Fundamental Consideration of Role Concepts for Ontology Evaluation

Kouji Kozaki Eiichi Sunagawa Yoshinobu Kitamura Riichiro Mizoguchi

The Institute of Scientific and Industrial Research (ISIR), Osaka University
8-1 Mihogaoka, Ibaraki, Osaka, 567-0047 Japan
+81-6-6879-8416

{kozaki, sunagawa, kita, miz}@ei.sanken.osaka-u.ac.jp

ABSTRACT

There are many approaches for Ontology Evaluation. We take an approach based on ontological theories for ontology content evaluation from them. In this approach, we need some criteria for judging whether an ontology is good or not. In this paper, we make a fundamental consideration of role concepts and show characteristics of role concepts as important and essential criteria for ontology evaluation.

Keywords

Role-concept, Ontology Content Evaluation, Ontology Evaluation Methodology, OWL

1. Intoroduction

As growth of Semantic Web, ontologies play more important role. Currently, a lot of ontologies are available, so developers of semantic web applications have to select suitable ontology for them. In this background, Ontology Evaluation is now one of the key techniques for Semantic Web.

We take an approach based on ontological theories for ontology content evaluation among many approaches for Ontology Evaluation. In this approach, we need some criteria for judging whether an ontology is good or not. For example, OntoClean is a methodology for Ontology Evaluation based on highly general ontological notions form philosophy (essentiality, rigidity, unity and so on) [1]. And it is complemented with an ontology of *particulars* called DOLCE[1] [2].

We especially focus on role-concepts [3, 4, 5] and make a fundamental consideration of role-concepts as important and essential criteria for ontology evaluation. By a **role concept**, we mean a concept of a role which an entity plays in a context. And, by a **basic concept**, we mean the other concept which can be defined without referring to other concepts. For example, role concepts include Teacher, Minister, Symptom, Input/Output, Lerner, Fuel and Food. Then, we strictly distinguish them from basic concepts such as Human, Gasoline and Yogurt. The role concept is discussed based on rigidity in OntoClean [6].

However, it is difficult to conceptualize and represent roles correctly. For example, a parent is often represented by a property such as a *parent-of* property or a *parent* property in RDF(S) or OWL without fundamental discussion of their conceptualization. Furthermore, these representations are often confused with each other although they are actually differentiated from each other. The former is a relation which is conceptualized according to a parent-child relation and represented as a binary relation like "*parent-of* (A, B)". On the other hand, the latter is conceptualized according to a parental characteristic and represented as a unary predicate like "*parent* (A)". Without recognition of such a difference, they are often confused with each other.

Needless to say, a parent is a role concept which is determined according to a manner of a person's participation in a parent-child relation. This conceptualization of a Parent Role is based on clear discrimination of a parent-child relation from a parental characteristic. However, it is not easy to represent this definition only in the framework which most of the ontology description languages provide, since we are often confused by the gap between our recognition of concepts and the conceptual framework of ontology languages. One of the approaches for controlling this problem is to use a framework which helps us to differentiate concepts and to represent such differentiation.

That gap resembles to the gap between a high-level programming language and an assembly language. Every program code in any high-level language can be translated down into an assembly language. The program code in the assembler language, however, does not represent directly programmer's intention about algorithms/data model because it regulates just lower level semantics. In contrast, a program code in a high-level language can represent it more accurately and directly. That is why a high-level language is more familiar to human programmers. Likewise, the frameworks of RDF(S) or OWL provide a framework of ontology representation, and they focus on making bases and common formats to represent definitions of concepts. So, many ontologies can be described in RDF(S) or OWL even if each of them is constructed on their own ontological primitives. However, the frameworks provided by RDF(S) or OWL are not enough for representing higher semantics of concepts. Developers of the ontologies can represent definitions of concepts more naturally and easily in such frameworks that provide higher level semantics. Such higher-level frameworks show ontological characteristics of fundamental concepts in ontologies. Thus, such characteristics can be used as requirements of ontologies and criteria for ontology evaluation.

In this paper, we discuss characteristics of role concepts based on fundamental theories of ontological engineering and present a framework for representing role concepts. Such characteristics of role concepts can be used as criteria for evaluation of representation of roles in ontologies. This paper shows some patterns of possible representations of role concepts in OWL and then evaluates them according to the characteristics as criteria for Ontology Evaluation.

[1] http://wonderweb.semanticweb.org/deliverables/D17.shtml

2. Role-concepts

2.1 Need of differentiation of role-concepts

Context dependence is one of the important characteristics of roles and explains how and why an entity changes its roles to play according to the context it depends on. For example, a Man would be regarded as a Teacher in a School and as a Husband in his Marital Relationship. While such roles can be modeled in connection with time passing, the context-dependence according to the aspect is also necessary semantics for capturing roles property.

Improper modeling of roles will greatly influence the semantics of *is-a* hierarchy of concepts [7]. We focus here on the semantics that an instance of a concept is always recognized also as an instance of its super-concept. For example, in WordNet[2], Dairy Product and Food are treated as hypernyms of Yogurt. If role concepts are not discriminated from basic concepts and these lexical hyponymies among the words are regarded as *is-a* relations among concepts with no distinction, instances of Yogurt are always recognized as instances of Dairy Product and also Food. In such a model, however, we may often have to struggle for faithful representation of events in the real world. To represent that some yogurt has been eaten, we delete the instance of Yogurt. And, it in turn means deleting instances of Dairy Product and Food, which is totally OK. However, in the case where a yogurt has rotted and become inedible, we need to manage instances more sophisticatedly. Because the instance of Yogurt has lost an identity as Food but keeps one as Dairy Product, we can delete only the instance of Food. These managements of an instance model might force us to make different semantics of is-a relation and to establish routines for ad-hoc management of instances. Such a strategy detracts from the value of an ontology, which ensures consistency of an instance model. Moreover, it is difficult in such a model to represent the instance of Yogurt changes its roles to play such as Load, Merchandise, Foodstuff, etc. according to changes of its contexts or aspects. It is advisable for a computer model and an ontology behind it to correspond to the real world as faithfully as possible.

On the other hand, based on fundamental theories of roles in an ontology [3, 6], we can differentiate clearly role concepts (e.g. Food) from the others and can cope with the problems caused by adulterating role concepts and the others. For example, the hyponymy between Yogurt and Food is not regarded as an *is-a* relation. And, we acquire a consistent policy to manage instances of yogurt and food consistently. It is not easy but worth for ensuring quality of an ontology as a backbone of an instance model to differentiate role concepts from others and organize them.

2.2 Roles in our research

2.2.1 Role-concepts and Basic concepts

With citing work by Charles S. Peirce, Sowa introduced the *firstness*, the *secondness* and the *thirdness* of concepts [8, 9]. The *firstness* can be roughly defined as a concept which can be defined without mentioning other concepts. Examples include iron, a man, a tree, etc. In a similar, the *secondness* can be defined as a concept which cannot be defined without referring to other concepts. Examples include a wife, a teacher, a child, etc. The *thirdness* links the *firstness* and the *secondness*. Examples include paternity, brotherhood, etc. Based on these theories, we call one kind of the *secondness* type a **role-concept** in this paper. It represents a role which an entity plays in a context or a label changed according to the context. On the other hand, we treat a concept which is defined without referring to a definition of other concepts as a **basic concept**.

Roughly speaking, by role, we mean what is recognized according to the way of participation of an entity in a context. Because, roles cannot be discussed without their context, we have been focusing on their context-dependencies as essential attributes rather than "player" link to date. The idea of dependency on the context corresponds roughly to "founded" of roles and "Role-of" [6, 10]. And, by basic concept, we mean a thing except roles in order to bring the contrast.

2.2.2 Role-concept, Potential player and Role-playing thing

Here, the authors introduce important distinctions among Role-concept, Potential player and Role-playing thing.

The fundamental scheme in which we capture roles is *"In a context, a **player (class constraint)** can play a **role (role-concept)** and then becomes a **role-holder**."* In the case of school teacher, for example, *"In a school, a **person** plays a **teacher role**[3] and then, becomes a **teacher**."* And, by "teacher", we mean a human who is playing a teacher role. This means that we divided roles into two kinds: **role-concept** and **role-holder** in our terminology. The authors will show that this distinction resolve many of the problems discussed to date.

Role-concept is defined as a concept played by something. By **class constraint** or **role-playable thing**, we mean a potential player: a thing which is able to play a role. In many cases, only basic concepts can be a class constraint. In this example, we say a player of a teacher role is a person. And, when a person is actually playing a teacher role, he/she becomes a teacher role-holder. By a role-holder, we mean a thing of an entity which is playing a role. This means player is divided into two. One is a potential player and the other is a thing which is playing a role. In other words, a player link is divided into two kinds: one is potential player link and the other is is-playing link.

2.3 Characteristics of role-concepts

Role-concepts are recognized dependently on their contexts and an entity changes its roles to play according to the context in which it participates. We call this characteristic of a role-concept as "Context-Dependency". Based on this dependency, we can uncover the following three characteristics which will be explained each of them one by one.

1. Instances of a role concept and a role-holder
2. Categories of role concept
3. Two kinds of role concepts: a primitive and a compound role concept

[2] http://wordnet.princeton.edu/

[3] When we mention a particular role, we put "role" after the role name like "teacher role".

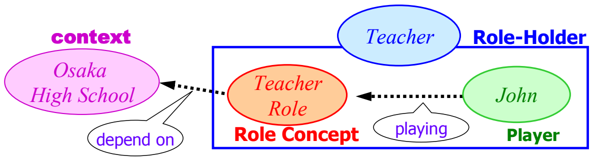


Fig.1 recognition of a role-concept and a role-holder

2.3.1 Instances of a role-concept and a role-holder

Recognitions of a role-concept and a role-holder

Here, let's consider an example:

> In Osaka High School, there is a vacancy on a teacher position. John fills it, and then he becomes a teacher of the school. In our terms, this example is explained and generalized in the following manner.

Firstly, a vacancy on the teacher arises when the teacher role is not played. In our framework, a vacancy is conceptualized as an instance of a role-concept.

Secondly, if Osaka High School does not exist, the instance of the teacher role never exists. So, generally, an instance of a role-concept can not exist without an instance of its context. This dependency is applicable to all kinds of role-concepts.

Thirdly, Teacher Role has two states: played and not played. Because, some properties of the teacher role, for example subject or class, can be determined independently of it is played or not. But name or age of the teacher cannot be done until someone plays. And, when John plays the teacher role, he is recognized as a role-holder (Fig.1).

Disappearance of a role-holder

Next, we consider disappearance of a role-holder. Assume John is a teacher, then, John is no longer recognized as a teacher when the position of the teacher John filled disappeared, John quits the teacher, or John dies. By this example, we can recognize characteristic of a role-holder. In general, a role-holder disappears in these cases: an instance of a role-concept disappears, an instance of a player stops playing the role or an instance of a player disappears.

In these behaviors of the role-holder, we can find its dependency on a role-concept and a player. Here, to explain the dependency in detail, let us focus on composition of the role-holder.

A conceptual framework of role

Fig.2 shows the conceptual framework of role we have proposed. These are properties of teacher role, person and teacher role-holder. They are divided into three groups. Properties in group A are determined by the definition of role-concept itself independently of its player. The second group B is shared by both of the role-concept and the player. And, the last group C is what the role-concept doesn't care about. Role-concept is defined by describing these properties in the context. Its player is defined by oneself. And, the role-holder is defined as a result and eventually includes all of these properties. Therefore, the individual corresponding to teacher is the composite of these two instances and totally dependent on them.

Dependencies of role-concepts based on a theory of part

First topic is discrimination between dependencies of role-concepts based on a theory of part. The facts that an individual corresponding to role-holder is the composite of the instances of role-concept and its player, and that it depends on them, are true to all the cases of Role. But, we think there are two cases concerning the behavior of the instance of role-concept according to the semantics of part-whole property of the context. In one case, there are two independent instances.

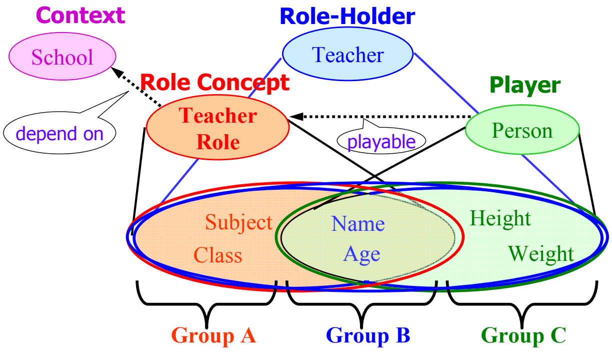


Fig.2 conceptual framework of role

In the example of Teacher, both the instance of teacher role and that of person exist independently of each other.

However, let's investigate the case of Wife. Contrary to the case of teacher, an instance of wife role cannot exist independently of its player because we never recognize marital relation before wife is determined. The appearance and disappearance of wife role totally synchronized with those of its playing relation. This is because of the difference of part-whole semantics between married couple and school which are context of these roles.

2.3.2 Categories of role-concepts

Next topic is classification of role-concepts through their context-dependency. Role-concepts are recognized in a context. So, in order to classify them according to categories of contexts, we can utilize their foundation. For example, task knowledge for solving problem can be discriminated from domain knowledge of a target world. Then, we can identify task-specific roles such as symptom role in a fault diagnostic task and conclusion role in a reasoning task. And, in a functional context in the artifact world, a steering wheel role (played by a wheel) and a level control valve role (played by flow control valve) are classified into a functional role. Note here that we don't claim artifact is role. We believe wheel is a wheel and flow control valve is a flow control valve in its nature. We are claiming that they can play another role according to functional contexts. Likewise, we can classify role-concepts into an action-related role, a relational role and so on. The enumeration is not exhaustive.

2.3.3 A primitive and a compound role concept

The third topic is compound roles. Teacher is recognized not only as a staff member of school but also as a person who teaches students. So, teacher role is composite of school staff role and teaching agent role. Next example is Japanese prime minister. It can be said in our terms that Japanese Prime Minister Role can be played by Japanese Citizen who is played by a human.

In such a manner, some role needs to be played together with other roles. And, in some case, a player stops playing one of the roles, and then, some of others will automatically be un-played. This relation between roles is discussed in other researches as "requirement" [10] and "roles can play roles" [11]. In our

- Task Role
  - Symptom Role (Fault Diagnosis)
  - Conclusion Role (Reasoning)
- Functional role
  - Steering Wheel role (Steering Function)
  - Level control valve role: played by a flow control valve (Function)
- Action-related role
  - Actor role (Any action)
  - Teaching Agent role (Teaching Action)
  - Target object role (Action object)
- Process-related role
  - Product role (Final output)
  - Residue role (How it is processed)
- Relational role
  - Friend role (Friendship)
  - Parent role (Parent-Child Relation)

Fig.3 Categories of role-concepts

framework, we say role-holders can play roles. Such a role-concept depends on multiple contexts.

For example, let us consider peer tutoring context in group learning. A learner is expected to play "peer tutor" role to learn by "learning by teaching" strategy. Peer tutor role depends on both of learning context and teaching context. So, here, we can identify two kinds of roles according to complexity of their context dependencies. They are primitive roles and compound roles. The former has single context dependency. The latter has multiple-context dependency. This is the third characteristic of role-concepts based on context dependency.

Our framework can model compound roles which are understood as "Role-holders can play roles". Fig.4 shows an example of that only Japanese citizen can be Japanese prime minister. A Japanese citizen role is defined dependently on Japanese political system as its context. And, Japanese prime minister role is defined as a role which has to be played in Japanese ministry context, not by a Japanese citizen role but a Japanese citizen, as a role-holder. So, Japanese prime minister depends on these two contexts.

3. Role-concepts in Hozo

3.1 Representation of role-concepts

In order to build good ontologies, we need a tool which provides an appropriate framework based on ontological theories. In these considerations of role concepts, we have developed an ontology building environment, which provides a framework for representation of role concepts and their characteristics. The system is named Hozo[4] [3, 12] and composed of Ontology Editor, Onto-Studio, Ontology Server and Ontology Manager. Users of Hozo can browse and modify ontologies with its ontology editor (in Fig.5). In Hozo, a role concept is defined within the context specified by the basic concept. The system manages some basic concepts as contexts of role concepts and provides a framework to define a role concept. Fig.5-a) shows the form of presentation for definitions of concepts on a browsing pane of Ontology Editor. A role concept is represented as a node connected with the other node representing a concept as its context. The connection is shown as a link representing a *part-of* relation (denoted by "p/o") or a *participate-in* relation (by "p/i") according to the classification of its context. For example, Fig.5-b) represents that a **Person**, who is referred to as the class constraint, plays a **Teacher Role**, and then becomes a role holder a **Teacher**.

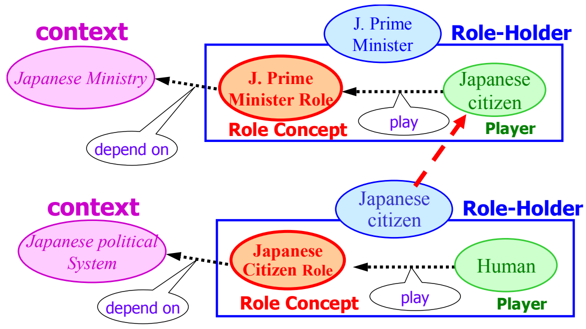


Fig.4 Compound Roles

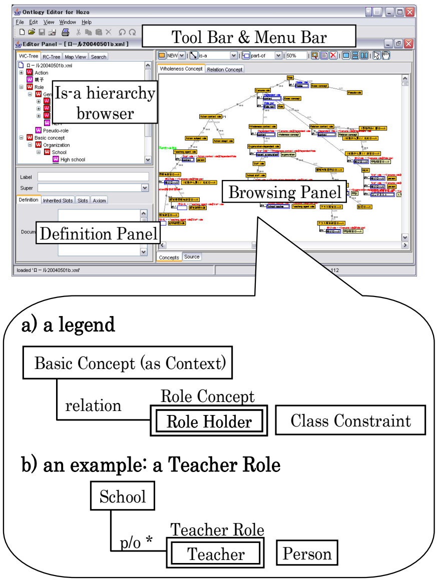


Fig.5 Ontology Editor in Hozo and its form of presentation for definitions of role concepts

3.2 Aggregation of role concepts

In order to represent compound role-concepts, we propose a framework called "Role Aggregation". **Role Aggregation** is a framework for organizing role concepts, which depends on several contexts, according to their essential dependencies. It is represented in two ways, and they have the same semantic information.

To summarize an outline of role aggregation, we here organize an example of role concept which depends on two contexts. At the start, the most essential context is chosen among the two contexts after investigating and decomposing the context-dependence of the role concept[5]. Assume that a **Teacher Role** depends on two

[4] http://www.hozo.jp

[5] The most essential context is decided by developers of an ontology. We do not discuss or conclude generally what the essential context should be. Based on the relativity of essence, we think that, essences of concepts are decided by the developers intended as far as the decision is consistent in the while ontology.

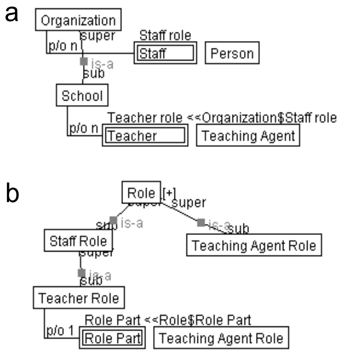


Fig.6 An example of Role Aggregation

contexts: an **Organization** as its essential (primary) context and a **Teaching Action** as its secondary one. And then, two primitive role concepts are identified; a **Staff Role** and a **Teaching Agent Role**. They depend on each of those contexts respectively.

As described in 2.2.2, we can constraint on a class which an instance plays the role. In our previous work, a class constraint refers to only basic concepts. Here, we extend our framework and enable the class constraint to refer to also role holders. In this way, a role holder, which is playing some role(s) already, can play other role(s). It also means aggregating context-dependences of these roles. This role aggregation is represented in the following manner (Fig.6-a); a **Teacher Role** is defined as a specialized concept of a **Staff Role** and a **Teaching Agent** (role holder) is referred to as a class constraint of a **Teacher Role**. Then, a **Teacher Role** is defined as a role concept which depends on both contexts of a **Staff Role** and a **Teaching Agent Role**. Next, we explain role aggregation in a hierarchy of role concepts (Fig.6-b). A role concept which has multiple context-dependencies is classified into a role concept which depends on an essential context. Role aggregation is represented by using a *is-a* relation and a *part-of* relation[6] as the following manner; a **Teacher Role** is defined as a sub-concept of a **Staff Role** through *is-a* relation, and a **Teaching Agent Role** is defined as a part concept of a **Teacher Role** through *part-of* relation. By **Role Part**, we mean a primitive role concept defined as a part of a role concept which has multiple context dependences.

In our framework of role aggregation, an essential context is decided for each role concept. Otherwise, without such a decision, it is possible to merge context dependences also in a framework of multiple inheritances. However, it makes relations among role concepts complicated enormously. So, we do not take multiple inheritances to aggregation of role concepts.

4. Role Concept in OWL

On the basis of the characteristics of role-concepts discussed in Sections 2 and 3, we consider how role-concepts should be represented in OWL. Firstly, we summarize the characteristics as some criteria to compare some representations of roles in OWL for ontology evaluation. Then, we discuss distinctions among some kinds of OWL representations or roles according to the criteria.

4.1 Criteria for ontology evaluation about role-concepts

We summarize characteristics of role-concepts as criteria for ontology evaluation in the following:

Characteristics of role-concepts as a class

1) ***Clear expression of context***: Context dependency is the most important characteristics of role-concepts. And it derives other characteristics. So, clear and explicit expression of context is critical.

2) ***Roles as classes***: A role-concept has characteristics as like owl:Classe It brings many other characteristics.

3) ***Distinction between role-concept and role-holder***: We have discussed this important distinction in 2.2.2.

4) ***Conceptual framework of role-holder definitions***: We have discussed how a role-holder is composed of in 2.3.1 with Fig.2.

5) ***Aggregation of role-concepts***: It is a framework to represent compound roles discussed in 3.2.

Characteristics of role-concepts as instances

6) ***An individual (instance) plays multiple roles***: An individual can play multiple roles at the same time. For example, John (an instance of *person*) can play a teacher role and a husband role at the same time.

7) ***Identity of role-concepts***: Role-concepts have two states: played and not played. And, an instance of role concept has its own identity even if it is not played. The same instance of role concept (with the same identity) can be played by different individuals.

8) ***Individual (instance) of role-holders***: An individual of role-holder is the composite of individuals of a role-concept and a player, and it have two IDs: the ID which is inherited from its role-concept and ID of the player.

9) ***Solution of counting problem*** (distinction between instances of player and instances of role-holder): This distinction solve counting problem because it is possible to count instances of player and instances of role-holder independently.

10) **Disappearance of roles**: We discussed disappearance of a role in 2.3.1.

4.2 Comparison of different representations of role-concepts in OWL

Fig.7 shows some examples of different representations of teacher role in OWL. In order to represent characteristics of role concepts discussed in this paper, we define some properties and classes which are indicated by namespace "hozo:".

In Fig.7-(3), the *hozo:hasPart* property means so-called "part-of relation, and a *hozo:dependOn* property expresses that the class in its domain depends on the class as context. Here, the *hozo:dependOn* property has no restriction on domain and range because role-concepts are defined as sub-class of owl:Class without distinguishing role-concept and basic concept.

[6] Here, we focus on a semantics of *is-a* relation that a sub-concept inherits properties of its super-concept and *part-of* relation that a whole concept possesses properties of its part concepts.

On the other hand, in Fig.7-(4), we define *hozo:BasicConcep* class and *hozo:RoleConcept* class to expresses basic concepts and role-concepts. And so, the domain of *hozo:dependOn* property is a *hozo:RoleConcept* Here, we emphasize that role concepts are dealt with not as an *owl:ObjectProperty* but as an *owl:Class*. They are often defined as a *owl:ObjectProperty* like Fig.7-(1),(2).

A *hozo:playedBy* property represents a relation between classes of role-concept and classes of potential player. Its domain is *hozo:RoleConcept,* and its range is *hozo:BasicConcept*. The definition of *hozo:RoleConcept* has a restriction on this property, and there the property indicates role-playable thing discussed in 2.2. And when a relation between an instance of role-concept and player is represented as a *hozo:playedBy* property, the property means a *playing relation* between them.

And a *hozo:RoleHolder* class represents a role-holder. It is composed of a role-concept and a player, and *hozo:inheritFrom* property expresses its semantics that only definitions (properties) are inherited without identity.

We investigate those representations shown in Fig. 7 from the view point of if it successfully represents the characteristics of role-concepts discussed in section 2. Table.1 shows the result of our investigation. We discuss the difference in detail as follows.

Characteristics of role-concepts as classes

1) ***Clear expression of context***:

In Fig.7-(1), teacher role is represented by "teacheOf" property. It is regarded as a role-concept which depends on "teacher-student relation" (Fig.7-(1)') while the others depend on school, which is the largest difference between this expression and the other three. Even worse, the context-dependency is implicit. It represents other characteristics the same as the representation of Fig.7-(2). So, we discuss about only (2), (3) and (4) in the following.

2) ***Roles as classes***:

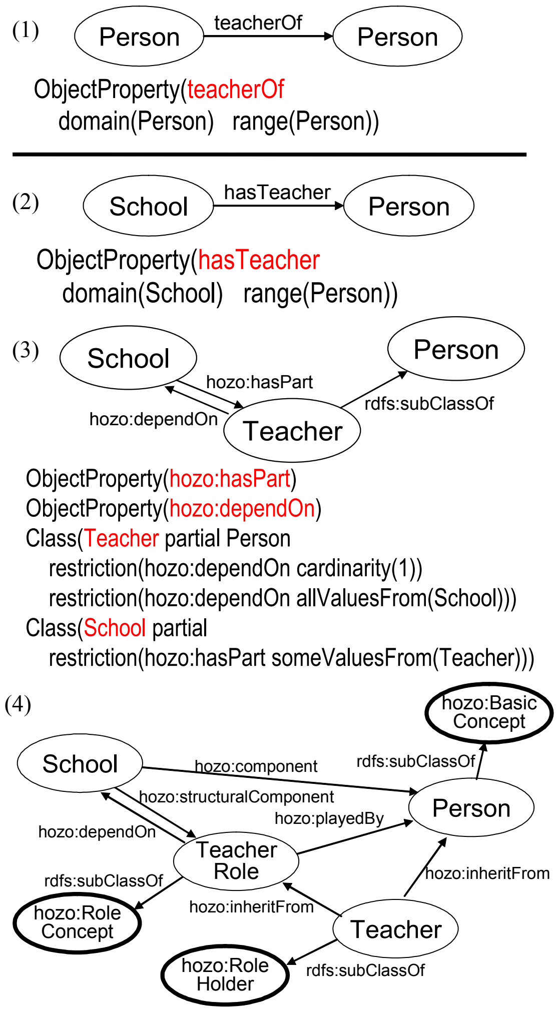
Only (3) and (4) represent the characteristics of role as class.

3) ***Distinction between role-concept and role-holder***

In (2), the property represents role concept, but role-holder is not represented. And in (3), "*Teacher*" class is regarded as a role-holder. But it is not clear if it is distinguished form role-concept.

		(1)	(2)	(3)	(4)
Characteristics of classes	1) Clear expression of context	–	OK	OK	OK
	2) Roles as classes	–	–	OK	OK
	3) Distinction between role-concepts and role-holder	–	–	–	OK
	4) Conceptual framework of role definitions (properties)	–/OK	–/OK	–/OK	OK
	5) Aggregation of role-concepts	–/OK	–/OK	–/OK	OK
Characteristics of individuals(instances)	6) Individual (instance) plays multi roles	OK	OK	OK	OK
	7) Identity of role-concepts	–	–	–	OK
	8) Individual (instance) of role-holders	–	–	OK	OK
	9) Solution of counting problem (distinction between instances of player and instances of role-holder)	–	–	–	OK
	10) Disappearance of roles	OK	OK	–	OK

Table.1 Comparison of different representations of role-concepts in OWL



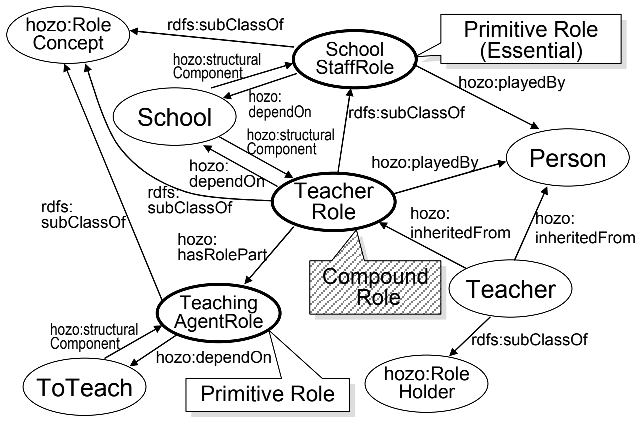
```
ObjectProperty(teacherOf
  domain(Person)  range(Person))
```

```
ObjectProperty(hasTeacher
  domain(School)  range(Person))
```

```
ObjectProperty(hozo:hasPart)
ObjectProperty(hozo:dependOn)
Class(Teacher partial Person
  restriction(hozo:dependOn cardinarity(1))
  restriction(hozo:dependOn allValuesFrom(School)))
Class(School partial
  restriction(hozo:hasPart someValuesFrom(Teacher)))
```

```
ObjectProperty(hozo:dependOn
  domain(hozo:RoleConcept))
ObjectProperty(hozo:playedBy
  domain(hozo:RoleConcept)  range(hozo:BasicConcept))
ObjectProperty(hozo:inheritFrom
  domain(hozo:RoleHolder))
ObjectProperty(hozo:component
  range(hozo:BasicConcept))
ObjectProperty(hozo:structuralComponent
  range(hozo:RoleConcept))
Class(hozo:BasicConcept partial
  DisjointClasses(hozo:RoleConcept hozo:RoleHolder))
Class(hozo:RoleConcept partial
  restriction(hozo:dependOn cardinarity(1))
  restriction(hozo:playedBy maxCardinarity(1)))
Class(hozo:RoleHolder partial
  restriction(hozo:inheritFrom cardinarity(2))
  restriction(hozo:inheritFrom someValuesFrom(hozo:RoleConcept))
  restriction(hozo:inheritFrom someValuesFrom(hozo:BasicConcept)))
Class(TeacherRole partial hozo:RoleConcept
  restriction(hozo:dependOn allValuesFrom(School))
  restriction(hozo:playedBy allValuesFrom(Person)))
Class(Teacher partial hozo:RoleHolder
  restriction(hozo:inheritFrom someValuesFrom(TeacherRole))
  restriction(hozo:inheritFrom someValuesFrom(Person)))
Class(School partial
  restriction(hozo:structuralComponent someValuesFrom(TeacherRole))
  restriction(hozo:component someValuesFrom(Person)))
```

Fig.7 different representations of role-concepts in OWL



```
ObjectProperty(hozo:hasRolePart
   domain(hozo:RoleConcept)
   range(hozo:RoleConcept))
Class(SchoolStaffRole partial hozo:RoleConcept
   restriction(hozo:dependOn allValuesFrom(School))
   restriction(hozo:playedBy allValuesFrom(Person)))
Class(TeachingAgentRole partial hozo:RoleConcept
   restriction(hozo:dependOn allValuesFrom(ToTeach))
   restriction(hozo:playedBy allValuesFrom(Person)))
Class(TeacherRole partial hozo:RoleConcept
   restriction(hozo:dependOn allValuesFrom(School))
   restriction(hozo:playedBy allValuesFrom(Person))
   restriction(hozo:hasRolePart allValuesFrom(Teaching-AgentRole)))
```

Fig.8 Representation of role-aggregation in OWL

4) ***Conceptual framework of role-holder definitions***:

In (2), inheritance of definitions of role concepts can be represented by owl:Restriction. And in (3), it is represented by semantics of inheritance in rdfs:subClassOf. But both of them do not represent properties in Group C (Fig.2).

Characteristics of role-concepts as an instance

5) ***Aggregation of role-concepts***:

Aggregation of role-concept can be represented partially by multiple inheritances in (2) or (3). But they cannot clarify which role is essential for the compound role. In (4), the essential primitive role is represented clearly (Fig.8). A *hozo:hasRolePart* property means "Role-Part" discussed in 3.2 with Fig.6-b.

6) ***An individual (instance) plays multiple roles***:

This can be represented in all representations by multiple properties.

7) ***Identity of role-concepts***:

Only (4) can represent that role-concepts have two states: played and not played by *hozo:isplaying* property.

8) ***Individual (instance) of role-holders***:

Only (3) and (4) can represent an individual of role-holder.

9) ***Solution of counting problem*** (distinction between instances of player and instances of role-holder):

In (3), an instance of Teacher and its player (an instance of Person) are same instance at all time. And, the player cannot stop to be an instance of teacher without stopping to be an instance of Person according to semantics of rdfs:subClassOf. So, in the representation, we cannot count instances of player and instance of role-holder independently.

On the other hand, in (4), instances of player and instances of role-holder are distinguished by associating two IDs: one from the teacher role and the other from player with one individual. Therefore, they can be counted correctly (independently).

10) **Disappearance of roles**:

As stated above, an instance of Teacher cannot stop be an instance of teacher without stopping to be an instance of Person because Teacher is sub-class of Person in (3). In contrast to it, a player (an instance of Person) can stop to play Teacher Role in (4). Moreover, (4) can represent the three cases for disappearance of the instance of role-holder: A) an instance of a role-concept disappears, B) an instance of a player stops playing the role, and C) an instance of a player disappears.

In (2), these disappearances of roles are represented by disappearances of properties.

4.3 Consideration for Content Evaluation

In the above investigation, we have discussed how each representation in OWL can represent characteristics of role-concepts. We believe these characteristics are important criteria for Ontology Content Evaluation because it shows some competency of the ontology to grasp its target world. And, developers of the ontology can evaluate and modify them according to the criteria.

As a first step of Evaluation, we are extending a OWL export funtionality of Hozo to support various representations discussed in 4.2. And we are investigating more detailed representation for role-concepts with SWRL. These will help users to develop their ontologies based on ontological theory of role concepts.

5. Related Work

Guarino and his colleagues aim to establish a formal framework for dealing with roles [6, 7, 10], and the notion is used for Ontology Evaluation in OntoClean methodology. And Gangemi and Mika introduce an ontology for representing a context and states of affairs, called D&S, and its application to roles [13]. Their research is concerned with formalities and axioms of an ontology. In contrast, we do not formalize role concepts because our goal is to develop a computer environment for building ontologies. Our notions of role concepts share a lot with their theory of roles; that is, context-dependence, specialization of roles, and so on. According to their theory, our framework can be reinforced in terms of axioms. They describe between role concepts. The former corresponds to *is-a* and the latter to role aggregation in our framework. However, they do not describe clearly that *is-a* relations between role concepts are established only if the two concepts share the same category of context-dependency. While we have discussed how to define a role concept which has complicated context-dependences, they only point out a requirement relation. Our notions differ from their work on other two points; that is dynamics of a role and clear discrimination of a role from its player (role holder). Firstly, we focus on context-dependence of a role concept and its categories. So, time dependence of a role concept is treated implicitly in our framework because an entity changes its roles to play according to its aspect without time passing. As opposed to this, their framework deals with time-dependency explicitly. Secondly, we distinguish role concepts and role holders [Kozaki 02, Mizoguchi 00]. On the basis of this distinction, we propose a tool for properties and relations on roles, such as an aggregation of role concepts.

Fan also recognizes the importance of constructing a hierarchy of role concepts based on differentiation of them from the others and shows an example in that a Thing is classified into an Entity and a Role in [14]. And, he gives an Agent and an Instrument as sub-concepts of a Role. However, he does not clarify a point of view for organizing them. To our knowledge, they are regarded as being organized according to their manner they participate in their contexts.

Breuker develops ontologies for legal domains based on epistemology and discusses characteristics of roles in [15]. He also mentions adulteration between a role itself and playing role and others between a role and its player. We share his notion in discriminations of these concepts and differentiate a role concept, a class constraint and a role holder from one another [3, 4]. He describes two kinds of roles; as a concept and as a relation. However, he does not organize them in more detail. And, in contrast of that he defines roles according to behavioral requirements and so on, we allow developers of an ontology to define role concepts just as the developers intended because it is outside the scope of our research to discuss how to conceptualize roles.

6. Conclusion

In this paper, we have discussed characteristics of role concepts for Ontology Content Evaluation and showed ten characteristics as important and essential criteria. It helps developers to know some competency of the ontology to grasp its target world.

We have developed an ontology development tool based on these fundamental considerations of role-concept to build well-organized ontologies. Now, we have a plan for a development of Ontology Evaluation Tool based on our tool and ontological theories.

REFERENCES

[1] Janez Brank, Marko Grobelnik, and Dunja Mladeniæ: A SURVEY OF ONTOLOGY EVALUATION TECHNIQUES, in Proceedings of Conference on Data Mining and Data Warehouses (SiKDD 2005), October 17, 2005

[2] Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., and Schneider, L.: Sweetening Ontologies with DOLCE, in Proceedings of 13th International Conference (EKAW 2002), Siguenza, Spain, October 1-4, 2002, 166-181

[3] Kozaki, K., Kitamura, Y., Ikeda, M. and Mizoguchi, R.: Hozo: An Environment for Building/Using Ontologies Based on a Fundamental Consideration of Role" and "Relationship", In Proceedings of the 13th International Conference Knowledge Engineering and Knowledge Management (EKAW2002), Spain, 2002, 213-218

[4] Mizoguchi, R., Kozaki, K., Sano, T., and Kitamura, Y: Construction and Deployment of a Plant Ontology. In Proceedings of 12th International Conference on Knowledge Engineering and Knowledge Management (EKAW2000), Juan-les-Pins, France, October 2-6, 2000, 113-128

[5] Sunagawa, E., Kozaki, K., Kitamura, Y., Mizoguchi, R.: A Framework for Organizing Role Concepts in Ontology Development Tool: Hozo, Papers from the AAAI Fall Symposium Technical Report FS-05-08, Arlington, Virginia, USA, November 4-6, 2005, 136-143

[6] Guarino, N.: Concepts, attributes and arbitrary relations. Data and Knowledge Engineering (8), 1992, 249-261

[7] Guarino, N.: Some Ontological Principles for Designing Upper Level Lexical Resources. In Proceedings of the First International Conference on Language Resources and Evaluation, Granada, Spain, 1998, 527–534.

[8] Sowa, J. F.: Top-level ontological categories. International Journal of Human-Computer Studies 43(5-6), 1995, 669-685.

[9] Sowa, J. F.: Knowledge Representation: Logical, Philosophical, and Computational Foundations.: Brooks/Cole Publishing Co, 2000

[10] Masolo, C., Vieu, L., Bottazzi, E., Catenacci, C., Ferrario, R., Gengami, A. and Guarino, N: Social Roles and their Descriptions. In Proceedings of the 9th International Conference on the Principles of Knowledge Representation and Reasoning (KR2004). Whistler, 2004, 267–277.

[11] Steimann, F.:On the representation of roles in object-oriented amd conceptual modeling, Data & Knowledge Engineering 35, 2000, 83-106

[12] Kozaki, K., Kitamura, Y., Ikeda, M. and Mizoguchi, R: Development of an Environment for Building Ontologies which is based on a Fundamental Consideration of "Relationship" and "Role". In Proceedings of the Pacific Knowledge Acquisition Workshop (PKAW2000), Sydney, Australia, 2000, 205-221.

[13] Gangemi, A., Mika, P.: Understanding the Semantic Web through Descriptions and Situations, In proceedings of International Conference on Ontologies, Databases and Applications of SEmantics (ODBASE 2003), Catania, Italy, November 3-7, 2003

[14] Fan, J., Barker, K., Porter, B., and Clark, P.: Representing Roles and Purpose. In Proceedings of the International Conference on Knowledge Capture (K-Cap2001) Victoria, B.C., Canada, 2001, 38–43

[15] Breuker, J. and Hoekstra, R: Epistemology and ontology in core ontologies: FOLaw and LRI-Core, two core ontologies for law. In Proceedings of the EKAW04 Workshop on Core Ontologies in Ontology Engineering, Northampton shire, UK, 2004, 15-27