Dynamic *Is-a* Hierarchy Generation System based on User's Viewpoint

Kouji Kozaki[1], Keisuke Hihara[1], Riiciro Mizoguchi[1]

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

Abstract. In ontological theories, *is-a* hierarchy must represent the essential property of things and hence should be single-inheritance, since the essential property of things cannot exist in multiple. However, we cannot avoid multi-perspective issues when we build an ontology because the user often want to understand things from their own viewpoints. Especially, in the Semantic Web, the variety of users causes the variety of viewpoints to capture target domains. In order to tackle this multi-perspective issue, we adopt an approach of dynamically generating *is-a* hierarchies according to the viewpoints of users from an ontology using single-inheritance. This article discusses a framework for dynamic *is-a* hierarchy generation with ontological consideration on *is-a* hierarchies generated by it. Then, the author shows its implementation as a new function of Hozo and its applications to a medical ontology for dynamically generation of *is-a* hierarchies of disease. Through the function, users can understand an ontology from a variety of viewpoints. As a result, it could contribute to comprehensive understanding of the ontology and its target world.

Keywords: ontology, dynamic *is-a* hierarchy generation, viewpoint, disease ontology

1 Introduction

Ontologies are designed to provide systematized knowledge and machine readable vocabularies of domains for Semantic Web applications. The competences of semantic technologies strongly depend on the ontology which they use. Ontology is defined as "An explicit specification of conceptualization" [1], and it clearly represents how the target world is captured by people and systems.

Semantics of concepts (classes) are defined clearly through the description of their relationships between other concepts in an ontology. In particular, the most important relationship is an *is-a* (sub-class-of) relationship which represents a relation between a generalized concept and a specialized concept. Class hierarchies according to *is-a* relationships are called *is-a* hierarchies, and they form the foundation of ontologies. That is, *is-a* hierarchies in an ontology reflect how the ontology captures the essential conceptual structure of the target world.

Therefore, in ontological theories, an *is-a* hierarchy should be single-inheritance because the essential property of things cannot exist in multiple. Imagine that objects, processes, attributes, all of them have their own unique and essential properties. The use of multiple-inheritance for organizing things necessarily blurs what the essential property of things is. This observation is strongly supported by the fact that both of the well-known upper ontologies: DOLCE and BFO use single-inheritance hierarchies.

Nicola Gunarino criticizes the careless usage of *is-a* relationships without enough ontological consideration as *is-a* overloading [2] and propose an ontology development methodology, called OntoClean, which defines concepts based on meta-properties such as rigidity and anti-rigidity. DOLCE is developed based on the OntoClean methodology using single-inheritance *is-a* hierarchy. BFO is the upper ontology used by the OBO Foundry[1] which aims to create a suite of orthogonal interoperable reference ontologies in the biomedical domain. BFO also uses single-inheritance hierarchy, and it is recommended in the guideline of OBO Foundry to avoid careless usage of multiple-inheritance.

However, we cannot avoid multi-perspective issues when we build an ontology across multiple domains. It is because domain experts often want to understand the target world from their own domain-specific viewpoints. In many cases, their interests are different even if they are experts in the same domain. In some domains, there are many ways of categorization of the same kinds of concepts and different taxonomies are used depending on the purpose and situation.

For example, in the medical domain, a disease is interpreted from various viewpoints. Consider diabetes as an example. Clinician may pay attention to the body parts with the abnormalities and classify diabetes as diseases which have abnormality in blood. On the other hand, certain specialists may pay attention to the main pathological condition and may classify diabetes as an abnormality in metabolism, and other specialists may classify diabetes as a lifestyle related disease. Staffs administering medical care implicitly understand which *is-a* hierarchy should be used for disease interpretation in correlation with their respective interpretations. This suggests that one *is-a* hierarchy of diseases cannot cope with such a diversity of viewpoints, since a single-inheritance hierarchy necessarily represents one viewpoint.

Many efforts are under taken to solve these multi-perspective issues. The OBO Foundry proposes a guideline for ontology development stating that we should build only one ontology in each domain [3]. This is an effort to exclude the multi-perspective nature of domains from ontologies. Ontology mapping is used as an approach to acceptance of multiple ontologies based on the different perspectives in a domain. It aims to make clear the relationships between different ontologies. Someone may consider that multiple-inheritance is an easy way to solve these multi-perspective issues. Because multiple-inheritance causes some ontological problems as mentioned above, our ontology development tool, named Hozo[4][2], allows the user to use a multiple-inheritance only when he/she represents clearly from which upper-concepts the essential property is inherited[3]. However, if we define every possible *is-a* hierarchy using multiple-inheritances, they would be very verbose and the user's viewpoints would become implicit.

---

1 http://www.obofoundry.org/

2 http://www.hozo.jp

3 It is represented by two kinds of *is-a* relationships: (essential) *is-a* and (non-essential) *IS-A*.

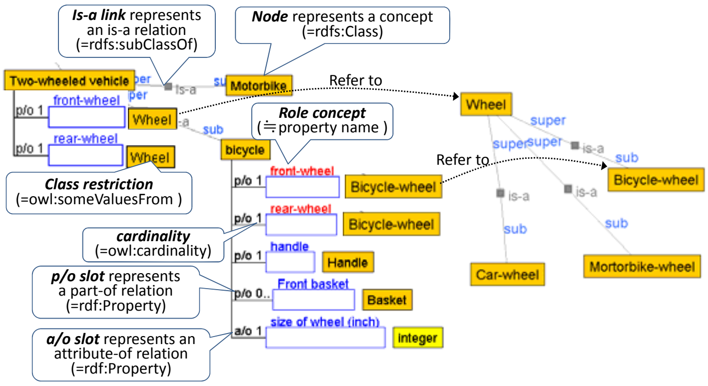


Fig.1 An example of ontology defined using Hozo.

In order to tackle these multi-perspective issues, the authors take an approach based on ontological viewpoint management. It is dynamically generation of *is-a* hierarchies according to the viewpoint of users from an ontology using single-inheritance. The main strategy is composed of: (1) fixing the conceptual structure of an ontology using single-inheritance based on ontological theories and (2) reorganizing some conceptual structures from the ontology on the fly as visualizations to cope with various viewpoints. Based on this strategy, the authors consider a framework for dynamic *is-a* hierarchy generation according to the interests of the user and implement the framework as an extended function of the ontology development tool Hozo. In this article, we discuss the framework for dynamic *is-a* hierarchy generation and its application to a medical ontology. It would solve the conflicting requirements of multi-perspective and single-inheritance in a good ontology, and it could contribute to deep understanding of the ontology.

The rest of this paper is organized as follows: In section 2, we introduce dynamic *is-a* hierarchy generation according to viewpoints. Next, we consider ontological characteristics of the generated *is-a* hierarchies in section 3. In section 4, we discuss implementation of the framework as an extended function of Hozo. In Section 5, we show its application to a medical ontology for dynamic *is-a* hierarchy generation of disease. In section 5, we discuss related work. Finally, we present concluding remarks with future work.

2 Dynamic is-a Hierarchy Generation according to Viewpoints

2.1 Ontology Representation in Hozo

We implement the dynamic *is-a* hierarchy generation system as an additional function of Hozo [4]. Fig.1 shows an example of ontology defined using Hozo. Ontologies are represented by nodes, slots and links. The nodes represent concepts (classes), *is-a* links represent *is-a* (subclass-of) relations, and slots represents *part-of* (denoted by “p/o”) or *attribute-of* (denoted by “a/o”) relations. A slot consists of its kind (“p/o” or “a/o”), *role concept*, *class restriction*, *cardinality*. Roughly speaking, a slot corresponds to *property* in OWL and its role name represents name of the property. Its class restriction and cardinality correspond to owl:someValuesFrom and owl:cardinality respectively. Their restrictions refer to other concepts which are defined elsewhere. However, semantics of Hozo's ontology includes some concepts related to role which are not supported in OWL because it is designed based on an ontological theory of role [5]. While we have designed three levels of role representation model in OWL to capture the semantics level-wise [6], we use the simplest model described above in this paper because it is almost compatible with OWL. That is, the proposed method for dynamic *is-a* generation is also applicable to not only ontologies in Hozo's format but also ontologies in OWL.

In the target ontologies, concepts (classes) are defined by several slots which represent properties and restrictions for them. These definitions are inherited from super-concepts (super-classes) to their sub-concepts (sub-classes) along with *is-a* links. Furthermore, in some sub-concepts, some inherited definitions are specialized according to *is-a* hierarchies of concepts which are referred to by their restrictions. For example, *bicycle* in Fig.1 inherit *front-wheel* from *Two-wheeled vehicle* and its class-restriction could be specialized from *Wheel* to *Bicycle-wheel*. This research focuses on these characteristics of *is-a* hierarchies and considers an approach to reorganize *is-a* hierarchies of concepts based on *is-a* hierarchies of concepts referred to by their definitions.

2.2 Dynamic *Is-a* Hierarchy Generation through Transcription of a Hierarchical Structure

Fig.2 outlines a framework for dynamic *is-a* generation. It generates *is-a* hierarchies by reorganizing the conceptual structures of the target concept selected by a user according to the user's viewpoint. The viewpoint is represented by an *aspect* and a *base hierarchy*. By aspect, we mean something which the user is interested in and selects from the definition of the target concept to generate an *is-a* hierarchy. By base hierarchy, we mean a conceptual structure of concepts which are referred to by the definition selected as the aspect. Because sub-concepts of the target concept could be defined by specializing their inherited definitions according to the base hierarchy, we

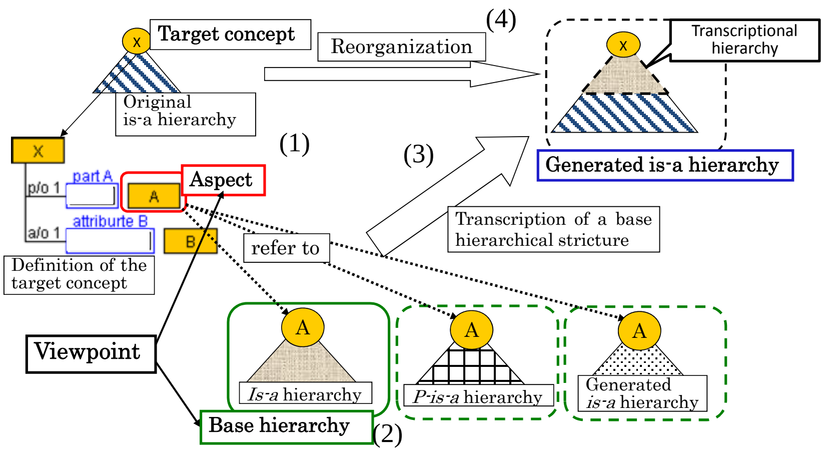


Fig.2 A framework for dynamic *is-a* generation.

could reorganize the *is-a* hierarchy of the target concepts according to the following steps:

Step 1: Selection of an aspect

The user selects something as an aspect from the definition of the target concept for dynamic *is-a* hierarchy generation (see. Fig.2-(1)). Because any concept is defined in terms of slots each of which consists of a role-concept, a role-holder [5] and a class-restriction, he/she can select one of them as an aspect. In this paper, we consider only a case where the user selects a class restriction as an aspect for simplicity.

Step 2: Selection of a base hierarchy

The user selects a base hierarchy from hierarchies of concepts which the aspect is referring to (see. Fig.2-(2)). In Hozo, three kinds of conceptual hierarchies could be the base hierarchy as follows: the *is-a* hierarchy of concepts referred to by the aspect, the *p-is-a* hierarchy which is generated by the system according to part-whole relationships of the concepts referred to and dynamically generated *is-a* hierarchies using the proposed method. A *p-is-a* hierarchy is obtained by abstracting parts from a part-of hierarchy [7]. The detail of the *p-is-a* hierarchy is discussed in section 2.3.2.

Step 3: Transcription of a hierarchical structure

The system defines new sub-concepts of the target concept by specializing the definition of it according to the class restriction selected as an aspect and base hierarchy (see. Fig.2-(3)). Then, their concept names are automatically determined by the system using a template such as "<the target concept name> with <the specialized aspect> as <the role name of the aspect>". As a result, an *is-a* hierarchy which has the same conceptual structure with the base hierarchy is generated. We call the generated hierarchy a *transcriptional hierarchy* and the operations to generate it a *transcription of a hierarchical structure*.

The scope of a transcription of the base hierarchy could be managed by specifying the number of the target layers rather than to use all concepts of the base hierarchy for transcription.

Step 4: Reorganization of *is-a* hierarchy based on a transcriptional hierarchy

The system reorganizes the *is-a* hierarchy by comparing the original *is-a* hierarchy and the transcriptional hierarchy generated in step 3. The system compares the sub-concepts of the target concept (we call them *existing sub-concepts*) with the concepts on the transcriptional hierarchy (we call them *generated sub-concepts*) according to the aspect and the base hierarchy. When an existing sub-concept's definition is subsumed by the definition of a generated sub-concept, the existing sub-concept is classified into a sub-concept of the generated sub-concept. If an existing concept is classified into sub-concepts of multiple generated sub-concepts, the existing concept is classified into the lowest sub-concepts. As a result, all existing concepts are classified into sub-concepts of the generated concepts in the transcriptional hierarchy according to the aspect and the base hierarchy[4].

[4] The result of reorganization corresponds to the result of classification using DL-reasoner while it is implemented by procedural ways in Hozo.

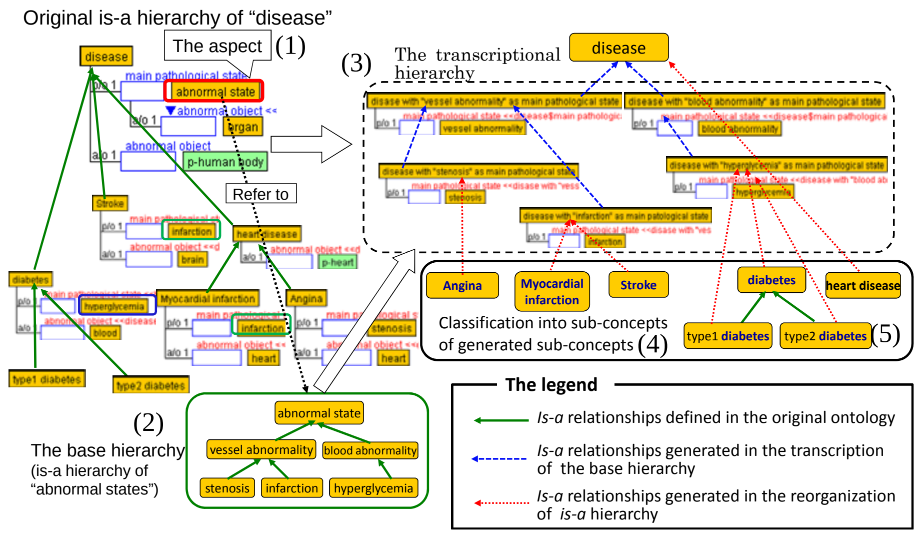


Fig.3 An example of dynamic *is-a* generation of disease in the case that is-a hierarchy of abnormal state is selected as the base hierarchy.

Through the above four steps, the system can dynamically generate *is-a* hierarchies by reorganizing existing sub-concepts according to the transcriptional hierarchies of base hierarchies.

Although DL-reasoners can classify classes (concepts) automatically by reasoning, the result of classification is only an *is-a* hierarchy which is determined uniquely according to the definitions of the classes. Therefore, it is different from our dynamic reorganization according to the users' view points. DL-reasoners can generate a different *is-a* hierarchy only when class definitions in the ontology have changed.

2.3 Examples of Dynamic *Is-a* Hierarchy Generation

§ 1 In the Case of that an *Is-a* Hierarchy is Selected as a Base Hierarchy

As an example, we consider a dynamic *is-a* hierarchy generation of diseases which is defined in terms of several slots such as "main pathological state", "abnormal object" and so on (see. Fig.3). Here, we suppose the user selects the class-restriction of "main pathological state" as an aspect (Fig.3-(1)) and the *is-a* hierarchy of "abnormal state" as a base hierarchy (Fig.3-(2)).

First, sub-concepts of "disease" such as "disease with vessel abnormality as main pathological state" and "disease with blood abnormality as main pathological state" are dynamically generated by specializing the definition of "disease" according to the class restriction selected as the aspect and the base hierarchy. After repetitions of generations of sub-concepts, the transcriptional hierarchy of "disease" is obtained (Fig.3-(3)). Then, existing sub-concepts of "disease", such as "myocardial infarction" and "angina pectoris" are classified into sub-concepts of the generated sub-concepts

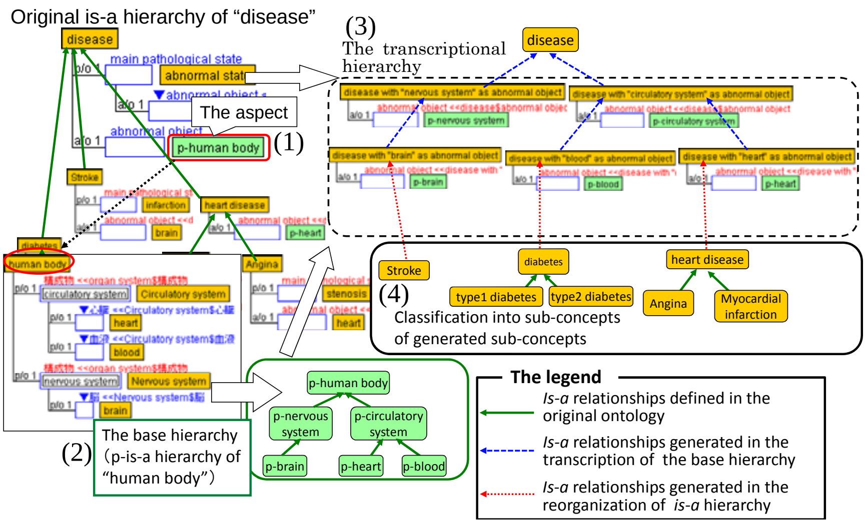


Fig.4 An example of dynamic *is-a* generation of disease in the case that p-is-a hierarchy of human body is selected as the base hierarchy.

on the transcriptional hierarchy through comparisons between definitions of them (Fig.3-(4)). When more than one existing sub-concepts are classified into the same generated sub-concept, they could be organized according to original *is-a* relationships between them. In the case shown in Fig.3-(5), because *is-a* relationships between "disease with hyperglycemia as main pathological state" and "type1/type2 diabetes" can be identified by reasoning, "type1/type2 diabetes" are classified into sub-concepts of diabetes according to the original *is-a* relationships.

§2 In the Case of that an p-is-a Hierarchy is Selected as a Base Hierarchy

In the next example, we suppose the user selects the class-restriction of "abnormal object" as the aspect and the p-*is-a* hierarchy of "human body" as the base hierarchy for a dynamic *is-a* generation of disease in the same ontology with the previous example (Fig.4-(1),(2)).

In the property inheritance mechanism of ordinary *is-a* relationship, when a super class and its sub-class have the same slot, the class restriction of the sub-class's slot must be a sub-class of the super-class's one as well. However, in some case, the class restriction of the sub-class's slot must be a part of the super-class's. For example, when <disease of a pulmonary valve *is-a* disease of heart>, both "disease of a pulmonary valve" and "disease of a heart" have a slot of "site of the disease" and the class restriction of the former must be a part of the latter, that is <pulmonary valve *part-of* heart>.

To cope with such cases, on the basis of our latest theory of roles, we introduced "p-" operator in Hozo which automatically generates a generic concept representing all the parts of the entity to which the operator is attached. The operator enables parts to be inherited by ordinary property inheritance mechanism[5]. In the case of Fig.5, for

[5] To deal with *p-is-a* hierarchies in OWL, we can represent them by some design pattern of ontologies such as SEP triple proposed by Udo Hahn and his group [8].

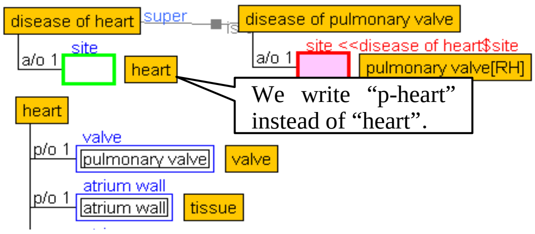


Fig.5 An example of usage of p-operator.

example, we write "p-heart" instead of "heart", and then the slot of its subclass inherits not subclass of "heart" but its parts. When p-X is used, Hozo automatically generates a generic concept representing all of the defined parts of X including all parts which have X as their ancestor. This is valid because each part *is-a* subclass of "X's parts class" which coincides with p-X. According to mereology, the theory of parts, p-X includes itself which is not the very X as an entity but X as its part.

Based on this theory, Hozo automatically generates *is-a* relationships between p-X such as <p-pulmonary valve *is-a* p-heart>. As a result, an *is-a* hierarchy of p-X is generated according to part-of hierarchy of X. The *is-a* hierarchy of p-X is called p-*is-a* hierarchy and could be selected as a base hierarchy for a dynamic *is-a* generation.

In the case of Fig.4, since the class restriction of "abnormal object" is "p-human body", we can select it as an aspect and p-*is-a* hierarchy as a base hierarchy for dynamic *is-a* hierarchy generation. Then, sub-concepts of "disease" such as "disease with p-nervous system as abnormal object" and "disease with p-circulatory system as abnormal object" are dynamically generated according to the aspect and the base hierarchy. As a result, the transcriptional hierarchy of "disease" based on p-*is-a* hierarchy of "p-human body" is obtained (Fig.4-(3)). Then, the existing sub-concepts of "disease" are classified into the transcriptional hierarchy like Fig.4-(4).

In addition to these examples, we can select *is-a* hierarchies which are generated using the proposed method as a base hierarchy to generate another *is-a* hierarchies. That is, our dynamic *is-a* generation could be executed recursively.

The dynamic *is-a* hierarchy generation is applicable to reorganizations of a portion of an *is-a* hierarchy. For example, we can select a middle-level concept (e.g. "disease of heart" as the target concept for the dynamic *is-a* generation.

In these ways, we can dynamically generate *is-a* hierarchies of diseases according to the selected aspects and base *is-a* hierarchies from various viewpoints.

3 Ontological Consideration on Generated *Is-a* Hierarchies

3.1 Three kinds of *Is-a* Relationship

We need to classify *is-a* relationships which appear in *is-a* hierarchies dynamically generated by the proposed method into the following three kinds according to their characteristics.

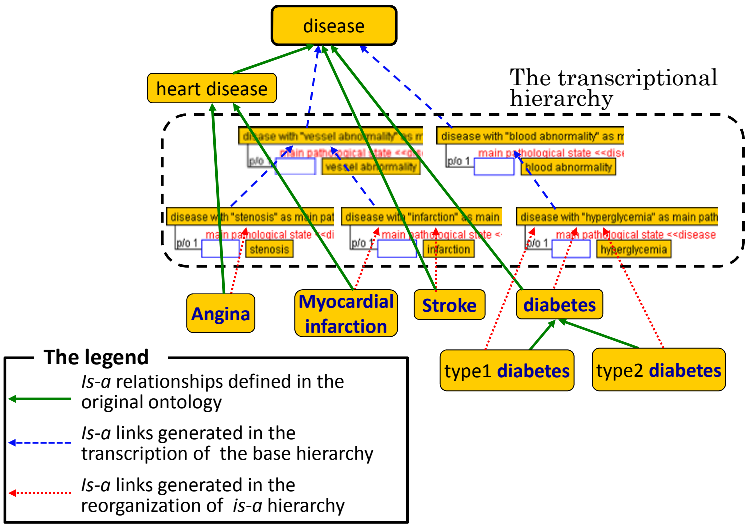


Fig.6 Three kinds of *is-a* relationships in a generated is-a hierarchy.

(A) Is-a relationships defined in the original ontology

They are *is-a* relationships which are defined in the original ontology before reorganizations. They are based on single-inheritance principle, and hence multiple-inheritance is not allowed following the ontological theory

(B) Is-a relationships generated in a transcription of a base hierarchy

They are automatically generated by the system when transcriptional hierarchies are generated according to the selected aspect and base hierarchy.

(C) Is-a relationships generated in a reorganization of *is-a* hierarchy

They are automatically generated by the system between existing sub-concepts and generated sub-concepts.

Note here that (B) and (C) are automatically generated by the system when dynamic *is-a* hierarchy generation is executed, whereas (A) are originally defined in the ontology by its developer.

For instance, Fig.6 shows these three kinds of *is-a* relationships in the generated *is-a* hierarchy illustrated in section 2.3.1. All the existing sub-concepts (e.g. Angina, diabetes) have *is-a* relationships of type (A). Therefore, when the existing concepts are classified into the generated concepts using *is-a* relationships of type (C), they always have multiple-inheritance according to two kinds of relationships of type (A) and (C). In order to identify from which concept an essential property is inherited to each generated concept, we have to distinguish these two kinds of *is-a* relationship in the generated *is-a* hierarchy. In the case of OWL, *is-a* relationships of type (B) correspond to sub-class-of relationships which are determined through classifications by reasoning using a DL reasoner. In the case of Hozo, such kind of *is-a* relationships are represented by *IS-A* relationships[6] which allows inheritance of only properties of super-concepts without identity criterion.

3.2 Consistency of *Is-a* Relationships

Since *is-a* links of type (B) and (C) are automatically generated by the system, there would be a concern about inconsistency of *is-a* relationships in the new *is-a* hierarchy. We investigate whether our method of automatic generation of *is-a* hierarchy causes inconsistencies or not.

§1 Consistency of *is-a* relationships of type (B)

Is-a links of type (B) are automatically generated by the system when generated sub-concepts are defined by specializing the definition of a target concept according to the selected aspect and base hierarchy. That is, only the target concept's definition specified by the aspect is specialized in the generated sub-concepts according to the base hierarchy. It also means that all the generated sub-concepts do not have any inconsistencies with its super-concepts as long as *is-a* relationships in the base hierarchy are consistent. Therefore, *is-a* relationships of type (B) are consistent in the generated *is-a* hierarchy as well.

§2 Consistency of *is-a* relationships of type (C)

Is-a links of type (C) are automatically generated by the system between existing sub-concepts and generated sub-concepts. Both of existing sub-concepts and generated sub-concepts inherit all properties of the target concept. Only when an existing sub-concepts' definition is subsumed by the definition of a generated sub-concept, the existing sub-concept is classified into a sub-concept of the generated sub-concept and an *is-a* link of type (C) is generated between them. Therefore, there is not any inconsistency such as inheritances of unintended properties or undefined properties between them.

The above discussion shows that *is-a* relationships of type (B) and (C) do not cause any inconsistency between the original *is-a* hierarchy and dynamically generated ones nor any change of definitions of existing sub-concepts while they are automatically generated by the system. Furthermore, because the proposed method does not generate *is-a* links between concepts defined in the original ontology, the original *is-a* hierarchy remains after reorganization.

That is, the proposed method enables us to dynamically generate *is-a* hierarchies without causing any inconsistency with the original ontology and changes of original definitions of concepts.

While many redundant concepts could be generated by the method, the user can manage them by specifying the number of the target layers rather than to use all concepts of the base hierarchy for transcription.

---

[6] In Hozo, multi-inheritance is represented by distinction between an *is-a* relationship and *IS-A* relationship in order to identify from which concept an essential property is inherited.

4. Implementation

We implemented a prototype of dynamic *is-a* hierarchy generation system as an extended function of Hozo. The system was developed as a Java client application using HozoCore, which is Java API for ontologies built using Hozo, and Hozo OAT (Ontology Application Toolkit), which is Java library for GUI of ontology-based applications, developed using HozoCore.

The new function consists of three modules: *is-a* hierarchy viewer, viewpoint setting dialog, and dynamic *is-a hierarchy* generation module (Fig.7). The *is-a* hierarchy viewer shows an *is-a* hierarchy of an ontology in a tree representation. The user selects a target concept[7] on the *is-a* hierarchy for a dynamic *is-a* hierarchy generation. The definition of the selected target concept is shown on the viewpoint setting dialog. In the dialog, the user selects a viewpoint for the dynamic *is-a* hierarchy generation by choosing class restriction of a slot as an aspect, a kind of base hierarchy and the number of target layers for a transcription of a base hierarchy according to his/her interests. The dynamic *is-a* hierarchy generation module generates *is-a* hierarchy according to the specified viewpoint. The generated *is-a* hierarchy is shown on the *is-a* hierarchy viewer and could be saved as an ontology file if required.

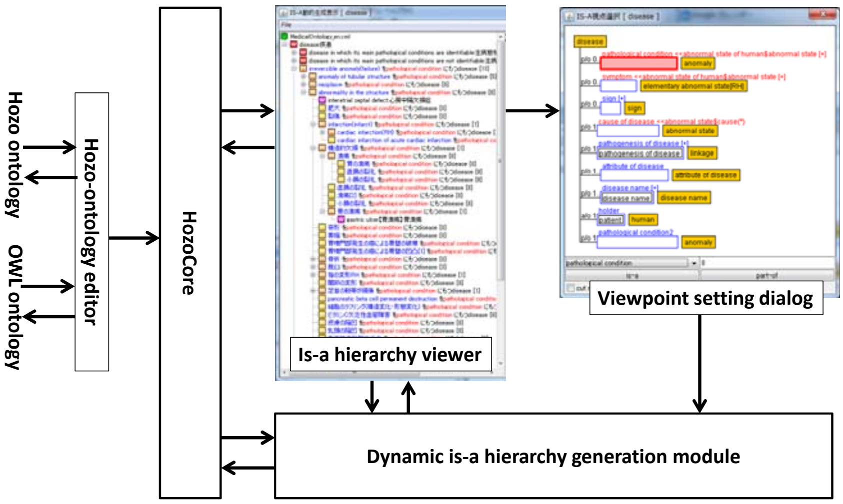


Fig.7 The architecture of the dynamic *is-a* hierarchy generation system.

[7] When the user uses it as the additional function of Hozo, he/she selects a target concept on several GUIs of HOZO for ontology representations.

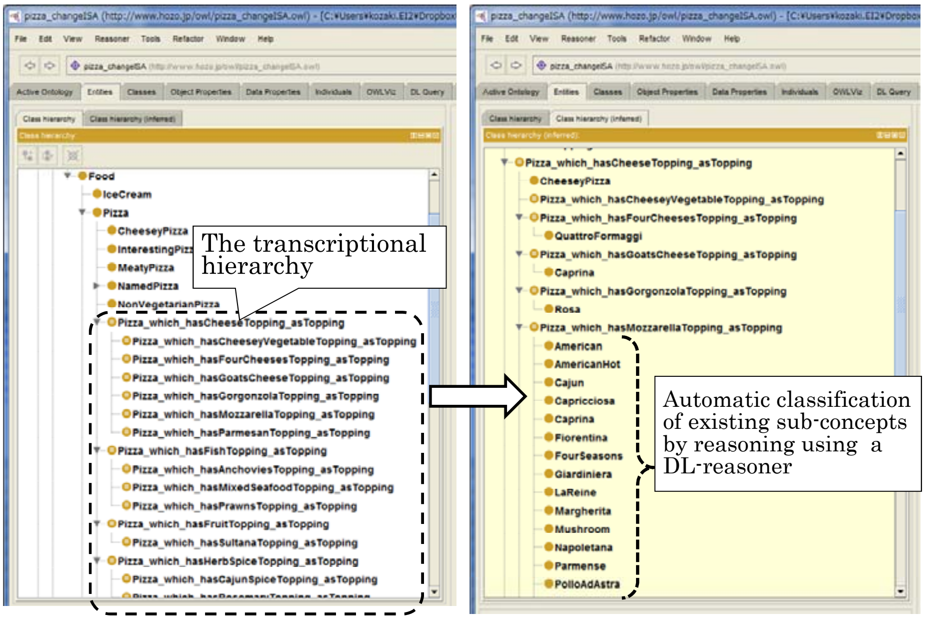


Fig.8 The dynamically generated *is-a* hierarchy from Pizza Ontology exported in OWL format.

While the target of the system is an ontology in Hozo's format, it also can support an ontology in OWL because Hozo can import/export OWL ontologies. When the generated *is-a* hierarchy is exported in the OWL format, its generated sub-concepts in the transcriptional hierarchy are represented by *owl:equivalentClass* which have restriction on properties selected as the aspect. On the other hand, *is-a* relationships of type (C) in the generated *is-a* hierarchy are not exported because they can be identified by reasoning using a DL reasoner. For example, Fig.8 shows a dynamically generated *is-a* hierarchy from Pizza Ontology[8]. As shown in the left of the figure, its transcriptional hierarchy is generated by selecting the restriction on *hasTopping* property as the aspect and the *is-a* hierarchy of *PizzaTopping* as the base hierarchy. The existing sub-concepts of *Pizza* are automatically classified by reasoning using a DL-reasoner like the right of Fig.8.

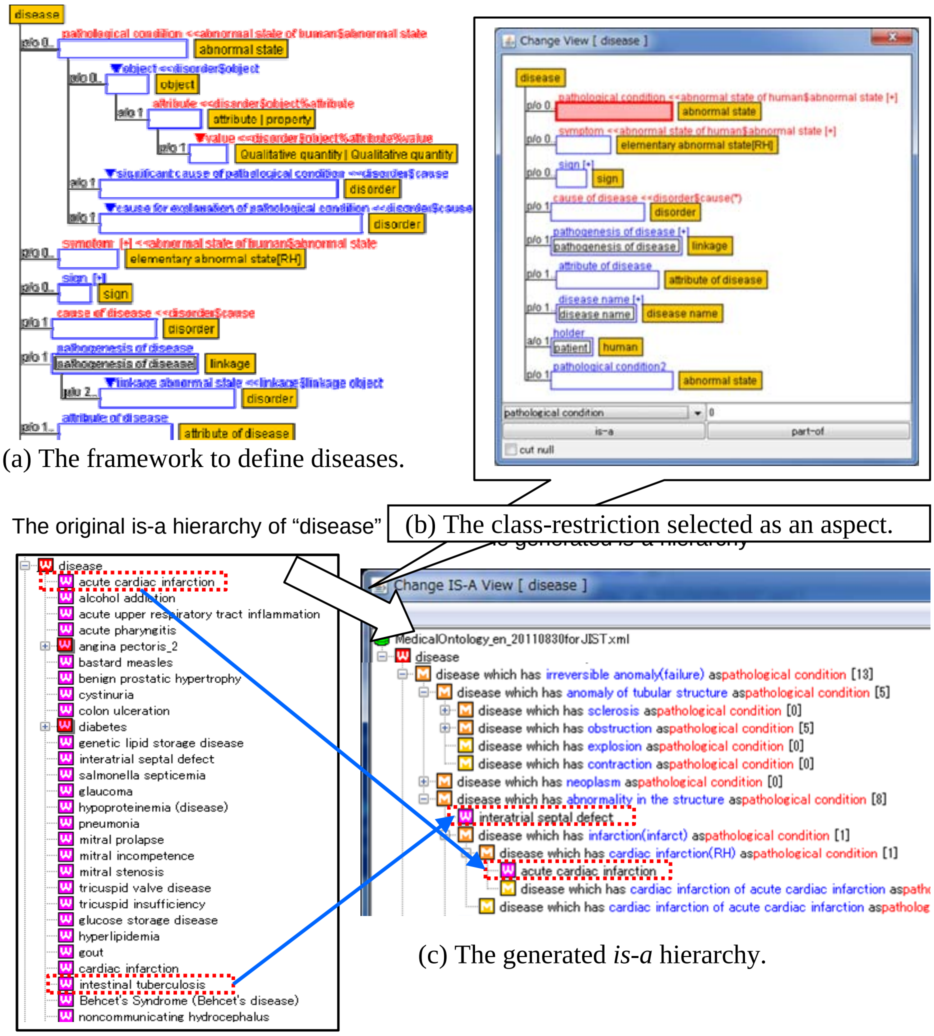
5. Application of Dynamic *Is-a* Generation to a Medical Ontology

We applied dynamic *is-a* hierarchy generation system to a medical ontology which we are developing in a project supported by Japanese government [7, 9]. In our medical ontology, diseases are defined by specifying typical *disorder roles*, such as *pathological condition*, *symptom*, played by *abnormal state*. Fig.9-(a) shows the framework to define diseases. Its *disorder roles* are represented as slots with class-

[8] http://www.co-ode.org/ontologies/pizza/2007/02/12/

restrictions for constraining slot values. These slots are used as aspects for dynamic generation of *is-a* hierarchies of diseases.

For example, when we select the *pathological condition* of disease as an aspect and the *is-a* hierarchy of abnormal state as the base hierarchy, the *is-a* hierarchy of disease is generated (Fig.9-(c)). In the generated *is-a* hierarchy, concepts which have names represented by "disease which has X as pathological condition" (e.g. *disease which has abnormality in the structure as pathological condition*) are sub-concepts generated through the dynamic *is-a* hierarchy generation. Their concept names are automatically determined by the system using a template. Exiting sub-concepts are reorganized as sub-concepts of them. For instance, *acute cardiac infarction* is classified into a sub-concept of *disease which has cardiac infarction as pathological condition*. From the generated *is-a* hierarchy, we can understand diseases according to



(a) The framework to define diseases.

(b) The class-restriction selected as an aspect.

(c) The generated *is-a* hierarchy.

Fig.9 Application of dynamic *is-a* generation to a medical ontology.

the classification of pathological conditions.

On the other hand, when we select the *object of pathological condition* as an aspect and p-*is-a* hierarchy of the *human body* as a base hierarchy, the system generates the *is-a* hierarchy of disease which is similar to the part-whole hierarchy of the human body. For instance, *acute cardiac infarction* is classified into a sub-concepts of *disease which has a pathological condition in the myocardium*.

Although we use a prototype of the medical ontology which includes about 200 diseases in the above examples, about 6,000 diseases have been defined in the current version medical ontology by 12 clinicians. We already have applied dynamic *is-a* hierarchy generation to the latest version and confirmed it could reorganize *is-a* hierarchy of 6,000 diseases. Through dynamic *is-a* hierarchy generation according to the users' viewpoint, they can understands diseases from a variety of perspective. We believe it could contribute to deeper understanding of them.

Moreover, we have developed a medical information system to consider how the dynamic *is-a* hierarchy generation function can be used in other systems [10]. It is used as an index for semantic navigation in the system. We also performed an informal evaluation of the implemented system in a workshop[9] and received favorable comments from medical experts. They especially liked the dynamic *is-a* hierarchy reorganization, which is the first solution to the multi-perspective issues of medical knowledge in the world.

Currently, we are refining the medical ontology based on a new disease model which captures disease as causal chains of disorders [9]. While the dynamic *is-a* hierarchy generation is applicable to the new medical ontology, we need to extend the proposed framework to cope with more sophisticated *is-a* hierarchy generation because we introduce a new kind of *is-a* relationships in the disease model based on an ontological consideration of causal chains.

We also plan to make more formal evaluation of the proposed method. Now, we are collecting several kinds of classification hierarchies of disease, and analyzing what kinds of viewpoints are considered in these classifications. Then, we are evaluating whether out method and medical ontology could support all kinds of them. We also plan applications of the dynamic *is-a* generation to other ontologies.

5. Related Work

In order to avoid multiple-inheritance, some researchers took an approach that they developed ontologies using single-inheritance and reorganized them by reasoning using a DL-reasoner [11]. It corresponds to reorganization of *is-a* hierarchy based on a transcriptional hierarchy in step 4 of the proposed method. However, the approach needs that the transcriptional hierarchy is developed in advance while it is dynamically generated by the system in the case of the proposed method.

Faceted Classification is used to represent classifications from multiple-perspectives. In the Semantic Web, some researchers proposed Faceted Search for semantic portals [12, 13]. They use Faceted Classification according to the user's

[9] The number of participants was about 25. It includes not only the members of the medical ontology development but also others who work in the medical domain.

choose of facets from the definition of ontologies to provide user-centric semantic search. In order to formalize the Faceted Classification, Bene Rodriguez-Castro proposed an ontology design pattern to represent Faceted Classification in OWL [14]. Although the proposed method use a similar technique to Faceted Classification for transcription of a hierarchical structure, it is different from Faceted Classification since we focus on considerations of ontological meaning of generated *is-a* hierarchies. Introduction of a p-*is-a* hierarchy is one of the results of the ontological investigations.

However, there are some rooms to ontological investigate on a method of dynamic *is-a* hierarchy generation. For instance, we need to investigate *is-a* hierarchies of role-concepts and role-holders [5] while this paper concentrated on *is-a* hierarchies of basic concept (normal type). Dynamic *is-a* hierarchy generation based on more complicated viewpoints is also important subject to be considered. For example, we are considering viewpoints to cope with a new disease model based on an ontological consideration of causal chains [9]. Because the latest version of our medical ontology based on the new disease model has more rich definitions than previous one, it would support more complicated viewpoints for dynamic *is-a* hierarchy generation based on causal chains in diseases. We believe these ontological considerations would clarify the feature of the proposed method.

6. Concluding Remarks

In this paper, we discussed multi-perspective issues of *is-a* hierarchy construction in ontologies and proposed a method of dynamic generation of *is-a* hierarchies. The main idea is reorganization of *is-a* hierarchies from the original ontology according to viewpoints of users. Then, we made ontological consideration on *is-a* hierarchies which are developed by the proposed method, and we showed that dynamic *is-a* hierarchy generation does not cause any inconsistency between the original ontology and the generated one. Moreover, we developed a dynamic *is-a* hierarchy generation system as new function of Hozo and applied it to a medical ontology. It enables the users to understand an ontology from various viewpoints according to their intentions.

We plan to investigate further about some issues of the proposed framework for dynamic *is-a* hierarchy generation. Although this paper concentrated on *is-a* hierarchies of basic concept (normal type), we need to consider *is-a* hierarchies of role-concepts and role-holders [5]. Because they can be selected as an aspect for dynamic *is-a* hierarchy generation as well, we also have to consider about such cases. An extension of the proposed frame work to cope with the new disease model based on an ontological consideration of causal chains is another important topic should be considered.

Evaluation of the proposed method is also very important thing which we have to do. We plan to evaluate it through comparison of dynamically generated is-a hierarchies of disease and existing classification of disease.

The demonstration of the dynamic *is-a* hierarchy generation is available at http://www.hozo.jp/demo/. The function is also supported by the latest version of Hozo. Currently, we are also developing the dynamic *is-a* hierarchy generation system for OWL ontologies using OWL-API while it is partly available through OWL import/export function of Hozo.

Acknowledgement

A part of this research is supported by the Ministry of Health, Labor and Welfare, Japan, the Japan Society for the Promotion of Science (JSPS) through its “Funding Program for World-Leading Innovative R&D on Science and Technology (FIRST Program)”, Grant-in-Aid for Young Scientists (A) 20680009.

References

1. Gruber,T.: A translation approach to portable ontology specifications, Proc. of JKAW'92 : 89-108 (1992)
2. Guarino, N: Some Ontological Principles for Designing Upper Level Lexical Resources, Proc. of International Conference on Lexical Resources and Evaluation (1998)
3. Smith, B. et al.:The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration, Nature Biotechnology, 25(11):1251-1255 (2007)
4. Kozaki, K. et al.: Hozo: An Environment for Building/Using Ontologies Based on a Fundamental Consideration of "Role" and "Relationship", Proc. of EKAW2002:213-218, Siguenza, Spain, October 1-4 (2002)
5. Mizoguchi, R. et al.: A Model of Roles within an Ontology Development Tool: Hozo, J. of Applied Ontology, 2(2):159-179 (2007)
6. Kozaki, K., et.al.: Role Representation Model Using OWL and SWRL, In Proc. of 2nd Workshop on Roles and Relationships in Object Oriented Programming, Multi-agent Systems, and Ontologies, Berlin, July 30-31, (2007)
7. Mizoguchi, R. et al.: An Advanced Clinical Ontology, Proc. of ICBO:119-122 (2009)
8. Hahn U, et al. Turning Lead into Gold? EKAW2002, LNCS; vol. 2473: 82-196.(2002)
9. Mizoguchi, R. et al.: River Flow Model of Diseases, Proc. of ICBO2011: 63-70 (2011)
10. Kou, H., Ohta, M. Zhou, J., Kozaki, K., Mizoguchi, R., Imai, T., and Ohe, K.: Development of Fundamental Technologies for Better Understanding of Clinical Medical Ontologies, Proc. of International Conference on Knowledge Engineering and Ontology Development (KEOD 2010):235-240, Valencia, Spain, October 25-28 (2010)
11. Nico Adams, Edward O. Cannon, Peter Murray-Rust: ChemAxiom -An Ontological Framework for Chemistry in Science, Proc. of ICBO:15-18 (2009)
12. Osma Suominen, Kim Viljanen, and Eero Hyvönen: User-centric Faceted Search for Semantic Portals, Proc. of ESWC2007, LNCS 4519 :356-370 (2007)
13. Markus Holi: Crisp, Fuzzy, and Probabilistic Faceted Semantic Search, PhD Thesis, Aalto University, Finland (2010)
14. Bene Rodriguez-Castro, Hugh Glaser, and Les Carr: How to Reuse a Faceted Classification and Put it on the Semantic Web, Proc. of ISWC2010, LNCS 6496:663–678 (2010)