Draft as of December 16, 2010

YAMATO: Yet Another More Advanced Top-level Ontology

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

Abstract

Upper ontology plays a critical role in ontology development by giving developers a guideline of how to view the target domain. Although some upper ontologies such as DOLCE, BFO, GFO, SUMO, CYC, etc. are already developed and extensively used, a careful examination of them reveals some room for improvement in a couple of respects. This paper discusses YAMATO[1]: Yet Another More Advanced Top-level Ontology which has been developed intended to cover three features in Quality description, Representation and Process/Event, respectively, in a better way than existing ontologies.

Keywords: Upper ontology, Ontological engineering, quality and quantity, informational objects

1 Introduction

Upper ontology is the key of ontology engineering. It plays a critical role in ontology development by giving developers a guideline of how to view the target domain. There already exist beautifully designed upper ontologies such as DOLCE(Guarino 2010), BFO(Smith 2010), GFO(Herre 2010), SUMO(Pie 2010), CYC(Lenat 2010), etc. It seems there is no need to develop yet another upper ontology. However, careful examination of them reveals some room for improvement in three respects: (1) Quality and quantity, (2) Ontology of representation, and (3) Distinction between processes and events.

First, quality and quantity need more careful investigation. One of the most sophisticated property ontologies is found in DOLCE. It specifies value space for each quality type under the name of Quality space(region). A remarkable feature of DOLCE's conceptualization about it is the clear distinction between the quality an object possesses and the quality value(quale/qualia) itself. This feature is good to support description using <Entity, Attribute Value> triple[2] (referred to as <E, A, V> triple hereafter) which has been used in knowledge representation in AI. However, there are other ways of quality description such as <patient_1, diarrhea>, <cupper, conductive>, and <rose_1, redness, faint> which seem not to be covered by DOLCE's quality ontology. On the other hand, BFO's quality ontology has been obtained under the principle of simple and capturing essentials of quality and covers only <Entity, Property> type description. As far as I know, there exist three kinds of data description such as <E, A, V>, <E, P> and <E, P, V> (Egana 2008). However, there is no ontology which supports all these three kinds of quality descriptions. Even worse, no existing upper ontology distinguishes between quality in reality and quality description, which causes some confusion when talking about quality and quantity. The author believes quality ontology should contribute to facilitation of data interoperability as well as capturing quality ontologically.

The second aspect is informational object. There are a lot of informational objects in the world: documents, books, music, Web pages, etc. They are different from ordinary things in that they have content, that is, they are "content-bearing things", while cars, tables, dogs, trees are not. Because representation-related things are classified as semiotics, it has not been discussed in the philosophical ontology very well. However, we need an ontology of informational object from the ontological engineering point of view. Ontology of informational object is discussed extensively in DOLCE D&S(Gangemi 2003) and SUMO. Although these ontologies contribute to better understanding of informational object, there is some room for improvement. One aspect is the clear differentiation between representation and representing things together with its elaboration. For example, while a book as representing thing is, say, a book you buy at a book store, a book as representation is what the author wrote, that is, something left after subtracting sheets of paper from the book you buy. It should also be the book in a CD-ROM. In addition, I believe finer-granular types for talking about informational object are needed. DOLCE D&S has rich conceptualization about representation. It incorporates *situation* and *concept* to realize very comprehensive ontology of representation. Although it is sophisticated, the framework introduced is a bit too large to deal with ordinary informational object such as sentence, music, plan, etc. Especially, *situation* is not necessary to talk about representation. What is more problematic is introduction of definition of a concept into ontology because as discussed in 4.1, it violates the principle "Theory should not talk about itself in it".

The third aspect is most fundamental and is related to differentiation among objects, processes and events. There are several views about them. The typical ones are 3D view and 4D views. BFO nicely reconciles the two views by introducing SNAP and SPAN ontologies. GFO is based on 4D view. DOLCE is based on 3D view. Although discussion on difference between objects, processes and events are made in these ontologies, what can change and what is an object at all are not very clear. I see both 3D and 4D views of the world are incorrect. I could say the idea of neither "objects prior to processes" nor "processes prior to

[1] YAMATO is the next version of YATO and the name was suggested by Peter Simons who loves the Japanese battle ship "Yamato". Yamato is also the name of the oldest Japanese government.

[2] In AI, "Object" is used instead of "Entity", but to extend the target things to be described to occurrent, I use "Entity".

objects" is correct, but both are mutually dependent. Furthermore, processes can change, though events cannot(Galton 2009).

This paper discusses a new upper ontology called YAMATO: Yet Another More Advanced Top-level Ontology developed to cover the above three issues which the existing upper ontologies fail to explain satisfactorily. YAMATO is not fully axiomatized yet, though it should be done in the near future. It is implemented in Hozo and OWL and open for use at http://www.ei.sanken.osaka-u.ac.jp/hozo/onto_library/upperOnto.htm. Some might see YAMATO is too large and complex. I believe such a criticism does not apply to YAMATO which tries to reveal secrets of reality as fine as possible under the condition of maximization of its utility in practice. Many of the existing upper ontologies are too simple to explain the reality and to guide domain people to build their ontologies. What they need are not only distinction between objects and qualities but also that between quality and quantity and that between quality and description of quality, not only that between objects and representation but also that between a copy of book and book and that between a novel and a musical score, and not only that between process and event but also that between a pulse and a sequence of pulses and that between to grow and to cut, etc. They also need to know how much similar a procedure and a piece of music are in what sense, why events cannot change while processes can, etc. YAMATO is designed to answer all the above questions and the like.

2 Overview of YAMATO

2.1 Basic distinctions

Ontology design is a kind of design activity which necessarily has some design rationale that largely influences the resulting ontology. In the case of BFO, it is "representation of reality", for example. In other words, any ontology cannot be free from some assumption and/or designer's standpoint. The standpoint taken in YAMATO consists of Newtonian world point of view and 3D-like 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 in a mutually-dependent manner which is discussed in detail in (Galton 2009). Furthermore, we do not go into micro-level world in which we need to talk about atoms and/or particles. By water, I mean amount of water rather than H_2O, since we need to say water is dissective. If we take H_2O into account, it is apparently not dissective. This premise is necessary to build a consistent ontology. The following list includes basic distinctions made in YAMATO.

(1) Substrate and entity
Space and time are indispensable for things to exist in the world, while these two can exist independently of entities. Such independence is essential and differentiates the two from entities that inherently need these two to exist in the real world. Matters are less basic than space and time, but it still is very substrate-like because every physical individual is made of/from matter.

(2) Entity and property
Any entity cannot exist without any property, e.g., any physical object has necessarily a couple of properties (color, mass, size, etc.). At the same time, any property cannot exist alone. It necessarily needs an individual to inhere in. Thus, both an individual and a property are inherently dependent on each other and cannot be separated. Such a deep mutual independence is an essential structure of being: objects vs. processes and matter vs. physical objects are other examples, though kinds of dependency are different from each other.

(3) Physical and abstract
We define a physical thing as something which needs time and space to exist, and introduce semi-abstract which needs only time to exist. Needless to say, there is nothing which requires only space to exist. Abstract things are defined as things that need neither time nor space.

(4) Continuant(Object) vs. Occurrent(Process[3])
This is one of the most controversial issues and has a long history of debate. It is sometimes called 3D model vs. 4D model. Common sense is based on 3D model which consists of the 3D Euclidean space with absolute time. YAMATO is based on a solid theory of objects, processes and events, and it deals with them of equal importance (Galton 2009). Furthermore, it clearly identifies events are made of processes and while processes can change, events cannot. We do not use the term "perdurant" for explaining processes because it blocks to talk about the difference between processes and events.

(5) Entity and relation
Relation is sometimes 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 that is time-dependent and hence cannot be abstract. Although it is intangible, it exists in the time frame of the real world. Friendship between persons, marital relation, part-whole relation, etc. exist in the world. People sometimes confuse relation as a formalism with relation as an existing thing. Typical examples are *action* and *attribute* that 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 ontologically. They are intrinsically entities included in an ontology.

(6) Informational object vs. non-informational object (Representation and non-representation)
Informational object and symbols are usually dealt with in semiotics rather than in ontology. However, from the real-world modeling point of view, we need to deal with informational object in our ontology, since there apparently exist music, novels, texts, symbols and so on in the real world. Informational object and non-informational object (object, process,

[3] Before coming to the discussion on the difference between processes and events, by the term process, we mean "occurrent".

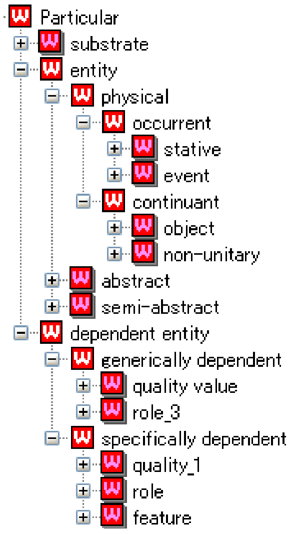


Fig. 1 Top-level categories.

relation, attribute, etc.) are very different from each other. For the informational object, it is not easy to identify what their instances are. For example, what an instance of a piece of music is, what an algorithm is, how both are similar ontologically, etc. need some in-depth consideration.

2.2 Top-level categories

YAMATO adopts single inheritance to make the taxonomic structure clean like BFO and DOLCE. Using *is-a* relation, it is realized in YAMATO that the type hierarchy is made only when the lower type inherits its intrinsic properties from the super types. Many of the multiple context-dependencies are covered using roles (Mizoguchi 2007). For the cases where genuine multiple inheritance is necessary, Hozo (Kozaki 2010) prepares *IS-A* relation which is nothing to do with identity problem of instances but only with property inheritance. It is allowed to use only when *is-a* relation already exists between the two types of interest. The early version of YAMATO has been designed under a considerable influence of Guarino's upper ontology (Guarino 1995). *is-a* relation in YAMATO is something more than usual property inheritance. That is, it implies inheritance of identity criterion, so that when an instance of such a class loses the essential property, then it stops being an instance of all its super classes. On the other hand, whatever happen with such properties of an instance that are inherited through *IS-A* relation, no influence occurs concerning the identity of the instance. Note that a type is not a property in YAMATO. Therefore, *human* is a type and is not dealt with as a property. Instead, *human* type has properties/qualities such as height, weight, age, etc[4].

Fig. 1 shows the top-level categories of YAMATO. At this level, YAMATO has little significant difference from other existing ontologies. A more finer-grained view of YAMATO is shown in Fig. 2 that reveals its features. It shows a clear distinction between *process(a)* and *event(b)*.

[4] Terminological issues about property and quality is discussed in 3.2.6 extensively.

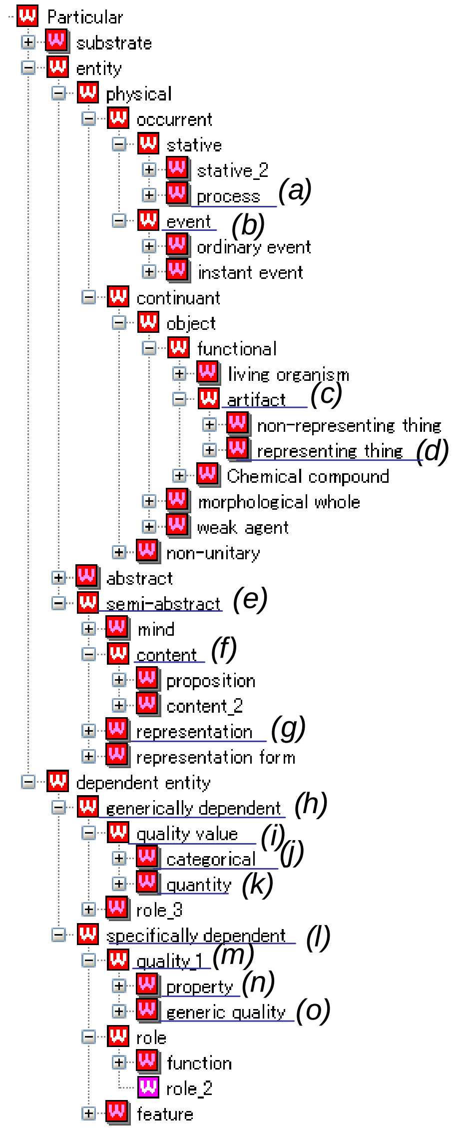


Fig. 2 Detail version of top-level categories.

Concerning ontology of representation, *representing thing(d)* is classified under *artifact(c)*, while *content(f)* and *representation(g)* are classified under *semi-abstract(e)* because they need only time to exist. *Dependent entity* is classified into *generically dependent(h)* and *specifically dependent(l)*. The former has *quality value(i)* as its subclass which is divided into *categorical(j)* and *quantity(k)* and the latter has *quality_1(m)* as its subclass which is divided into and *property(n)* and *generic quality(o)*. The details are discussed in the following sections.

3 Quality and Quantity

3.1 Background

e-Science needs data exchange world-wide. Especially, in biology and bio-informatics, data exchange has been intensively conducted through global collaboration in the daily activities. In order to make it smooth to exchange scientific data, the way of description must be compatible with each other. Description of data/objects is usually done by determining attribute values. However, quality description is not well-understood by practitioners. Even

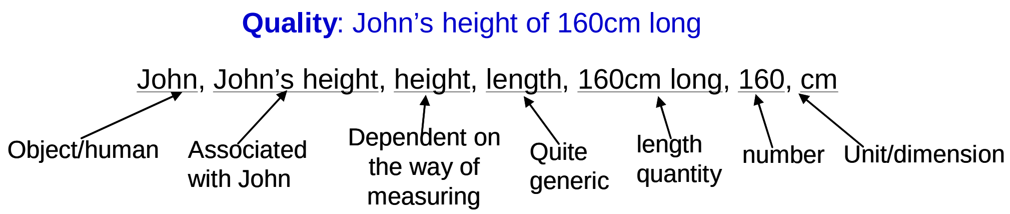


Fig. 3 An example of quality.

professionals seem not to agree on the common way of quality description of objects. This is why in OWL best practice WG, there proposed several patterns which would be useful in many cases of quality description.

As is discussed in (Egana 2008), there are at least three ways of quality description recommended by BFO, DOLCE and Galen(Alan 2010). BFO recommends <Entity, Property> (e.g., <John, tall>) formalism, while DOLCE <Entity, Attribute, Value>(e.g., <John, height, 180cm>) and Galen <Entity, Property, Value>(e.g., <John, tallness, large>[5]). The problem is two fold: (a) one way of representation is not enough for real world problems because there exist multiple kinds of quality descriptions in the real data and (b) there is no explicit model of how these three are different from and interrelated with each other. The ontology of Quality and Quantity in YAMATO tries to give fundamental conceptualization of those existing descriptions in one ontology. To do so, YAMATO distinguishes between quality in reality and quality description and has a type hierarchy which reflects existing notions used in quality descriptions.

Another problem is that there exist several ways of implementation of a formulation. In OWL, attribute is usually represented in *object property*. That is, (Entity --attribute--> value) in which attribute is not dealt with as an entity but as a relation. This implementation fails to appropriately represent the growing boy discussed below. It suggests that not only ontology but tools supporting the ontological formulation are also key factors of successful data description. Note here that Hozo(Kozaki 2010), the tool our group has been developing, is compliant with YAMATO proposed in this paper.

3.2 Quality

3.2.1 The underlying philosophy

One of the contributions of our ideas is clear distinction between attribute and attribute values. As far as I know, there are three dependent entities related to the term *attribute*: (1) attribute as a type that plays as dimension, (2) attribute as an instance and (3) attribute value that might correspond to quantity. However, some ontologies do not distinguish between (2) and (3). In this respect, DOLCE nicely differentiates instance of attribute from attribute value by introducing quality for attribute as an instance and quale/qualia for attribute value. This differentiation enables us to capture a thing, holding a quantity(qualia), that can change keeping its identity as we discuss it below. In other words, any quantity exist only one in the world independently of how many things have those quantities as their values. We build an ontology for each of attribute(*generic quality*) and attribute value(quantity) intended to describe things in terms of the combination of those dependent entities. We discuss each of the two in turn.

3.2.2 An example

See Fig. 3 that shows an example of quality which I believe many people can agree that it is an example of quality. The issue here is how to wisely model this quality. Fig. 3 also shows types and individuals underlying the quality. Among them, "John", "160" and "cm" have no issue to discuss. Let us briefly investigate characteristics of the others. Firstly, "John's height" is something associated with John and exists at the instance level. "Height" is generic in the sense that the entity which it is associated with is unspecified, but is more specific than "length" in the sense that it is dependent on what to measure and how. This suggests the notion of "quality role" played by length, which is an interesting topic, but it will be discussed later. The "length" is quite generic because it can be height, depth, distance, etc. according to the context and exists at the class level. Finally, "160cm long" seems to be quantity and exists at the instance level. In summary, issues include (1) if "160cm long" is a quality or a quantity, (2) if an instance of a quality is a quantity or not, (3) what is "John's height"?, (4) what is "height"?, and (5) how is "length" different from "height"? The following subsections are devoted to answer these questions.

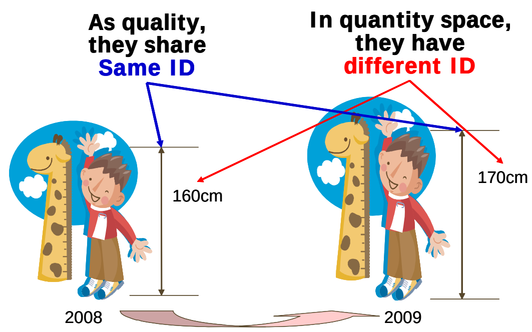


Fig. 4 A growing boy and his height quality.

3.2.3 On change

See Fig. 4 that shows a boy named John who grows. His height was 160cm in 2008 and 170cm in 2009. Ontologically, in order to talk about *change* properly, we need a non-changing thing keeping its identity during the change process, otherwise, there cannot exist a change but just a difference. In the case of growing John, John and his height must keep their identities. That is, "John's height"

[5] A better example would be <John, diarrhea, severe>.

must have the same identity at the times of 2008 and 2009. The values, on the other hand, are 160cm in 2008 and 170cm in 2009, respectively, and they have their own different identity. This analysis reveals that "John's height" has something having its identity and it is generic in the sense that its possible values are indefinite.

3.2.4 Quality role

We here discuss the notion of *quality role*. Many of the qualities people know are not basic quality[6] but *quality role*. Let us take an example of height. It is a quality role whose basic quality is length. In the role theory (Mizoguchi 07), it can be said that "height role is played by length in the context of a human body". Height is what is measured in the direction roughly along the vertical axis from the ground level to the top. Similarly, depth, width and distance are quality roles played by length as well. It is just like a man plays a husband role when he has got married and like a human plays a teacher role in the context of a school. Let us call quality such as length as *basic quality* temporarily. Examples of basic qualities include length, area, mass, temperature, flow rate, voltage, etc. Basic qualities are context-independent and intrinsically represent kinds of quality, and roughly equal to physical dimensions. Quality roles include height, depth, input flow rate, maximum weight, area of cross section, etc. Table 1 shows examples of quality roles.

Table 1 Some quality roles.

Quality role	Basic quality	Quantity (number + unit)
age	years	integer + year
height	length	real + m
width	length	real + m
distance	length	real + m
area of cross section	area	real + m^2
input flow rate	flow rate	real + m^3/hour

The notion of *quality role* satisfies context-dependence which is one of the most important properties of roles. In the case of teacher role, a school is its context in which teacher role is defined. In the case of height, which is a quality role played by length, on the other hand, its context would be human body, building, etc. with the measuring manner from the ground level to its top.

3.2.5 Summary of the discussion on quality

A quality must be a certain dependent property possessed by an entity and keeps its identity while its value changes. In the case of growing John, this understanding suggests that not "John's height of 160cm long" but "John's height" should be taken as a ***quality*** as shown in Fig. 5. As is already clear, quality roles are necessarily associated with a particular entity which is the context of them at the instance level. The fact satisfies the requirement of quality. Therefore, *quality roles* are what should be called ***quality***. Then, "John's height of 160cm long" would be called ***quality instance***[7] because it is a realization of quality role "John's height" at a certain time. In terms of role theory, "John's height of 160cm long" is called a *role holder*, which will be explained later, which means a length playing the "John's height" role. Then, "height" is called ***quality role type*** which has "John's height" as its instance and "length" should be called ***generic quality type*** because it is not associated with any entity yet and specifies what kind of quality there exist. It was called basic quality in the above. The results are summarized in Fig. 5. Definitions of these basic terms are shown in the following:

1. ***Generic quality type:*** The most generic property which is not yet associated with any particular context and can play *quality role*. It represents kinds of property at the class level and its instances represent concrete values corresponding to quantity to play *quality role*: e.g., length, weight, mass, color, area, flow rate, etc.
2. ***Quality role type:*** It is a type of property associated with a context, and hence is equal to *quality role* at the class level. Its instance is played by an instance of *generic quality type*: e.g., height, depth, (something's) length, (someone's) weight, mass of cross section, input voltage, etc.
3. ***Quality:*** It is a property associated with an individual entity and is a generic name to denote each instance of ***quality role type***: e.g., John's height, Tom's weight, area of the cross section of this pipe, length of this pen, etc.
4. ***Quality instance:*** A realization of ***quality***: e.g., John's height of 160cm long, Tom's weight of 50kg, etc.

These types and their relations are depicted in Fig. 6.

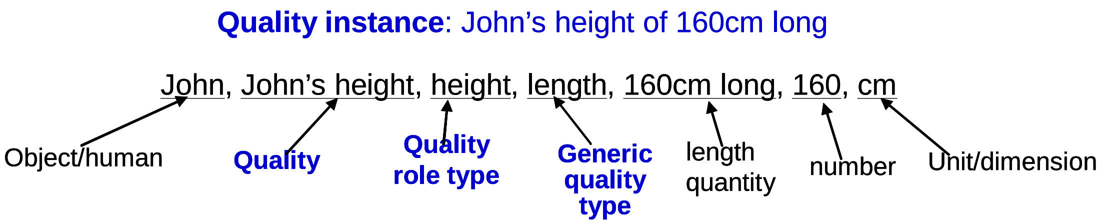


Fig. 5 Informal definitions of quality-related terms.

[6] This term is temporary and will be renamed as *generic quality type* later.

[7] Although *quality realization* might be a better name, I use *quality instance* to minimize peculiarity.

Examples of relations between these types are shown below for further clarification:

<*“John’s height of 160cm long” realization-of “John’s height”*>

defined as shown in Fig. 7. In the case of a pen, its *length role* is played by *length*. Note here that the term “length” appears twice in the role name and in the role player name, which is the source of confusion. The former *length* is a

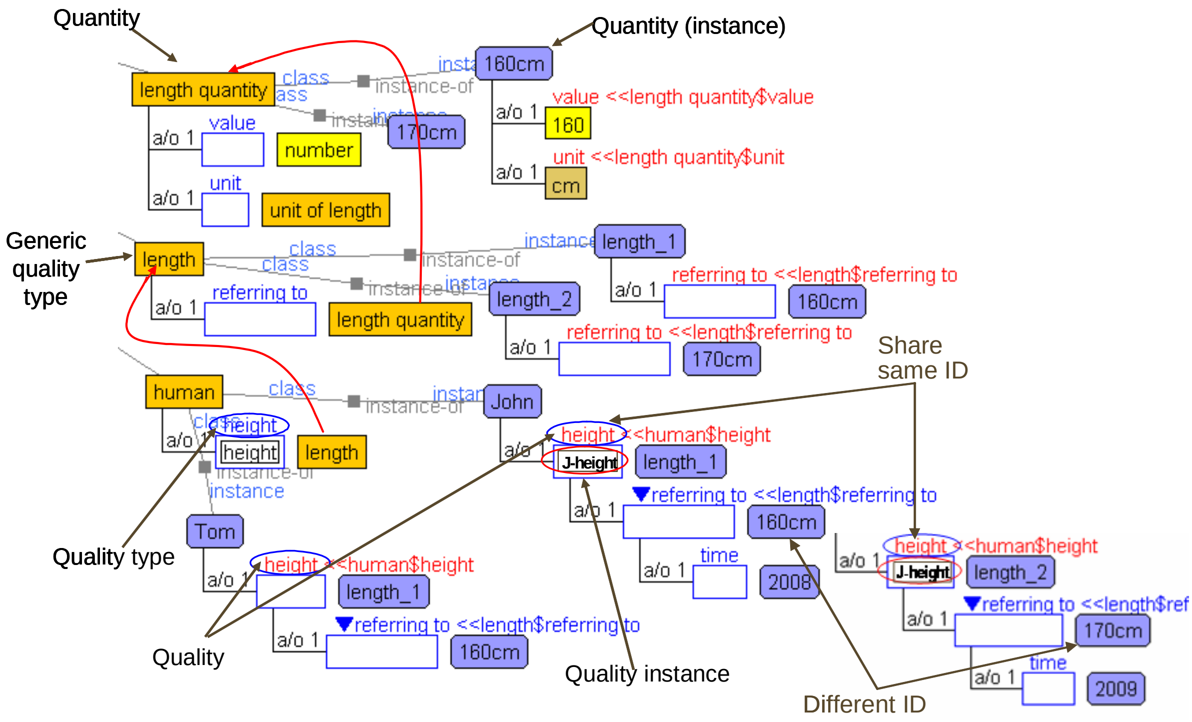


Fig. 6 Quality ontology and its use in Hozo (Rectangles with round corners denote instances).

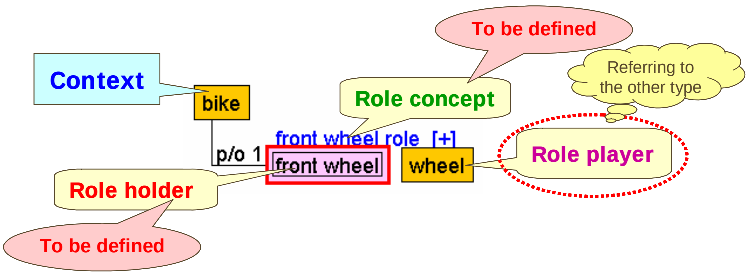


Fig. 7 Hozo legend of role definition [Mizoguchi 2007].

<*“John’s height” instance-of height*>
<*”height” (role) is-a* ***quality role type***>
<*“height” (role) is-played-by length*>
<*quality role type* *is-played-by* ***generic quality type*** >
<*length is-a* ***generic quality type***>
<*160cm long instance-of length quantity*>
<*length quantity is-a* ***quantity***>

Fig. 6 also shows two height quality values of John at times of 2008 and 2009. Although two height quality roles of John are shown, they share the same identity, while the two length quantities, 160cm and 170cm have their own identities.

I suspect readers would need an explanation about examples of “(something’s) length” and “(someone’s) weight”. Before explaining them, however, I need to explain role theory using Hozo way of representation of roles (Mizoguchi 2007). Fig. 7 shows Hozo legend of role definition in which *front wheel role* is defined in the context of bike. It says that a *front wheel role* of a bike must be played by a wheel, and *front wheel* which is a wheel playing a *front wheel role* is called ***role holder*** and

type similar to height and depth because it is tightly associated with its context and the latter *length* means one-dimensional extent and is a type similar to area and voltage. This is why “(something’s) length” is dealt with a ***quality role type*** rather than ***generic quality type***. Exactly the same applies to “(someone’s) weight”, too, that is, “(someone’s) weight” is a quality role played by an instance of weight as a generic quality type.

Height role holder of human type also needs an explanation. *Height role* of human is a type of *height role* of John and the former height is a *quality role* type and the latter *height* is a *quality*. These relationships are consistent with type-instance relation. On the other hand, however, it is not clear what is *height role holder* of human type. It is a type of *J-height role holder* of John in Fig. 6. At the same time, it looks like *quality role type* as well. This asymmetry comes from the nature of roles. We have no space to discuss this topic in detail, and I would like to discard what *height role holder* of human is and to leave the definitions as they are because *height role* of human has less association with "definite value" than *height role holder* of human.

3.2.6 On terminology

There exist similar terms such as *property, attribute, quality, quality type, quality role type and quantity*. All are dependent entities and cannot exist without an entity with which they are associated. It was almost impossible to define these terms without discussing deep issues related to them. Because of this difficulty, terms such as ***attribute*** and ***property*** used thus far should not be taken as technical terms but as common words which are vaguely defined. But, now it is the time to differentiate them based on our discussion thus far. *Quality, quality role type and quantity* are already defined with examples. So, the problem is the rest two. Therefore, these terms appearing below should be taken as defined here.

Property: As a common term, property is almost a synonym of quality in our context. As a technical term, however, they are different. The major difference is that while a *property* can be used as any predicate to any individual like human(x) or animal(x), *quality* is used only for what an entity has. However, considering that our problem in this paper is to talk about how to describe/characterize entities, property and quality become synonym again because there is no necessity for talking about "human as a property" when describing "human" entity. This definition is only valid in the following sections in this paper, that is, the term *property* appearing in the above should not be taken as defined here.

Because we defined ***quality*** differently from what is defined in BFO, by *property*, I mean *"quality"* used in BFO, that is, value combined with its variable or the variable taking a value that can be possessed by an entity. I think I need to defend my decision on this. ***Quality*** must be something associated with an entity and can change with keeping its identity. These characteristics necessarily derive that ***quality*** cannot be something combined with its value, otherwise, it cannot keep its identity when its value has changed. Therefore, the notion of *quality* employed in BFO is not appropriate in this respect. As was already discussed, an instance of BFO-defined quality such as "John's height of 160cm long" is called *quality instance* and any length whose value is 160cm is represented as *property* in YAMATO, though *property* is defined mainly for *qualitative property* such as *being red, being natural, being artificial, etc.*.

Attribute: Originally, it is a binary predicate which relates an entity to its *quality instance*, e.g., length(pen_1, 10cm). In AI, <E, A, V> is often used to describe entities. In such cases, "A" stands for *quality role type* and/or *generic quality type*. Following the use of this term in AI, we could define *attribute* as a synonym of ***generic quality type***. However, we intentionally leave this term undefined for flexible use in text. So, the term *attribute* should be taken as a common term in this paper.

In YAMATO implemented in Hozo, the term *property* is used to mean BFO's *quality*. Because the role hierarchy is implicit in Hozo, hierarchies of ***quality*** and ***quality role type*** are invisible unless Role hierarchy mode of Hozo is used.

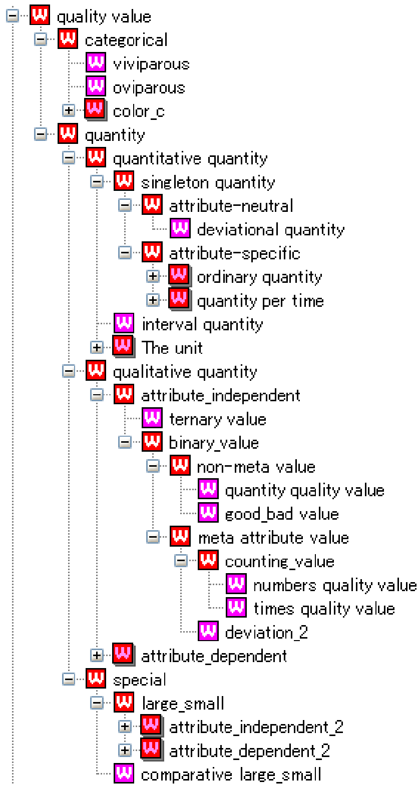


Fig. 8 *Quantity*

3.3 Quantity

3.3.1 What is quantity?

I understand there is a strong temptation to say that quantity is instance of quality. For example, "1m long" seems like an instance of length quality. At first glance, it looks true because it seems simple and natural. However, the facts that quantity must have its own subtype structure to represent quantized quantity for qualitative values which should be independent of quality role type and that quantity has something to say about their instances independently of their corresponding quality, e.g., *"large* is larger than *small"* and "green is the complement of red"(Dolce) show it is not appropriate. Quality and quantity must have their own hierarchies to represent their mutual independence. At the same time, their tight relationship must be modeled appropriately.

Now, what is quantity? Philosophically speaking, quantity is generically-dependent entity in the sense that while it is not specifically-dependent on any physical entity, it cannot exist if all physical objects disappear from the world. Practically, it is what is denoted by its measurement. We never know the true quantity of anything because measurement is always incomplete. What is correct is that there are two kinds of quantity: true quantity and its measurement in which the latter is an approximation of the former. True quantity is unit-independent. A length quantity is independent of how long 1m is. We will come back to this issue later.

3.3.2 is-a hierarchy of quantity

Quantity has its own hierarchical structure as shown in Fig. 8. Major features of this *quality value* hierarchy include separation of *categorical* and *quantity*, and introduction of qualitative values. By *categorical*, I mean genuine categorical quality value such as viviparous/oviparous, colors such as red, blue, green, etc. Note here that categorical value and qualitative (quantized) value(*qualitative quantity*) are intrinsically different. While categories have typicality, qualitative values do not. For example, red as a category has the typical redness at a certain frequency with peripheral redness at higher and lower frequencies. On the other hand, quantized qualitative values do not have such typical thing but still have, like quantity, ordinal relationship between values which categories do not. *Quantity* is further divided into *quantitative quantity* and *qualitative quantity*. The former is divided into *attribute-neutral* and *attribute-specific,* and the latter includes *"large", "high", "big", etc*. Examples of *attribute-dependent qualitative quantity* include *long, heavy, expensive, etc. Quantitative quantity* includes ordinary values such as *length quantity, weight quantity*, etc. ***Large_small*** is special in the ontology of quantity. It is introduced to make powerful qualitative values. Examples include high/low, big/small, etc. One of the well-known issues related to these values is how to realize qualitative values that are compliant with "A small elephant is bigger than a big ant" while guaranteeing "big" is larger than "small" in each context. ***Large_small*** realizes such qualitative values in a sophisticated way. Details are not discussed here. Those who are interested in it are encouraged to read the ontology. This is one of the most remarkable advantages of YAMATO over existing upper ontologies which cover only higher level types which are nowadays becoming a bit obvious.

3.3.3 Confusion about quality as ontology and quality description

Quantity is intrinsically close to *description*, since it must be measured to be known by humans and then it must be described to exist as data about the measured entity. Without measuring, we cannot talk about particular quantity. This is an aspect of quantity that is tightly connected to *description*. However, we have to understand that quantity is ontologically measurement-independent, and hence it is unit-independent, that is, length quantity, for example, can exist as the same quantity independently of how long 1m unit is. What is influenced by the size of the unit is the measurement result which is a description (representation). Quantity is tightly connected to representation in another sense, e.g., a length quantity has multiple ways of representation depending on which unit is used among km, m, cm and mm or inch, etc. In YAMATO, these issues are clearly covered which are not covered by BFO or DOLCE. YAMATO has ontology of data representation as well as ontology of quality discussed thus far, which is discussed below in 4.3.

There are two kinds of unit: one whose size is standardized like 1 m and the other one that represents a kind of quantity such as length. In YAMATO, the former is called "The unit" and the latter "unit".

3.4 Quality and its description

There must be a clear distinction between reality and its description. That is, <E, A, V>, <E, P> and <E, P, V> are not reality but quality description of reality. Note here, however, that these descriptions are also reality in the sense that they exist as data or description just like books exist. This two-level reality is one of the sources of confusion about quality-related issues. Both must be included in ontology with clear distinction between the two. As far as the author knows, both BFO and DOLCE deal only with the primary reality, that is, they do not deal with quality description. YAMATO would be the first ontology which takes care of both.

In order to deal with description properly, we first need to establish ontology of description/representation. As usual, BFO deals with representation (informational object) only at the highest level. It is called *generically-dependent entity* and it is not further elaborated what they are. The author has once proposed the ontology of representation(Mizoguchi 2004) which has not been well-disseminated yet. In this paper, we re-visit the topic in more detail with some modification of the old one in Section 4.

3.5 Evaluation and discussion

Examples of several quality descriptions are shown in Fig. 9 in which "(XYZ)" shows the example is the way recommended by XYZ ontology. These are presented for demonstrating variety of existing data descriptions and how well YAMATO can cope with them. Note here that <E, A, V>-type description is *de facto* standard in all the engineering data and that <E, P, V>-type descriptions are often found in clinical domains. These facts strongly suggest that only BFO or DOLCE cannot cover existing quality descriptions by itself. As we have discussed above, all the attributes of an entity are *quality roles* including degenerated cases, and hence "A" in <E, A, V> represents

1. **He is *tall*. (BFO)** $< E, P_1 >$
2. **He is *185cm* high. (BFO)** $< E, P_2 >$
3. **His *height* is *high/big/large*. (DOLCE)** $< E, A_1, V_2 >$
4. **His *height* is *185cm*. (DOLCE):** $< E, A_1, V_1 >$
5. **This rose is *red*. (BFO)** $< E, P_3 >$
6. **The *color* of this rose is *red*. (DOLCE)** $< E, A_1, V_3 >$
7. **The *color* of this rose is *xyz Hz*. (DOLCE)** $< E, A_1, V_{1\_1} >$
8. **The *redness* of the *color* of this rose is *high*. (Galen)** $< E, P_3, V_2 >$
9. **This rose doesn't have *redness*. (Galen)** $< E, P_3, V_4 >$
10. **Diarrhea of this patient is severe (Galen)** $< E, P_4, V_2 >$
11. **The *length* of the pen is *short*. (DOLCE)** $< E, A_1, V_5 >$
12. **The *conductivity* of this material is *high*.** $< E, A_2, V_2 >$
13. **The *insulativity* of this material is *high*.** $< E, A_3, V_2 >$
14. **This road has *many* curves.** $< E, A_4, V_6 >$ **(This road is curvy)**
15. **Tom visited Kyoto three times.** $< E, A_5, V_7 >$ **(He is a frequent traveler)**

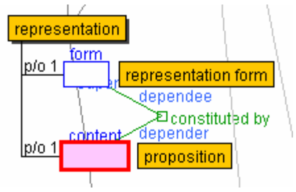
Fig. 9 Examples of quality description.(See appendix for Ai, Pj, Vk) only *quality role type* and do not represent *generic quality type* such as length(1D-extent), weight, mass, etc. Precisely speaking, therefore, the description "(DOLCE) <E, Ai, Vj>" in Fig. 9 should be understood as "DOLCE supports <E, A, V>-type of description" excluding the meaning of "DOLCE supports Ai". As shown in Fig. 9, various quality descriptions are supported by YAMATO. Among them, descriptions 14 and 15 need special attention. "many" (curves) and "many times" are qualities not of "curves" nor of "events", but of the road and Tom, respectively.

We formulated the mutual transformation between the above three kinds of representation. We have conducted two evaluation experiments of our quality ontology as well as the transformation. One has been done using Nanotechnology Index Ontology (NIO) developed by Yamaguchi, et al.(Yamaguchi 2010) and the other using clinical observation descriptions. NIO consists of 2,300 concepts in its *is-a* hierarchy. These concepts are categorized into 5 categories (Process, Structure, Function, Material, and Application). In clinical observation, we used all the 3465 observations contained in the master file of MEDIS(Medis 2010) which is currently used in the description of clinical observations in Japanese medical practice. In both cases, we confirmed our ontology and the transformation among three kinds of quality descriptions works quite satisfactorily. In addition, Masuya has been demonstrating YAMATO's role of promoting interoperability for integration of mouse phenotype databases(Masuya 2009) in which ontology of quality and quantity has been exploited as well as the transformation formalism among three kinds of quality description.

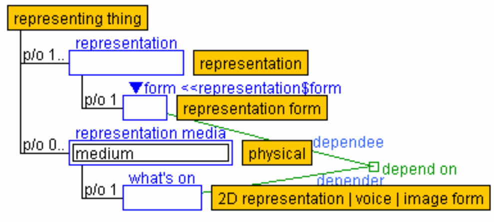
YAMATO's ontology of quality and quantity is more complex than that of BFO and DOLCE. People might consider this is a drawback of YAMATO. But, it is not the case. Although they are simple and seem to be right, BFO and DOLCE deal only with high-level types so that it is not enough to help people capture the reality and cannot support variety of existing data descriptions by itself. YAMATO's ontology of quality and quantity has been achieved thanks to the power of our role theory. The differentiation between height and length (one-dimensional extent) is beautifully realized by introducing ***quality role type*** which shows that it is tightly associated with an entity which necessarily differentiates height from length(1-D extent) which is not quality.

4 Ontology of informational objects

Ontology of representation (informational object) is badly needed. Imagine WWW information resources. All of them are representations. It would not be a useful ontology if it does not contain presentation in it. By representation, we here mean "content-bearing thing". That is, anything which has content as its essentials rather than itself which is playing the role of carrier/bearer of the content. A typical example is a sentence. What exist in WWW, that is, what we can be reached at URLs are not real entities but representations. Similar 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 individuals? These are the questions to answer in this section. To answer them, we need a sophisticated



(a) Representation



(b) Representing thing

Fig. 10 Slot structures of representation and representing

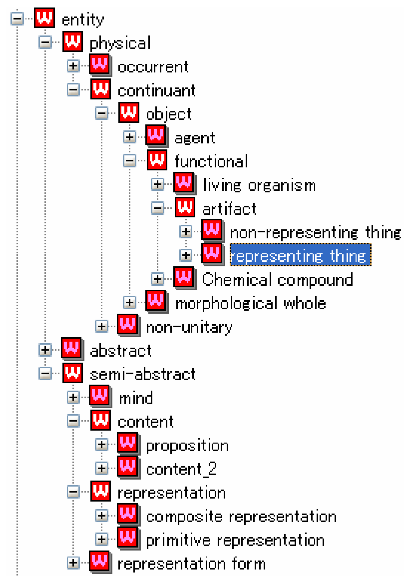


Fig. 11 Top-level structure of representation

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: a book of the novel, the sentences of the novel and their meaning (story). Concerning the procedure, a document of the procedure description, the procedure description, its meaning (sequence of steps) and its execution of those steps are the candidates. Similarly, a piece of music also has four candidates: a 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 these questions is that a representation has several deeply-related concepts such as its embodiment, the mode or the form of representation and its content.

4.1 A conceptual model of representation

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

Definition of a class is not a *representation*

An ontology is a theory of being which explains the real world. Therefore, it exists 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. It is true that a 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 would accept the above observation, as will be clarified below, we would 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.

A representation is composed of two parts, *form* and *content* shown in Fig. 10(a) like SUMO(Pie 2010). A representation is not concrete yet as it is. It becomes a physical individual only when it becomes a representing thing shown in Fig. 10(b). Top-level structure of representation is shown in Fig. 11.

This separation is critical. Without this we could not expect a convincing ontology of representation. We need to realize something exists as content independently of its representation. *Content* is the hidden part of representation and it is a proposition which the author of the representation would like to convey through the representation. Apparently, there can be many ways to code the same content into different forms to produce different representations. The identity of "Representation" is made of both *form* and *content*. If their *contents* are different, then the representations are different. Concerning things of the same content, the identity is determined only by *form* which is usually what people sense its existence. When the language is written down on a sheet of paper, it becomes a representing thing.

Figs. 12 and 13 show *is-a* hierarchies of *representation form* and *proposition*, respectively. The former is divided into *ordinary form* and *unit-related form*. As is seen by the subclasses of *ordinary form*, typical *representation form* includes language as a symbol sequence.

Unit-related form is special because it is introduced to cope with quantity as representation discussed in Section 3 and is discussed later intensively. *Ordinary form* is divided into *symbol sequence* and *image form*, and the latter into *still image* and *motion image*.

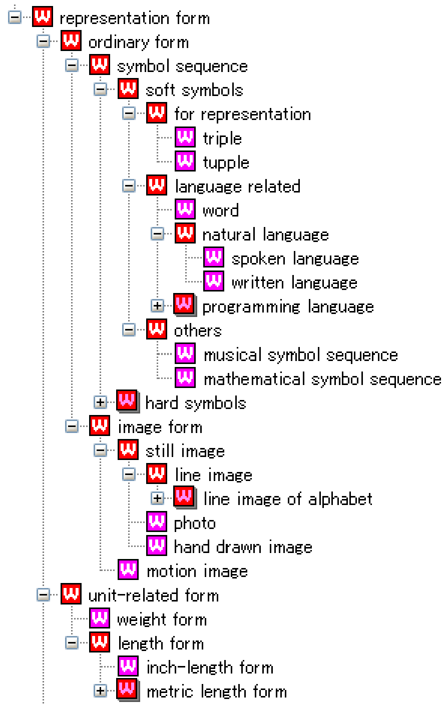


Fig. 12 *Is-a* hierarchy of *representation form*.

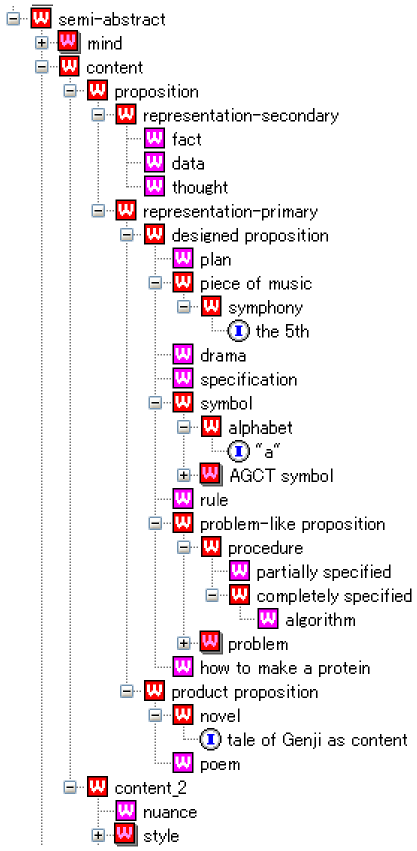


Fig. 13 *Is-a* hierarchy of *proposition*

Proposition is what is to be represented and is divided into *representation-primary* and *representation-secondary*. The latter includes fact, data and thought, etc. By *fact*, I mean a happening that a human has recognized and is ready for representation to talk about it. The very fact happening in the real world is an instance of event. *Representation-primary* proposition has two subclasses: *designed proposition* and *product proposition*. The former works as **specification** of the production of something. The latter itself is the product. For example, the content of a *piece of music* is a specification of the music sound produced by a music player. *Content of a 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. Note that as will be discussed later, appreciation of a novel is different from specification which is common to anything.

4.2 Evaluation and discussion

It is important to clearly distinguish between a representation and a representing thing. Any representation is not embodied unless it becomes a representing thing. A sentence “This is a book” is a representation in the form of natural language(English) whose content is the meaning of “This is a book” and what you see is its printed realization on a sheet of paper which is a representing thing. Before writing it down, the symbol sequence “This is a pen” is not a physical individual because it does not specify what particular icon of symbols are used to represent each of the symbols. Symbols without icon are semi-abstract individuals as discussed below. Note here that “This is a pen” before writing it down, that is, before becoming a concrete thing is not something in one’s mind but simply a symbol sequence independently of where it exists.

4.2.1 On copying

There seem to exist two kinds of “copy” of *representation by symbol*: (1) copying the symbol sequence as a symbol level thing and (2) copying the symbol sequence as an image(photo-copying). YAMATO’s representation ontology covers both (1) and (2). Copying of type (1) cannot be done for *representation* because any sequence of symbols is unique and is independent of what concrete icons of letters are used. That is, there is only one “This is a book” sentence in the world at the representation form level[8]. In other words, contrary to intuition, what you can type-(1)-copy is only *representing thing*. In such cases, you can copy it in any hand-written letters or type it in different fonts and sizes. It is obvious that type-(2)-copying can be done only for *representing things*. In the following, therefore, I use the term “copy” discarding the distinction between types (1) and (2). The copied stuff is what is on the representation medium in both cases. Copying is just a generation of the same representation on a different medium. That is, what you can copy is not *representation* but *representing thing*. In fact, a novel, say, Tale of Genji, exists in the form of a book. A copy of a book is physically a 3D thing with a logical structure of chapter, section, etc. and with the content composed of a representation in terms of sentences/images. Tale of Genji exists independently of on what medium it is written.

[8] Do not confuse “This is a pen” as a representation with “This is a pen” as representation form. While the former includes its content and exist in the world as many as its content denoted by it (representation), the latter is just a symbol sequence.

4.2.2 Identity

Identity of a *representation/representing-thing* is divided into four kinds: (0) content level, (1) symbol level, (2) icon level and (3) medium level. Obviously, at the content level, two *representation/representing-things* are identical **iff** the contents are identical. At the symbol level, the identity comes from the representation form, that is, they are identical **iff** the representation forms are identical in terms of symbol sequence. At the icon level, they are identical **iff** the representation forms are identical in terms of icon level. The medium level applies only to representing things, and two representing things are identical **iff** they are icon-level identical and their media are the same as physical individuals. That is, it follows the common principle of identity of physical entity. Therefore, a hand-written copy of a sentence on a sheet of paper is identical with the original one, at the level (1), **iff** there is no transcription error. A photocopy of a sentence on a sheet of paper is identical, at the level (2), with the original one **iff** the photocopying quality is higher than a threshold and they are always identical with each other in terms of symbol level. Concerning medium level identity, there is only unique representing thing, since copying physical entities is impossible in principle.

4.2.3 Some examples

Now we are ready to discuss what are procedure, music, etc. The following lists samples of definitions of some types and individuals:

```
Algorithm representation
  is-a:  Representation
  p/o"form": Language
  p/o"content": Algorithm
A representation of Quicksort algorithm with C
  instance-of: Algorithm representation
  p/o"form": <C language>
  p/o"content": <Quicksort algorithm>
Musical score
  is-a: Representation
  p/o"form": Musical symbol sequence
  p/o"content": a piece of music
A score of symphony, the 5th
   instance-of: Musical score
  p/o"form": <A sequence of musical symbols>
  p/o"content": <Symphony, the 5th>
A copy of a book of musical score of the 5th
   instance-of: Representing thing
   p/o"representation": <A score of symphony, the 5th>
   p/o"medium": <Pieces of paper>
Sentence
  is-a: representation
  p/o"form": natural language
         p/o: sequence of alphabets
  p/o"content": proposition(the meaning)
<This is a book> as a sentence
   instance-of: Sentence
   p/o “form”: <This is a book> as a symbol sequence
   p/o “content”: <the meaning>
<This is a book> as a symbol sequence
   instance-of: symbol sequence which is-a
          representation form
   p/o: ”T”,”h”,”i”,”s”,” “,”i”,”s”,” “,”a”,…..
```

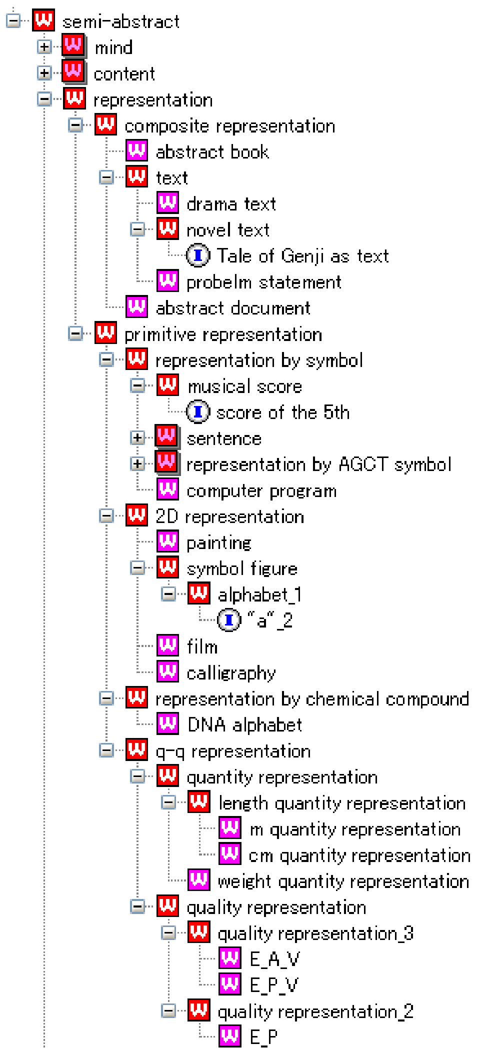


Fig. 14 Hierarchy of *representation*.

<This is a book> as a speech
  instance-of: Representing thing
  p/o "representation": <This is a book> as a sentence
  p/o "medium": <a sound>
<This is a book> as a written sentence
  instance-of: representing thing
  p/o "representation": <This is a book> as a sentence
  p/o "medium": <a piece of paper>

The above models declare that an instance of an algorithm is composed of an instance of language description called *form* and an instance of proposition which is a subclass of *designed 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 an algorithm is not a sequence of actions performed according to the algorithm description but a specification of valid action sequences. That is, an instance of an algorithm is a specification of its execution. A specification can virtually exist not as being an embodiment(representing thing), but as the content of representation. In general, all the *designed propositions* are specification of their succeeding realization actions.

The next problem might be what is the sound (symphony the 5^{th}) 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 which should be constructed in the hearer's mind by interpreting the sequence of musical sound and coincide hopefully with the specification of the piece of music. The playing action is an instance of playing action and at the same time a realization action of the musical sound. Furthermore, the relation between the musical sound and symphony the 5^{th} is *realization-of* relation. Fig. 14 shows an is-a hierarchy of representation.

4.3 Quality representation

As was discussed in Section 3, quality representation must be dealt with explicitly with careful attention on distinction between reality and description of reality. YAMATO introduces *tupple, triple* and *unit-related form* as representation form(Fig. 12), *E_A_V, E_P_V,* and *E_P* as representation of quality, and *quantity representation* (Fig. 14) for dealing with *m, cm, mm, etc*. Fig. 15 shows definitions of *measure, cm length form* and *cm quantity representation. Measure* somehow transforms the real quantity to its approximation as a representation, that is, we understand the measurement produces a representation. In the ontology in Fig. 15, the measurement result is referred to as *quality measurement* which is a role holder and must be an approximation of the real quantity which should be unit-independent. *Quality measurement(RH) is* referred to in the class constraint of the content slot of *cm quantity representation* to represent the measured quantity in the form of *cm*. The length representation is referred to as *length in cm* role holder in the form slot of *cm quantity representation*. *Cm length form* is defined using *unit of length* and value slot which stores a *number* as a value referring to *1m unit length* as the standard length. It also defines *cm* as a role holder which is a new finding on how to define *cm, mm, km, etc*. In summary, quantity is made accessible as a *proposition* as a result of *measuring*. Then, it is representing in, say, cm quantity representation in which length in cm of the quantity is obtained. Thus, a quantity is made an accessible thing as a *representation form* in a *representation* rather than a *proposition* keeping the measured quantity in its corresponding proposition.

On the basis of the preparation, a unique type named ***equivalence*** is introduced to explicitly show the correspondence between the three ways of quality descriptions. Although this topic is interesting to discuss, details are omitted in this paper.

5 Objects, Processes and Events

Although this topic is the most philosophical and hence the most important, we here present only the summary of the discussion as follows because it is in-depth discussed in (Galton 2009):

(Excerpt from (Galton 2009)) Any change must be a change of something. This is already an argument against a 'pure process' view of reality, since we cannot conceive of processes without their material support. One might ask: what is a person over and above the sum of its internal processes? But what makes this sum worthy of consideration at all is that they constitute some kind of unity; the unity comes from the fact that there are other processes, its external processes, which it enacts.

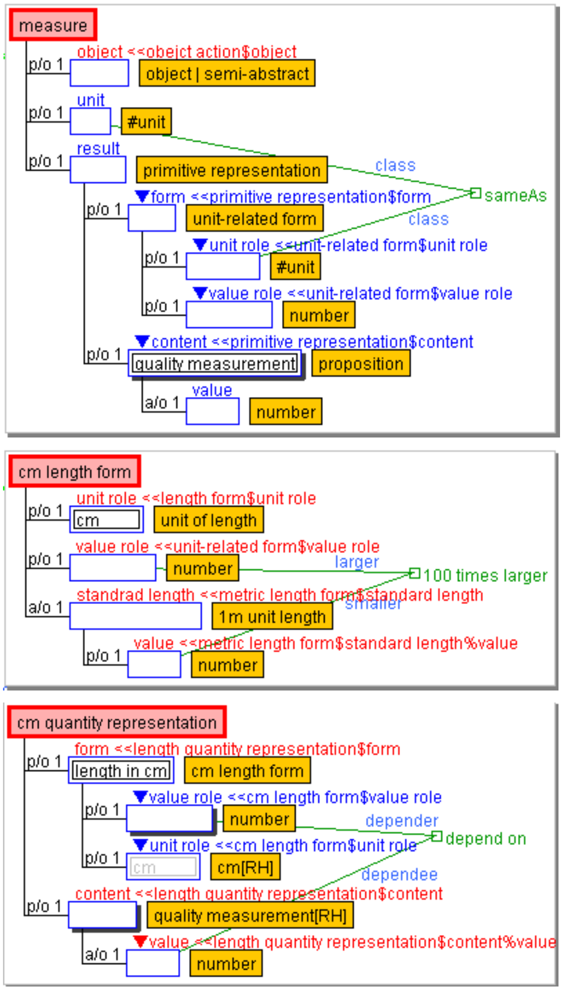


Fig. 15 Measurement and length form.

Thus these questions make the mistake of focusing only on the internal processes of a person, whereas the external processes play an essential role in determining the identity of the object. Hence, rather than trying to characterize an object in terms of its internal processes (e.g., by identifying the object as the sum of those processes), we would rather say that an object is a unity which is what enacts its external processes. We could indeed say that the object is the interface between its internal and external processes: it is a point of stability in the world in virtue of which certain processes are characterized as internal and others as external. The issue of external vs. internal processes summarizes as "The water falls, but the waterfall doesn't fall". That is, what a waterfall is doing is not the water falling but migration upstream as it carves its way into the rock.

Similarly, what a river is doing is not the water flow but changing the shape of its course of the flow. This is why we can consider a river is an object which has water flow as its inner process.

Another important topic is what can change. It is also summarized as follows(see (Galton 2009) in detail): An event cannot change. This is not because it is already a change, but because it exists with unity in the temporal space, that is, it must be always viewed as a whole. Events are made of processes. Processes are dissective and hence they have no unity. It has no whole, and hence it has no (temporal) part. Processes are essentially on-going stuff and hence they can change. YAMATO is based on these findings and has sophisticated upper categories of events and processes.

Parts of an occurrent in YAMATO need discussion. There are two kinds of parts of occurents: temporal parts and causal parts. While the former correspond to mereological parts of continuant, the latter to functional parts. Considering that occurrents are things in the temporal space, the idea of mereological parts is straightforward, that is, interval-based segmentation for identifying parts makes sense. However, any process and event has its causal parts. Imagine, John sneezes while he is walking. The sneezing is nothing to do with his walking and hence it is not a causal part of his walking. On the other hand, the motion of his legs is essential to his walking. The alternate motions of his left and right legs are causal parts of his walking. The bending motions of his both legs are causal parts of the leg motion. Any causal part is temporal parts, but, not vice versa.

YAMATO is said to be based on 3D view because it clearly distinguishes between continuant and occurrent, at the same time, however, it is said to be based on 4D because it accepts what exist in the world are occurrents rather than continuants. It is based on a solid understanding on what can change and what is an object at all. I could say YAMATO is 3.5D by which, I mean it is based on the idea of neither "objects prior to processes" nor "processes prior to objects, but "both are mutually dependent".

Events are composed of sub events and all those events are constituted by processes as material. Actions whose operand is anything other than actions are divided into *explicit action* and *state action*. The former represents actions which imply how to do it, while the latter actions which mainly imply what state change(what to do) occurs keeping how to do it implicit. The implications of this classification of actions are rich and new. While *walk* belongs to the former, *move* to the latter. YAMATO distinguishes between *arrive* as an action and *arrival* as an event. As described above, an *action* is essentially on-going stuff, while an *event* is a unitary whole. An action exists in an open interval without either end of the time interval, while an event exists in a closed interval. So, an action *arrive* does not include the very end point.

The separation of what to do and how to do is the very philosophy of our functional ontology (Kitamura 2006, 2010) which provides us with an innovative view of actions.

6 Roles, functions and relations

Roles are not adequately visible in YAMATO, since it is already reflected/embedded in Hozo tool. While ordinary types such as *human*, *table*, *etc.* are defined independently of each other, roles are defined within a context which they necessarily depends on in our role theory (Mizoguchi 2007). In fact, roles are used in defining all the types in YAMATO. These are the reasons why YAMATO does not have to incorporate roles in it. When one uses Hozo with YAMATO, he/she can enjoy maximally the benefits offered by both.

Kitamura and I have been intensively involved in building functional ontology for years(Kitamura 2006)(Mizoguchi 2009b). Although we have already developed a convincing functional ontology which is compliant with YAMATO, it is not incorporated into YAMATO because it is too professional for general readers.

Relations are not incorporated in YAMATO neither because Hozo deals with relations and the wholeness concepts (ordinary types) in different worlds. Similarly to Roles, YAMATO implemented in Hozo exploits built-in and user-defined relations. Hozo has three built-in relations: *is-a*, *part-of* and *attribute-of* which are exploited to define types in YAMATO. These are used to introduce fundamental definitions of types, and user-defined relations are used to impose semantic constraints among parts and attributes of a type.

7 Concluding remarks

YAMATO has been implemented in Hozo and is available at http://www.ei.sanken.osaka-u.ac.jp/hozo/onto_library/upperOnto.htm with a tool-independent browser. Its OWL version will be available. It is currently being axiomatized in the EuJoint project(Borgo 2010). YAMATO was first built in 1999. Since then, it has been refined and revised several times. The current version has been extensively used in several projects such as development of medical ontology(Kou 2008)(Mizoguchi 2009a), ontology of learning and instructional theories which is the first ontology in the community (OMNIBUS), mapping ontology between PATO and YAMATO (Masuya 2009), ontology of genomics (GXO (Masuya 2009)), modeling of mobile users' behavior(Sasajima 2008), functional ontology(Kitamura 2006), etc.

As is shown in 3.5 on evaluation, ontology of quality representation has been well evaluated and its utility has been demonstrated. Our role theory has been used more than ten years with Hozo tool, which suggests it has been already well-established. Ontology of representation is being used in Genomic ontology, GXO (GXO). Although the object/process/event ontology is rather new, it is carefully designed to be compatible with our functional ontology. The key point is the structure of *action* whose taxonomic structure is designed based on device ontology so that it is clearly differentiated between what is performed and how it is performed. In summary, YAMATO, an upper ontology which has appeared at last after DOLCE, BFO, GFO, SUMO and CYC, would be a powerful ontology in the three respects discussed in this paper. Its full power can be enjoyed when it is used with Hozo.

Acknowledgement

The author is grateful to my colleagues Yoshinobu Kitamura, Kouji Kozaki, Munehiko Sasajima and Yusuke Hayashi for their valuable comments. Special thanks go to Stefano Borgo for his comments during his work on axiomatization of YAMATO.

8 References

Alan, R. et al. (2010) http://www.openclinical.org/prj_galen.html

Borgo, S. et al. (2009) http://www.ei.sanken.osaka-u.ac.jp/main/project-en.html

Egana, Mike, et al.(2008): Applying ontology design patterns in bio-ontologies, *Proc. of 16th International Conference, EKAW 2008*, Springer, LNAI 5268, pp.7-16.

Galton, A. and R. Mizoguchi (2009): The water falls but the waterfall does not fall – New perspectives on objects, processes and events, *Journal of Applied Ontology*, Vol. 4, No.2, pp. 71-108.

Gangemi A, Mika P, (2003): "Understanding the Semantic Web through Descriptions and Situations", Meersman R, et al. (eds.), *Proc. of ODBASE03 Conference*, Springer.

Guarino, N. et al. (2010) http://www.loa-cnr.it/DOLCE.html

Herre, H. Et al. (2010) http://www.onto-med.de/ontologies/gfo/

Guarino, N. (1995): Top-level ontological categories, *International Journal of Human and Computer Studies*, 43, pp.669-685.

Kozaki, K. et al. (2010) http://www.hozo.jp/

Isotani, Seiji and Riichiro Mizoguchi (2008): Theory-Driven Group Formation through Ontologies, *Proc. of the 9th International Conference on Intelligent Tutoring Systems (ITS'08)* pp. 646-655.

Yoshinobu Kitamura, Yusuke Koji and Riichiro Mizoguchi (2006): An Ontological Model of Device Function: Industrial Deployment and Lessons Learned**,** *Journal of Applied Ontology (Special issue on "Formal Ontology Meets Industry"),* Vol. 1, No. 3-4, pp. 237-262.

Yoshinobu,K and Mizoguchi, R. (2010): Characterizing Functions based on Ontological Models from an Engineering Point of View, *Proc. of the Sixth International Conference on Formal Ontology in Information Systems (FOIS 2010),* Toronto, Canada, May 11-14, pp. 301-314, IOS Press.

Kou, H. et al. (2008): A Fundamental Consideration toward Development of Medical Ontology, *Proc. of the 22nd Annual Conference of the Japanese Society for Artificial Intelligence*, 2E3-01 (in Japanese).

Lenat, D. et al. (2010) http://www.opencyc.org/

Masuya, M. and Mizoguchi, R. (2009): Toward fully integration of mouse phenotype information, *The 2nd Interdisciplinary Ontology Conference*, pp. 35-44, Tokyo, Japan.

Masuya, H. (2010) http://www.brc.riken.go.jp/lab/bpmp/Ontologies/GXO/GXO.html

http://www.ei.sanken.osaka-u.ac.jp/hozo/onto_library/GXO.html

MEDIS (2010) Medical Information System Development Center: http://www2.medis.or.jp/master/syoken/

Mizoguchi, R. (2004): Tutorial on ontological engineering - Part 3: Advanced course of ontological engineering, *New Generation Computing*, OhmSha & Springer, Vol.22, No.2, pp.198-220.

Mizoguchi R., Sunagawa E., Kozaki K. and Kitamura Y. (2007): A Model of Roles within an Ontology

Development Tool: Hozo**,** *J. of Applied Ontology,* Vol.2, No.2, pp.159-179. Sept.

Mizoguchi, R. et al. (2009a): An Advanced Clinical Ontology; *Proc. of International Conference on Biomedical Ontology (ICBO)*, pp.119-122, Buffalo, NY, June 24 - 26.

Mizoguchi, R. and Y. Kitamura (2009b) A Functional Ontology of Artifacts, *The Monist*, Vol. 92, No.3, pp.387-402, July.

Mizoguchi, R. Bourdeau, J, and Hayashi, Y. (2010) http://edont.qee.jp/omnibus/doku.php

Yamaguchi, Y. et al. (2010) http://mandala.t.u-tokyo.ac.jp/english/index. html and Development of Contents Management System Based on Light-Weight Ontology, K. Kozaki, Y. Kitamura and R. Mizoguchi: *Proc. of the 2007 IAENG International Conference on Internet Computing and Web Services*, Hong Kong, 21-23 March, pp.987-992, 2007. http://www.ei.sanken.osaka-u.ac.jp/pub/kozaki/IMECS2007_koza_cr.pdf

Pie, A, et al. (2010) http://www.ontologyportal.org/

Sasajima, M., et al. (2008): Obstacles reveal the needs of mobile internet services – OOPS: Ontology-based obstacle prevention and solution modeling framework., *Journal of Web Engineering*, Vol. 7, No. 2., pp. 133-157, Rinton Press.

Sasajima, M., Y. Kitamura, M. Ikeda and R. Mizoguchi: (1995) FBRL: A Function and Behavior Representation Language**,** *Proc. of IJCAI'95*, pp.1830-1836, 1995.

Smith, B. et al. (2010) http://www.ifomis.org/bfo

Appendix : Details of quality and quantity hierarchy shown Fig. 9.

- ***property***
  - ***qualitative property***
    - ***intrinsically qualitative(P3)***
      - ***attribute-inclusive(P1)***
    - ***extrinsically qualitative***
      - ***attribute-inclusive***
        - » ***being diarrhea(P4)***
  - ***quantitative property(P2)***
- ***generic quality (attribute)***
  - ***intrinsic generic quality***
    - ***basic generic quality***
      - ***non-meta generic quality(A1)***
      - ***meta-generic quality***
        - » ***counting generic quality***
          - ***+ number of objects(A4)***
          - ***+ number of events(A5)***
    - ***I_O relational***
      - ***continuous***
        - » ***forward/(A2)***
        - » ***backward(A3)***
  - ***Accidental generic quality***
- ***quality value***
  - ***categorical(V3)***
  - ***quantity***
    - ***quantitative quantity***
      - ***singleton***
        - » ***attribute-specific***
          - ***+ ordinary quantity***
            - ***- length quantity(V1)***
            - ***- frequency quantity(V1_1)***
        - » ***attribute-neutral***
    - ***qualitative quantity***
      - ***attribute-inependent***
        - » ***binary_value***
          - ***+ non-meta value***
            - ***- quantity quality value(V2)***
          - ***+ meta attribute value***
            - ***- counting_value(V6)***
            - ***- times quality value(V7)***
      - ***attribute-dependent***
        - » ***length quality value(V5)***