunderstanding that *part-of* relations are transitive would be because of *Functional part-of* found in all the artifacts, since transitivity holds in this kind of *part-of* relation. However, some of the combination of different kinds of *part-of* relations shown above would not allow it to hold transitivity.

The above five are different from each other in the semantics of the contribution of parts to the whole, that is, what happens when parts are removed from the whole.
(1) If a wheel is removed from a bike, then bike is not a bike anymore, but the wheel remains a wheel.
(2) If the husband is removed from a married couple, then the married couple collapses, and the man is not a husband any more.
(3) Even if a tree is removed from a forest, the forest remains a forest and the tree remains a tree.
(4) Both a piece of pie and the rest of the pie are pies.
(5) We cannot remove the material without completely destroying the thing.

When making an inference of the result of the removal of a part from the whole, we need these five different kinds of *part-of* relation. Otherwise, the inference results in failure. Nevertheless, we currently have only transitive and intransitive *part-of* relations. In order to model the world correctly, *part-of* relations reflecting the above difference are required.

1.7 Data, information and knowledge

Many people have been discussing the differences between the three close categories, data, information and knowledge for years but the results are not convincing and still controversial. I will give a solution to this by suggesting a better question to ask, since the question people have tried to answer seems to be inappropriate. People have tried to find the differences among them in their properties/characteristics. But such a goal seems not promising. One thing can be any of the three in according to the situation. The fact that "Tokyo is the capital of Japan" can be data when it is stored in a database, can be information when it is told to a person and can be knowledge of a person who knows it. This suggests the thing itself has no clue for such a question.

Data, information and knowledge are ***Roles*** played by a representation(proposition). The following is the explanation to this hypothesis. By "representation", I here mean a proposition coded in a form. The detailed discussion on representation is done in Section 2.

(a) Data: A representation as an operand of "to process"

A representation becomes data when it is or is supposed to be an operand of a verb "*process*". A person name is a bare representation when it is not an operand of "to process" but it is a data when it is processed or it is supposed to be processed.

(b) Information: A representation as an operand of "to inform"

Although it becomes a data when it is processed, there is an exceptional processing, that is, to *inform*. A representation becomes information when it is or is supposed to be an operand of a verb "*inform*". In short, a piece of information is an informed representation. This definition implies that information as a role is person-dependent, while data is person-independent. That is, data is data regardless of if a person is interested in it or not, if a person knows it or not, etc. Data as a role can be considered as objective in this sense. On the other hand, information is different. It is necessarily related to a person who informed it or a person who is informed of it. A representation becomes information when it is supposed to be informed and is independent of its usefulness; e.g., "I have very important information" which apparently assumes to inform someone of it, "the information was useless", etc.

(c) Knowledge: A representation as an operand of "to know"

A representation becomes knowledge when it is or is supposed to be known by a person. I easily expect few people are satisfied with this definition. Possible counterarguments against this definition might be: "No, knowledge is nothing if it is not used by the person who possesses it. It is something which is effectively used to solve a problem. Knowledge needs to be structured and mastered by …." I understand such opinions very well. But, at the same time, I find confusion between a *thing*(representation) and *to know* in such arguments. Such properties of "knowledge" are not possessed by the thing itself but imposed by "to *know*". The following are the counter examples to such opinions: There exist some concepts such as "fragmental knowledge", "heuristic knowledge", etc. which are not structured. We often see a statement like: "Professors are not well qualified in solving real-world problems in spite of that they possess a lot of knowledge", "I find it difficult to master the knowledge", etc. When we are not interested in someone *knows* it or not, neither "Tokyo is the capital of Japan" nor $f=m\alpha$ is knowledge but they are a fact and a relation(or principle), respectively.

The key idea here is decomposition of the properties of the thing into those which are attributed to the thing(representation) itself and those imposed by the verbs such as *to process, to inform* and *to know*. The former includes fact, formula, relation and so on and the latter include what people have thought they need to differentiate between data, information and knowledge. The latter is exactly the same concept of Role. In other words, the thing people call data, information or knowledge itself has no intrinsic properties which make them data, information or knowledge. What we need to ask is neither what is data, what is information, nor what is knowledge, but what is to process, what is to inform and what is to know. As a byproduct of changing the question to ask, the proposal has already solved the problem of differentiation between the three, since it is obvious that how the three actions: to process, to inform and to know are different. The only problem left unsolved is what is to know. I fully agree with those people who think the question of what is to know is extremely

important. However, it is not the main question in ontological engineering.

2. Ontology of Representation

What exist in WWW, that is, what we can reach at URLs are not real entities but representations. Similarly to WWW, there exist quite a few representations in the real world: *novel, poem, painting, music, procedure, symbol, etc*. What is the instance of a representation? How are representations different from real-world objects? These are the questions to ask in this section. To answer them, we need a sophisticated ontology of representation. What are the instances of Procedure, *music, drama, symbol, calligraphy, painting, poem* and *novel*? Some look easy to answer, but others might look difficult. Concerning novel, there are three kinds of candidates for its instance: the book, the sentences and its content. Concerning the procedure, the document of procedure description, the procedure description, its content and its execution action are the candidates. Similarly, a piece of music has the musical score book, the musical score, the sound people hear and the playing action which produces the sound. As the above examples suggest, the source of the difficulty in answering the question is that a representation has several deeply-related concepts such as its embodiment, the mode or the form of representation and its content and in some cases there are two embodiments.

What makes a representation different from other artifacts is that a representation is a content-bearing thing, while the other artifacts are just a thing itself. How do you recognize Beethoven's the 5^{th} as an instance of music? The sound you hear is an instance? The score of Beethoven's the 5^{th} is an instance? Or, what else? What is an instance of a procedure? How do you identify the quick sort algorithm as an instance? What is an instance of a symbol? Is the letter "a" you write is an instance of a letter?

2.1 Two principles

Before going into discussion on what is representation, let us share the following two principles needed to properly understand this issue.

(a) Definition of a class is not a *representation*

An ontology is a theory of concepts which explain the target world. Therefore, it is inherently outside the world and hence the class definition is not an ordinary representation. It is analogous to the fact that any theory cannot include itself as an object to explain by itself. The process of an ontology building is a kind of normal process and the source code representation of the ontology is a kind of representation. However, once the representation is used as an ontology, it goes to the meta-world(outer world) to explain the real world. This separation is critical to avoid the common confusion as follows:

> *A class definition of a concept is a specification of all of its instances and it is a kind of representation. Therefore, the relationship between a piece of music and the sound produced and that between class and its instance are identical. That is, the former is a specification representation and the latter is its realization. Therefore, a piece of music is a class and the sound you hear is its instance.*

Although this line of reasoning looks reasonable at first glance, it is not appropriate according to the above observation. It also leads to a difficulty, that is, it implies a composer designs/generates a class rather than an instance, which contradicts with common sense that assumes man-made thing should be an instance. If we accept the above observation, as will be clarified below, we will have to accept that all the individuals such as a procedure, a piece of music, a plan, a symbol and all the specifications are not instances but classes.

(b) Separation between a concrete design product and what it means.

A common erroneous understanding about a *procedure* is something like the following:

A procedure is a description about how to perform a set of actions/operations. So, an instance of a procedure is equal to its description.

Needless to say, a description is not identical to its meaning. The problem here is specification is inherently descriptive, I mean, specification is inherently related to representation because it is not transferred to other persons without representing it. I suspect English-speaking people might have little difficulty of this kind but Japanese often have confusion between specification and description, since both are representation.

2.2 A conceptual model of representation

Let us go to the discussion on ontology of representation. A representation is composed of two parts, form and content shown below.

-Representation
p/o"form": Representational form
p/o"content": Proposition

where p/o stands for *part-of* relation/slot, "form" slot name and ": Representation" is a class constraint the slot value has to satisfy. Its identity is inherited from the *form* which is usually what people sense its existence. On the other hand, the content is the hidden part and it is a proposition which the author of the representation would like to convey through the representation.

Representational form and Proposition has the conceptual structure shown below.

-Representational form
-Symbol sequence
-Natural language
-Spoken language
-Written language
-Artificial language
-Musical symbol
-Mathematical symbol, etc.
-Speech
-Still image
-Photo
-Hand-written image
-Figures, tables, etc.
-Motion image
-Proposition
-Design Proposition
-Procedure
-Music
-Drama
-Symbol
-Specification
-Product Proposition
-Novel
-Poem
-Painting
-Calligraphy

One of the key issues here is that the content of the symbolic representations are recognized as proposition which has two kinds of proposition as its subclasses: Design proposition and Product proposition. The former works as specification of the production of something. The latter itself is the product. For example, a piece of music composed is a specification of the music sound produced by the music player. Procedure is specification of the valid sequence of actions. An execution of the procedure generates a result(product). *Novel* cannot be specification of anything because it is already a product. A tricky thing is that a design proposition can have a product as well, i.e,, a document of the specification. Musical score, procedure description, etc. are examples of such products.

It is important to clearly distinguish between a representation and a represented thing. Any representation is not embodied unless it becomes a represented thing. A sentence "This is a book" is a representation in the form on natural language(English) and what you see is its printed realization on a sheet of paper which is a represented thing. This model enables to deal with "copy" of representation. The copied stuff is what is on the representation medium. Copying is just a generation of the same representation on a different medium. In fact, a novel, say, Tale of Genji, exists in the form of a book. A book is physically a 3D thing with the logical structure of chapter, section, etc. and with the content composed of a representation in terms of natural language/images. Tale of Genji exists independently of on what medium it is written.

-Represented thing
  p/o"representation": Representation
  p/o"medium": Representational medium
-Representational medium
  -Physical thing
    -paper
    -canvas
  -Electronic thing
    -CD-ROM
    -etc.

It is the time to show what are procedure, music, etc.

Procedure representation
  is-a Representation
  p/o"form": Language
  p/o"content": Procedure
A representation of Quicksort algorithm
  instance-of: Procedure representation
  p/o"form": <C language>[5]
  p/o"content": <Quicksort algorithm>
Musical score
  is-a Representation
  p/o"form": Musical symbol
  p/o"content": Music
A score of symphony the 5^{th}
  instance-of: Musical score
  p/o"form": <A sequence of musical symbols>
  p/o"content": <Symphony the 5^{th}>
A book of musical score of symphony of 5^{th}
  instance-of Represented thing

[5] In the case of instance, the semantic constraint denotes not the class constraint but the value. By <>, I mean it is a generic description of the value which cannot be writen due to space limitation. <C language> denotes a certain program coded in C.

p/o"representation": <A score of symphony the 5th>
p/o"medium": <Paper>

2.3 An ontology of representation

The above models declare that an instance of a procedure is composed of an instance of language description called "form" and an instance of procedure class which is a subclass of *design proposition* called content. It can explain various representations, as instances, of quick sort algorithm by changing the form part which can take not only the computer languages C, Lisp, Java, etc. but also the real code in such a language keeping the same content, Quicksort algorithm. Note here that an instance of a procedure is not a sequence of actions performed according to the procedure description but a specification of valid action sequences. That is, an instance of a procedure is a specification of its content. A specification can virtually exist not as being an embodiment(represented thing), but as the content of representation. In general, all the design *propositions* are specification and its instances are also specification which has no embodiment by itself.

The next problem might be what is the sound(symphony the 5th) people hear? It is an instance of musical sound and is a realization of the symphony the 5th, not an instance of a piece of music. The playing action is an instance of playing action and at the same time realization action of the musical sound. Furthermore, the relation between the musical sound and symphony the 5th is *realization-of* relation.

The above discussion is summarized in Fig. 1 which has four top-level categories such as *action, concrete, proposition(content)* and *representation*. Note that this figure is drawn only for the purpose of making the idea of essentials of representation so that details are omitted or not seriously elaborated. For example, *represented thing, procedure* and many intermediate concepts are omitted. Under the proposition, we have Design proposition and Product proposition. Under the former, we have *music, drama, letter* and *specification*. *Music, drama and letter* are specification as well. The *specification* at the same level is one such as requirement specification which is a specification from the beginning. One of the main points to make here is that *music, procedure, novel*, etc. exist as a proposition in the real world. That is, they cannot exist by themselves, since any proposition is insensible for human. To make it sensible, it has to become a *represented thing*(a book in the case of a *novel*) or to be produced(played in the case of *music* or performed in the case of a *procedure*). Both the producing and performing actions are called here "realization action". The instance of proposition and the sensible thing corresponding to it is linked by *realization-of* link like "the 5th," and its musical sound which is the result of a generation action linked by *generated-by* link. Thick dotted lines indicate the proposition and content of a representation are equivalent. For example, Tale of Genji which is the content part of its sentences and which is an instance of novel are equivalent and hence they linked together by a *equivalent-with* link. In the above, "sentences" should be neither "book" nor sentences written on a sheet of paper. They have to be just "sentences" without any medium which carries the graphical image of the sequence of letters/words. "Book" is not a representation but a physical object which physically consists of pages of papers and logically carries the content of the representation.

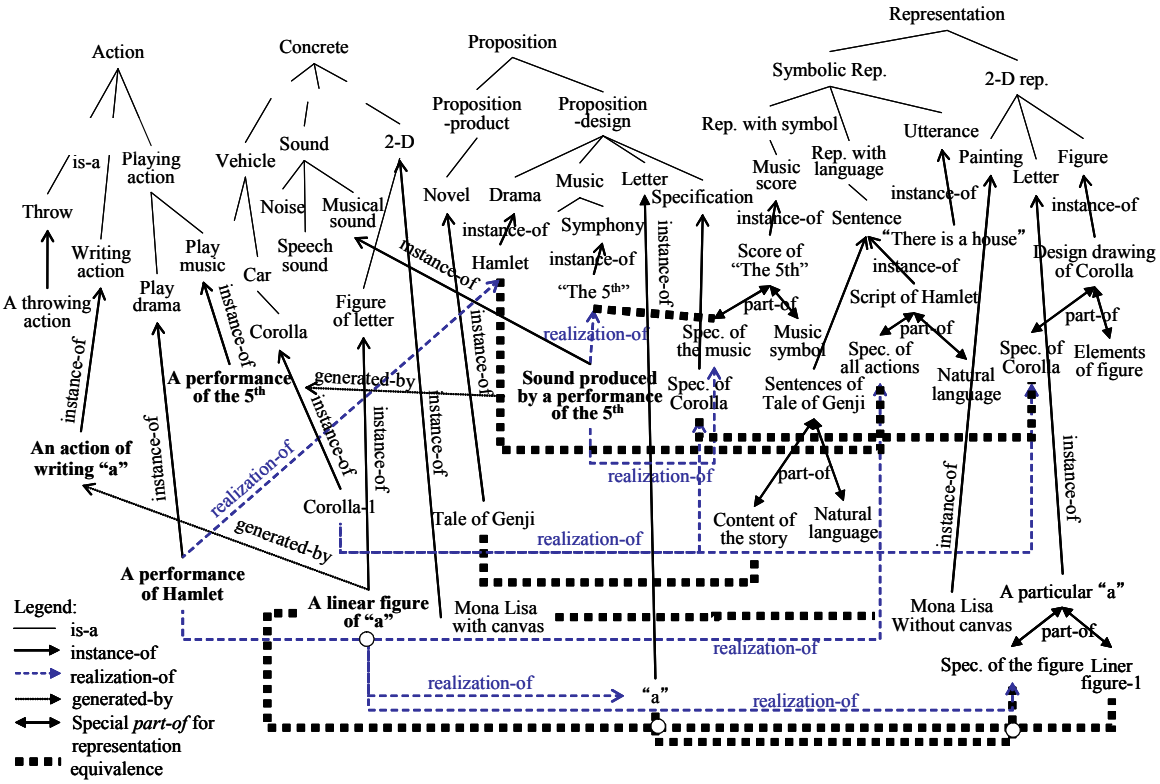

Fig. 1 Ontology of representation.

It is critical to distinguish among proposition(content), representation and form of representation. In fact, although a novel is written in terms of sentences, *novel* is not a subclass of representation. What exists as a subclass of representation are what have the form of representation as its intrinsic property, that is, *sentence, musical score, painting, etc*. The sentences of Tale of Genji are *instance-of* sentence. However, representation and form of representation are different. Concerning a novel, representation is "sentence" which is composed of its content(novel) and "natural language" which is the syntactic part of the sentence, as the form of representation.

One of the most interesting results here is about symbol. Let us take a letter "a" as an example. As you see in Fig. 1, *letter* appears three times in the hierarchy: a two-dimensional drawing under a physical object, specification of its ideal figure as a design proposition and a representation of a particular figure. It implies that for each letter, there exists only one letter in the world as a designed artifact like a piece of music with tremendous amount of realizations. At the same time, a realization of the letter "a" as a representation consists of content and form; the former is specification of the particular figure "a" and the latter is the particular linear figure. What people see when they look at "a" as a real entity in the real world is a two-dimensional linear figure, generated by a writing action, which is equivalent with the form of representation. The key issue concerning letter is that an instance of letter is a specification of its ideal figure. All what people see are realizations of the specification.

Roughly speaking, all design propositions are analogous with each other in their conceptual structure in Fig. 1. For example, *procedure* and *music* are completely analogous. However, music and symbol(letter) has a non-negligible differences: Physical objects related to music are musical sound which is the result obtained by the producing action using the musical score(proposition) as specification and a musical score book which is the design product of the specification itself. On the other hand, those related to a letter is only its linear figure in both of the cases: product and design.

Precisely speaking, the both linear figures are categorically different. One is just a two-dimensional image and the other is a represented thing written on something(paper). *Letter* is special in that its identifier and its entity degenerate into one thing, the figure.

Table 2 shows a summary of what exist concerning representation. Novel has no sensible entity in the real world. It only has Book, etc., a represented thing, as its corresponding sensible entity. The others have two categories of sensible entities. This difference comes from the fact that a thing which is a design proposition has its realization as a product and an additional product as a represented thing of the specification.

Someone might disagree with Table 2 especially on the content of painting. Such a doubt is originated from a deep issue, that is, what is the content. Let us discuss it. By content, we here mean what "consumers" of the representation get from it which is hopefully identical to what the author of the representation wanted to convey to them. First of all, all the representations are consumed by people who appreciate them and get some impression as the result of appreciation. Such kind of content is common to all beings and very subjective. Therefore, they are omitted in the representation ontology. This is why *painting* have no content as a proposition. *Painting* is special in the sense that its main content is the painting itself which is visible like *calligraphy* whose main content is also its visible stuff even if it expresses a word or sentence. It contrasts well with *novel* which has both visible thing(figures of letters of the sentence) and its content as a proposition(the story).

As readers might have already noticed, ontology never defines *music, painting, novel, etc*. but it investigates something like meta-properties of them to differentiate them and organize them in a proper(convincing) taxonomy. It is assumed that people share the meaning of each term. On the basis of the assumption, ontology researchers try to pick up the meta and intrinsic properties avoiding their definition in the computer.

Precisely speaking, sentences have two contents: meaning of the sentence and style(beauty) of the sentences. The discussion thus far only dealt with the former content. To make the discussion more complete, the latter content needs to be incorporated in the theory. The key here is the idea that the content is something generated by people's interpretation. A good writer has his/her own style of sentence which is the heart of writing. This is an object to be learned or mastered by novice writers. In this view, sentence(novel) which is classified in the "Product proposition" becomes to have "Design proposition" as a specification of his/her typical style of writing. Fig. 2, an extension of Fig. 1, shows a model which deals with this issue. The explanation is skipped here because Fig. 2 is rather self-explanatory. This theory also applies to *music* and its performance. A musical sound sequence generated by a performance is a kind of representation. As is discussed above, whether or not a thing has a design proposition depends on the consumer's interpretation and/or author's intention. A musical sound sequence can have one when people see "a style" of performance just like the style of writing letters in *calligraphy*. Our model can thus cope with variation of performance style of players of the same piece of music and it also can deal with many copies of the same performance piece of music by the model of represented thing.

Table 2 Summary of representational concepts.

	Individual in the real world	Representation	Proposition (Content)	Represented thing
Industrial Product	Itself	Its design drawing/specification	Specification of the structure	Printed design drawing/specification
Painting	(as represented thing) Painting	Painting without canvas	None	Painting on a canvas
Music	Sound sequence	Musical score	Specification of the sound sequence	Printed musical score
Procedure	Execution process of procedure	Description of procedure	Specification of the actions/operations	Printed description of procedure
Letter	2-D linear figure	(written) letter	Specification of its ideal figure	Letter written on a sheet of paper
Novel, Poem	None	Sentence	Story(content)	Book, Reading aloud

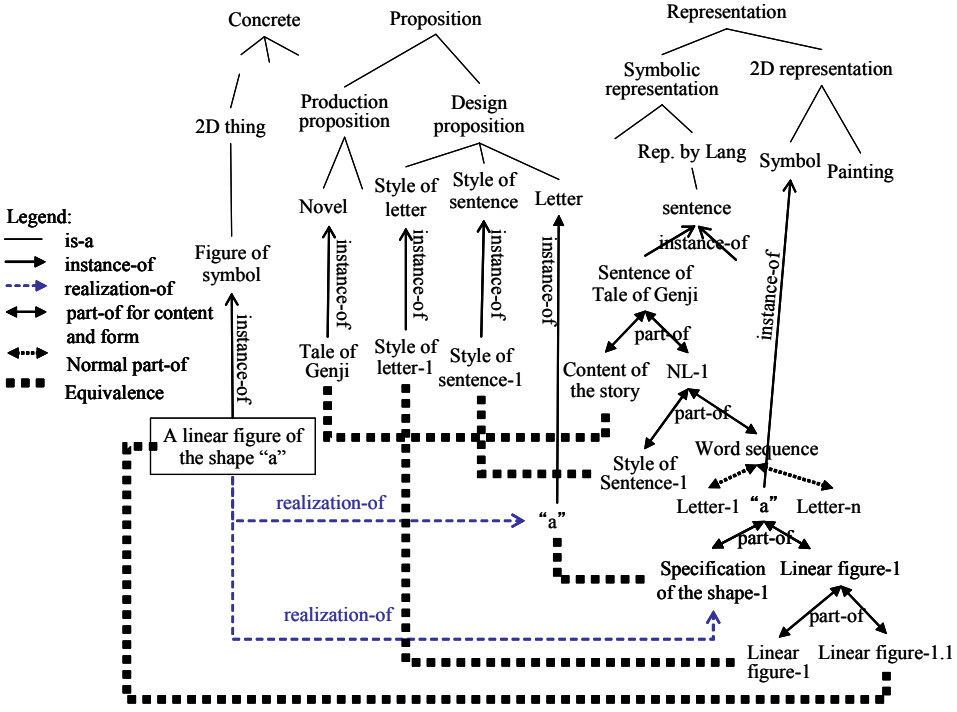

Fig. 2 Ontology of style.

3. Guidelines of ontology building

For some reasons, the detailed guideline for ontology building has been put in Part 2. This section is devoted to make the underlying philosophy explicit to better understanding of it.

In general, domain experts work with strongly domain-specific knowledge which is task-dependent as well, and hence they find a difficulty in coming up with objective and generic knowledge. Furthermore, they even tend to consider such knowledge as useless. This is one of the grounds of the statement that generic knowledge, and hence an ontology is useless. While such a statement sounds acceptable, it would lead us to a dangerous idea, that is, we only need very narrow, domain- and task-specific knowledge which had been the key idea of old-fashioned expert systems. But, it was the main reason why expert systems had difficulties in surviving for longer years. Knowledge should be shared by many people, knowledge has to accumulate in KBs, knowledge should be reused, etc. are our great understanding about knowledge and are the basis of the future research on knowledge processing. I believe that ontology helps people realize such an enterprise. At the same time, a sophisticated methodology for connecting the objective and reusable knowledge(ontology) with domain- and task-specific knowledge is needed.

Articulation of the world is the first key step of ontology development. Therefore, proper identification of classes is critical. It is correct to say that an ontology should not be dependent on its use because it necessarily specifies things in the target world objectively. On the other hand, it is also true that useful ontologies are those designed intended to reflect its purpose. The problem is to find a compromise between the two views on ontology. At least, we can say that an ontology cannot play the role of Messiah if we follow the latter view.

The next issue here is how to find such a compromise. An idea for it is to exploit task ontology, discussed in Part 1, which potentially solves the task-dependency issue by providing the task-dependent roles which help bring domain ontology designed task-independently into the task context and another idea is the use of role concept supported by the proper class identification based on the intrinsic property, discussed in Part 2. It contributes to appropriate organization of domain ontology under a useful guidance by the upper ontology consistently. Let us take an example: woman. Woman is an objective concept, that is, use-independent concept. On the other hand, Nurse is a role played in the hospital context. Wife and mother are roles played in the family context. All the three are played by a woman. When separate ontologies for objective and use-dependent concepts are established, they would be more sharable and reusable. The above philosophy together with the proper use of *is-a* relation and *part-of* relation discussed in 1.2 and 1.6 will contribute to building good ontologies.

4. A success story of ontology research

The last topic is a success story of ontological engineering. Among many possibilities, the authors believe its use for knowledge systematization is one of the most promising [Mizoguchi and Kitamura 2000]. This is indeed a topic of content-oriented research and is not that of a knowledge representation such as production rule, frame or semantic network. Although knowledge representation tells us how to represent knowledge, it is not enough for our purpose, since what is necessary is something we need before the stage of knowledge representation, that is, knowledge organized in an appropriate structure with appropriate vocabulary. This is what the next generation knowledge base building needs, since it should be principled in the sense that it is based on well-structured concept with an explicit conceptualization of the assumptions. This nicely suggests ontological engineering is promising for the purpose of our enterprise. This section presents the work done by the present author and his colleague, Yoshinobu Kitamura[Kitamura, 2002, 2003] on building a framework for engineering knowledge systematization and its deployment into industries.

4.1 Systematization of functional knowledge

While every scientific activity which has been done to date is, of course, a kind of knowledge systematization, it has been mainly done in terms of analytical formulae with analytical/quantitative treatment. As a default, the systematization is intended for human consumption. The knowledge systematization adopts another way, that is, ontological engineering to enable people to build systematized knowledge bases for computer consumption. The philosophy behind the enterprise is that ontological engineering provides us with the basis on which we can build knowledge and with computer-interpretable vocabulary in terms of which we can describe knowledge systematically.

By building a framework for knowledge systematization using ontological engineering, we mean identifying a set of backbone concepts with machine understandable description in terms of which we can describe and organize design knowledge for use across multiple domains. The system of concepts is organized as layered ontologies as shown in Fig. 3.

4.1.2 Functional Ontology and Knowledge Systematization

No one would disagree that the concept of function is an important member of a top-level ontology of design world. One of the key claims of the knowledge systematization is that the concept of function should be defined independently of an object that can possess it and of its realization method. The claim has a strong justification that the concept of a function originally came from the user requirements which is totally object- and behavior-independent, since common people have no knowledge about how to realize their requirements and are interested only in satisfaction of their requirement by a device built. Another justification is reusability of functional knowledge. If functions are defined depending on object or their realization method, few functions are reused in and transferred to different domains. As is well understood, innovative design can be facilitated by flexible application of knowledge or ideas across domains.

4.1.3 Functional representation

Functional representation has been extensively investigated to date [Hubka and Eder 1998, 2001], [Chandrasekaran and Josephson 2000] and a lot of functional representation languages are proposed with sample descriptions of functions of devices. However, because it is not well understood how to organize functional knowledge in what principle in terms of what concepts, most of the representation are ad-hoc and lack generality and consistency, which prevents knowledge from being shared. One of the major causes of the lack of consistency is the difference between the ways of how to capture the target world. For example, let us take the function of a super heater of a power plant, *to heat steam* and that of cam of a cam and shaft pair, *to push up the shaft.* The former is concerned with something that comes in and goes out of the device but the latter with the other device that cannot be either input or output of the device. This clearly shows the fact that there is a difference in how to view a function according to the target object or the domain. The difference will be one of the cause of inconsistency in functional representation and non-interoperability of the knowledge when functional knowledge from different domains is put into a knowledge base.

The above observation shows that we need a framework which provides us with a viewpoint to guide the modelling process of artefacts as well as primitive concepts in terms of which functional knowledge is described in order to come up with consistent and sharable knowledge. However, conventional research of function is not mature enough to propose such a framework and only a few functional concepts are identified to.

4.1.4 Hierarchy of functional knowledge and ontology

Fig. 3 shows a hierarchy of functional knowledge built on top of fundamental ontologies. The lower layer knowledge is in, the more basic. Basically, knowledge in a certain layer is described in terms of the concepts in the lower layer. Top-level ontology defines and provides very basic concepts such as time, state, process and so on. This ontology is under development and not discussed in this article. Extended device ontology is developed to provide a common viewpoint which supports to realize consistent interpretation of artefacts. These two ontologies collectively work as a substrate on which we can build consistent knowledge in upper layers.

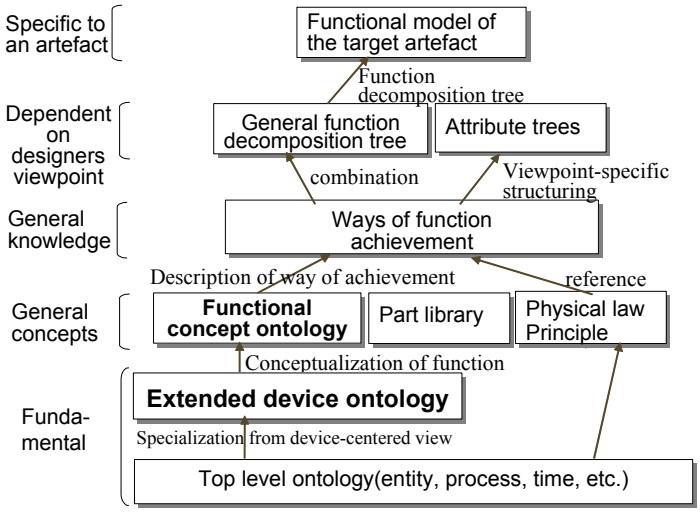

Figure 3. Hierarchy of ontology and knowledge of function

The functional concept ontology specifies functional concepts as an instance of the concept of "*function"* defined in the device ontology. Their definitions scarcely depend on a device, a domain or the way of its implementation so that they are very general and usable in a wide range of areas. Theories and principles of physics and the abstract part library, which are out of the scope of the article, also belong to this class of knowledge called *general concept layer*.

Way of function achievement is knowledge about *how* (in what way) a function is achieved, whereas the functional concept is about *what* the function is going to achieve. Although the way of function achievement way looks similar to functional decomposition knowledge like that discussed in [Pahl and Beitz 1988], the former is much richer than the latter in that it consists of four kinds of hierarchies of different roles and principles [Kitamura and Mizoguchi 2003]. The inherent structure of such knowledge is organized in an *is-a* hierarchy from which the other three structures are derived according to the requirement. The *is-a* structure is carefully designed identifying inherent property of each *way* to make it sharable and applicable across domains. One of the key issues in knowledge organization is clear and consistent differentiation of *is-a relation* from other relations such as *part-of*, *is-achieved-by*, etc. keeping what is the inherent property of the target thing in mind. All the concepts are based on the extended device ontology.

One of the key contributions of the above ontologies to the systemization of functional knowledge is clear distinction between what to achieve(functional concept) and how to achieve(function achievement way). For example, the function of a welding machine is thought to weld, but it is not correct. To weld is a composite concept of to join(what to achieve) and the fusion way(how to achieve). The separation gives us a very generic functional concept, to join which is applicable to many target devices in other domains.

4.2 Deployment into the production division of an industry

The ontology and the systematization framework of functional knowledge are currently being deployed at production systems division of Sumitomo Electric Industries Ltd., Japan, for sharing functional knowledge of devices used in the daily activities among engineers in the division. The test use and the deployment was started in May, 2001 and December, 2002, respectively They have described about 103 functional models of production machines to date. Currently, about 50 people in two factories use the framework in daily work. As an example, Fig. 4 shows the function

decomposition tree of a wire saw for cutting ingots.

Engineers in the production systems division have been suffering from the difficulty in sharing and reusing knowledge among engineers in charge of different devices for long years. They have been regularly writing a technical report for design review, maintenance report, etc. and have accumulated a lot. Unfortunately, however, it has been difficult for them to understand a report written by other engineers who are in charge of different devices, and hence few of the reports or documents are reused. The reasons include:

- Descriptions are specific to the target objects
- Knowledge in such documents s is task-specific
- Vocabulary is not consistent or common
- Much knowledge is left implicit
- To retrieve appropriate report(knowledge) is hard

These are caused by deeper causes:

- There is no principle for representing and organizing functional knowledge
- Every engineer has his/her own viewpoint and there is no common way of how to view a device
- There is no common vocabulary for representing function
- There is no guideline for representing functional knowledge with little domain-dependence.

Our functional ontology and the framework for knowledge systematization is a solution to overcoming all the difficulties. In fact, engineers in the production systems division liked the framework very much and are happy to use it to represent their knowledge about devices they take care of. The system we built (named SOFAST$^{(R)}$) in this deployment is a server of functional models and function achievement way knowledge and clients for writing, storing and retrieving them through intranet.

Basically, for a target production facility (in general, it can be a product also), the usage of our framework is categorized into (1) to communicate with other designers about the target facility using its (general) function decomposition tree, (2) to explore causes of a problem of the facility using its function decomposition tree, and (3) to redesign (improve) the target facility using its function decomposition tree and general functional way knowledge. The following give summary of

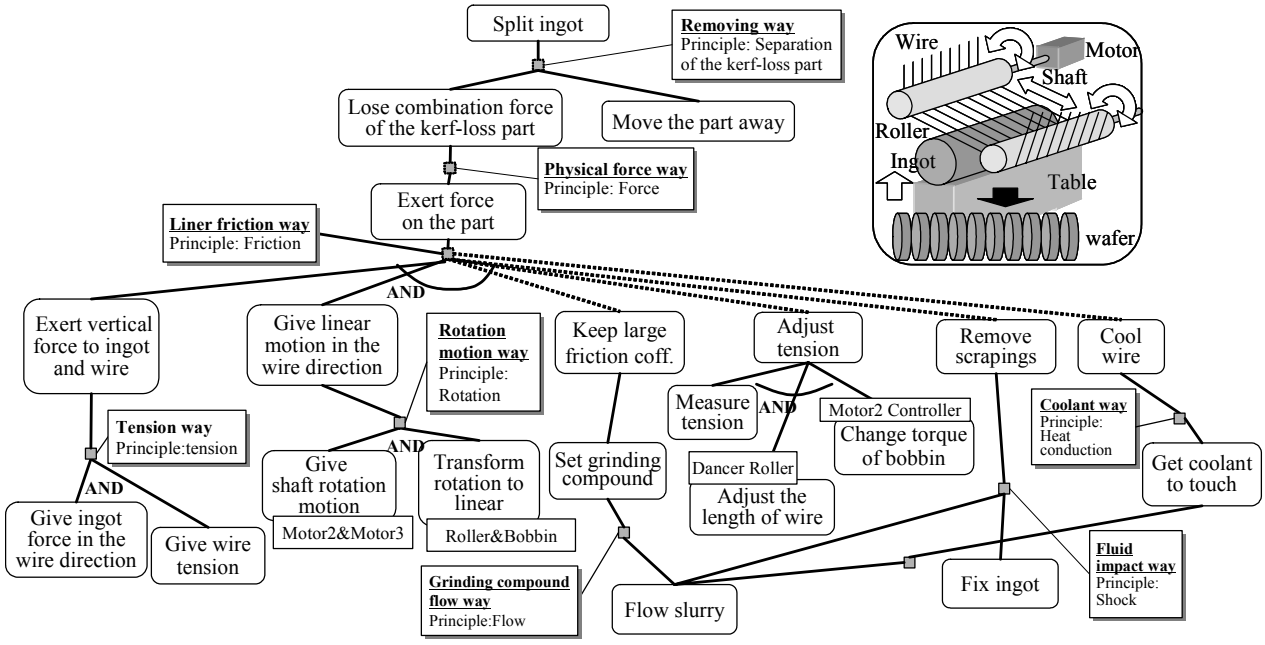

Figure 4. A function decomposition tree of a wire-saw for slicing ingots (portion).

remarkable results in each type of usage in the deployment.

As one of the first usage, the models of ways of function achievement were used as knowledge media for collaborative work by people having different viewpoints such as manufacturing engineers, manufacturing equipment engineers, equipment operators and equipment maintainers. Although mutual understanding and collaboration among them was strongly required, it never happened before. The use of the framework, however, enabled them understand and collaborate with each other in a facility improvement project. It turned out that the framework worked as a common vocabulary which lacked before.

In design review activities, the general function decomposition trees are used as required documents (i.e., the designer must submit a tree of the target device) for discussion. As the result, the times of the design reviews has been reduced to one third.

As one of the second usage, a designer was not able to solve a problem of low quality of semiconductor wafers after 4-month investigation. By exploring causes of the problem in the model of ways of function achievement with a clear description of physical principles, he found a solution for the problem within 3 weeks. The reasons of this success can be considered as follows;

- To write a function decomposition tree of the target machine with explicit physical principles makes the designer's understanding clearer.
- The micro-macro hierarchy of the function decomposition tree enables the designer to explore the possible causes of the problem for each function systematically. The fault tree analysis (FTA) tends to be difficult to enumerate all possible causes without clear understanding of function structures.

As one of the last usage, a feasible improvement of the wire-saw was found from the knowledge-base by adopting the way of using magnetic fluid for controlling tension of the wire. This can be done by applying a way originating from the textile industry to the semiconductor industry. This indicates feasibility of our framework for general functional knowledge.

The success factors include:

- Extended device ontology enables users to be consistent in interpreting how a device works.
- Clear distinction between functional concept (what to achieve) and way (how to achieve) makes the knowledge highly domain-independent
- Functional concept ontology provides a rich set of well-defined functional terms

Clear distinction between a general-specific hierarchy(*is-a* tree) and a whole-part hierarchy (*is-achieved-by* tree) enables to have consistent descriptions of functional decomposition trees and *is-a* hierarchies of ways of function achievement. This avoids the confusion between the two which has occurred very often. SOFAST$^{(R)}$ is currently being used by 13 companies which belong to the SOFAST$^{(R)}$ user's group established in April, 2003.

5. Future of ontological engineering

The author believes that ontological engineering is one of the promising directions AI should go to according the following reasons:

(1) Ontology instead of knowledge
Knowledge is domain-dependent, and hence knowledge engineering which directly investigates such knowledge has been suffering from a rather serious difficulty caused by its specificity and diversity. However, ontology is different. In the ontology research we investigate knowledge in terms of its origin and elements from which knowledge is constructed. An *is-a* hierarchy of concepts and decomposability of knowledge are exploited to deeply investigate primitives of knowledge as well as background theories of knowledge which enables us to avoid the difficulties knowledge engineering has faced with.

(2) Ontological engineering is not yet another application-oriented research
While ontological engineering deals with domain-specific knowledge in addition to very generic concepts like top-level categories, it tries to establish theories and technology for "accumulating" knowledge within reasonable size of stratified domains utilizing *is-a* hierarchy. It is such a branch of AI research that investigates basic theories and technology to treat real world knowledge and is such an enterprise that denies the simple dichotomy of AI research: "Basic research" and "Application research".
(3) Content-oriented research
Content-oriented research is an attempt of knowledge sharing with humans and computers. The dichotomy of general theories of a vessel which can be easily formalized and "domain-specific knowledge" which lacks generality has been widely accepted in AI community. Also, an idea that there cannot be any theory about "Content" has been accepted. The objective of Content-oriented research is to get rid of such a misleading understanding about AI research and to provide effective theories and techniques enabling "knowledge accumulation" and "knowledge interoperability" which do play critical roles in the next generation knowledge processing.

While each branch of sciences has established their own knowledge respectively, computer science has pursued domain-independent theories and techniques treating "information" and "data" obtained by abstracting things existing in all the domains. It is thanks to appropriate abstraction and careful consideration about them. Similarly, content-oriented research tries to exploit abstraction and decomposability of knowledge providing sophisticated guidelines and tools for understanding and model building of the target world through ontological engineering in a similar way as taken in Philosophy.

6. Concluding remarks

We have discussed ontological engineering thus far. I have stressed the underlying philosophy of the ontological engineering which has to be a successor of knowledge engineering. Knowledge processing research has to be promoted further than it is now. It is something different from the traditional knowledge base research. It is something different from the knowledge-based problem solving. Ontology gives you:

(a) A common vocabulary,
(b) Well-justified data structure,
(c) Explication of what has been left implicit,
(d) Semantic interoperability,
(e) Explication of design rationale of the model/knowledge base,
(f) Systematization of knowledge,
(g) Meta-model function, and
(h) Theory of content.

Ontology engineering has already usable tools. It is not an abstract theory. There already exist usable ontologies in various domains and a few success stories. I would be more than happy if this tutorial might contribute to the promotion of ontological engineering.

[Acknowledgement] The author is grateful to Prof. Toyoaki Nishida for his valuable comments on the draft of this article.

[References]

[3Dvs4D] http://ontology.teknowledge.com:8080/rsigma/dialog-3d-4d.html

[Guarino 02] Guarino, N. and C. Welty, Evaluating ontological decisions with OntoClean, Communications of the ACM, 2(45) (2002) 61-65.

[Guarino 98] Guarino, Nicloa, Some Ontological Principles for Designing Upper Level Lexical Resources, Proceedings of First International Conference on Language Resources and Evaluation. ELRA - European Language Resources Association, Granada, Spain: 527-534, 1998.

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

[Hubka, 98] Hubka, V., Eder, W. E., 1998, *Theory of Technical Systems*, (Berlin: Springer-Verlag).

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

[Kitamura, 02] Y. Kitamura, T. Sano, K. Namba and R. Mizoguchi, A Functional Concept Ontology and Its Application to Automatic Identification of Functional Structures, Advanced Engineering Informatics, 16, 2, 145-163 (2002).

[Kitamura, 03] Kitamura, Y., and Mizoguchi, R., 2003, Ontology-based description of functional design knowledge and its use in a functional way server, *Expert Systems with Application*, 24(2), 153-166.

[Mizoguchi, 00] Mizoguchi, R. and Kitamura, Y., 2000, Foundation of Knowledge Systematization: Role of Ontological Engineering, *Industrial Knowledge Management - A Micro Level Approach*, Rajkumar Roy Ed., Chapter 1, 17-36, (Springer-Verlag)

[OpenCyc] http://www.opencyc.org/

[Pahl and Beitz 1988] Pahl, G., and Beitz, W., Engineering design - a systematic approach (The Design Council) 1988.

[Winston 87] Winston, M.E., R.Chaffin and D.Herrmann: A taxonomy of part-whole relations, Cognitive Science, 11, 417-444, 1987.