Role Organization Model in Hozo

Eiichi Sunagawa, Kouji Kozaki, Yoshinobu Kitamura and Riichiro Mizoguchi

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

Abstract. The establishment of a computational framework of roles contributes effectively to the management of instance models because it provides us with a useful policy for treatment of views and contexts related to roles. In our research, we have developed an ontology building environment, which provides a framework for representation of roles and their characteristics. In this paper, as an extension of the framework, we present a framework for organizing roles according to their context dependencies. We especially focus on defining and organizing compound roles, which depend on several contexts.

Keywords: Role Modeling, Context Dependency, Ontology Development

1 Introduction

Currently, Ontological Engineering attracts a lot of attention in many research areas and has been investigated from various view points; fundamental theory, development, application, and so on. One of the major roles of ontology is to properly represent the underlying conceptual structure of the messy world reflecting the reality as much as possible. All the existing ontology building tools are designed to help people develop such a good ontology. However, few ontology development tools have enough frameworks to provide an advanced framework for ontology description compliant with fundamental theories of ontology.

It is one of the important and essential topics for ontology development to discriminate *role concepts* from the others [1, 2, 3, 4]. By a role concept, we mean a concept of a role itself 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 Lerner, Fuel and Food. Then, we strictly distinguish them from basic concepts such as Human, Gasoline and Yogurt.

However, it is difficult to represent roles in computer properly. 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 in spite of that they have to be differentiated from each other. The former is a relation which is conceptualized according to a parent-child relation and represented also 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 represented as properties in the same manner.

Needless to say, parent is a role which is specified according to a manner of a person's participation in a relation between a parent and a child. This conceptualization of a Parent Role is based on a 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.

In order to represent characteristics of roles and to ensure their semantic interoperability, developers need to specify conceptualization of roles based on a consistent policy for dealing with them. One of the approaches to this issue is to realize a framework which helps to make necessary and sufficient differentiation among concepts and represent them with a high fidelity to conceptualizations by developers. This is why intensive work has been done on OWL representation patterns of *part-of, has-part, attribute, etc.* in the best practice working group[1].

In this background, we have developed an ontology building environment, named Hozo, which provides a framework based on the theory of role concepts and their characteristics [4, 5]. Although Hozo allows users to represent roles better than other existing tools, its theoretical foundation is left unclear and it has some room for improvement concerning the generality of how to deal with roles. In the Hozo framework, role concepts are organized in a basic-concept-centered view and their definitions are scattered around in the respective related concepts which give the context of the roles. This is why Hozo users still have some amount of difficulty in representing relations among role concepts. In this paper, as an extension of the previous framework, we present a framework for organizing role concepts in a hierarchy in the role-centered view. After an overview of the idea of role concept in Hozo, we investigate how to organize role concepts according to their contextual dependencies. We especially focus on defining and organizing a role concept which depends on several contexts, that is, the case where a role plays a role by introducing the idea of role aggregation. And we design a system to realize organization of role concepts. We also discuss related work followed by concluding remarks.

2 Role Concepts

2.1 Needs 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, John is regarded as a Teacher in his School and as a Husband in his Marital Relationship. While such roles can be modeled in connection with time passing, its context-dependence is also necessary semantics for properly capturing roles.

[1] http://www.w3.org/2001/sw/BestPractices/

Improper modeling of roles will greatly influence the semantics of *is-a* hierarchy of concepts. We focus here on the semantics of *is-a* 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 the others 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 an instance 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 the changes of its contexts or aspects. We believe it is advisable for a computer model and an ontology behind it to correspond to the real world as truly as possible.

On the other hand, based on fundamental theories of roles in an ontology [2, 5], we can clearly differentiate 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 Role Concepts in Hozo

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 [1, 6]. 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 way, 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 in this paper.

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

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

corresponds roughly to "founded" of roles and "Role-of" [7, 8]. 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, we introduce important distinctions among Role Concept, Potential Player and Role-Playing Thing (Role-Holder). A ***role concept*** represents a role itself and is defined as a concept played by something. By a ***potential player***, we mean a thing which is able to play a role. It is called also a ***class constraint*** from the view point of that it constraints classes which can be a player of the role. And, while an entity is actually playing the role and behaving as its ***role-playing thing***, the entity becomes a ***role-holder***.

The fundamental scheme in which we capture roles is "In a *context*, a *player (class constraint)* can play a *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*." This means that roles are divided into two kinds: a role concept and a role-holder in our terminology. And, players are also divided into two kinds: a potential player and a role-playing thing (a role-holder). The latter is applied only to relations among individuals.

Fig. 1 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 a 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 does not care about. In the case of a Teacher, his/her Subject and Class can be determined only by its role independently on its player (a Person). On the other hand, a teacher's Name and the Age limit are defined in relation with properties of a Person. And, his/her Height and Weight does not matter for description of a Teacher Role.

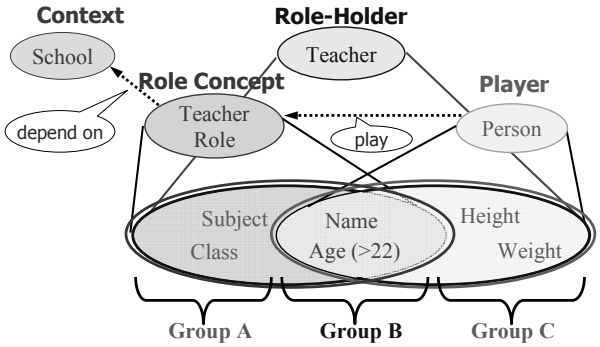

Fig. 1. Conceptual framework of role

In general, a 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

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

eventually includes all of these properties. Therefore, the individual corresponding to a teacher is the composite of these two individuals and totally dependent on them.

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] [4, 5] 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.2). In Hozo, two kinds of basic concepts: a whole concept and a relational concept are defined. And, 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.2-a) shows the form of presentation for definitions of concepts on a browsing panel 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.2-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**.

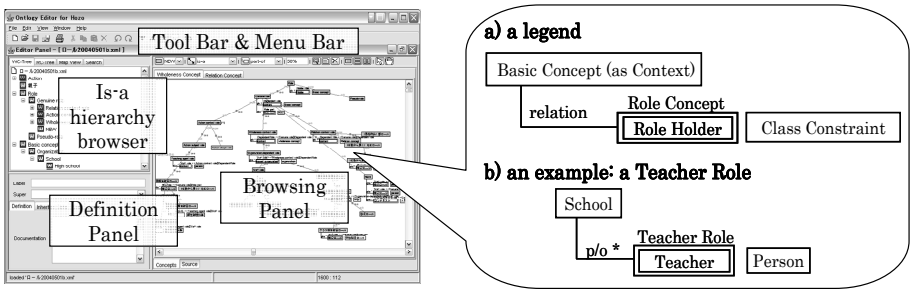

Fig. 2. Ontology Editor in Hozo and its form of presentation for definitions of role concepts

3 Organizing Roles

In this section, we present a framework for organizing roles in an ontology. By organizing roles, we mean mainly constructing a hierarchy of role concepts in order to explicitly grasp and represent relations among them and structures of their context dependences. In the following sections, we explain some considerations as guides to organizing roles through discussions on two hierarchies: hierarchies of basic concepts and role concepts constructed with using Hozo. The hierarchies are composed of some concepts in school[5] (in Fig.3 and Fig.4).

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

[5] This ontology is developed in order to discuss semantics of role concepts. So, it is incomplete and concepts are not defined in detail.

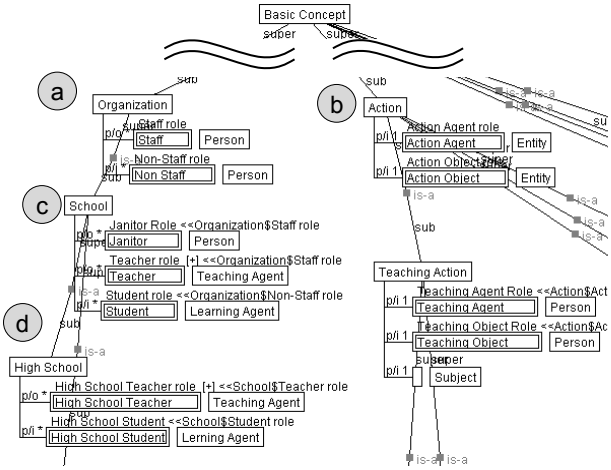

Fig. 3. An example of the hierarchy of basic concepts

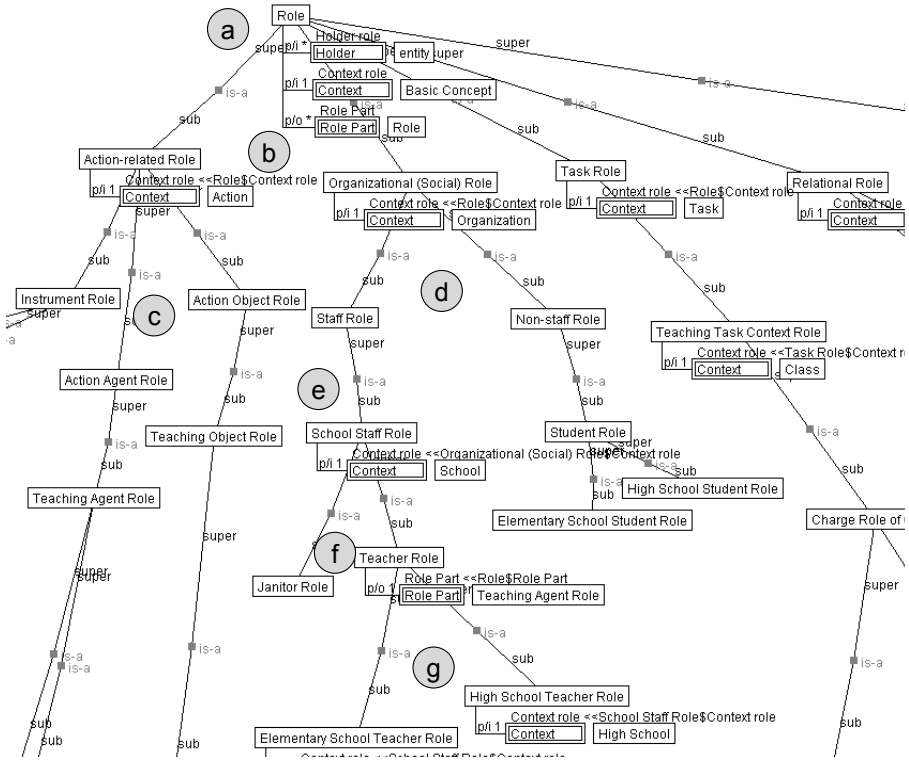

Fig. 4. An example of the hierarchy of basic concepts

To begin with, a **Role** is defined at the top of the hierarchy of role concepts (in Fig.4-a) as a class which has three slots: *context*, *holder*, *role part*. The first is related by a *participate-in* relation and describes in what context the role concept is recognized. The second is also related by a *participate-in* relation and show a basic concept which can carry the role concept. The third is related by a *part-of* relation and associated with role aggregation (described in 3.2).

3.1 Organizing Role Concepts According to Classification of their Contexts

Roles are recognized in a context. So, in order to classify role concepts, we can utilize them as the 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 [9]. Note here that we do not 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. Although the enumeration is not exhaustive, Fig. 5 shows typology of typical roles.

In Fig. 4, an **Action-related Role** and an **Organizational (Social) Role** are defined and classified into a **Role** as a top-level category of the hierarchy (Fig.4-b). The relations among these role concepts describe from the role-centered view that an **Action** and an **Organization** are categorized as contexts at the top-level of the hierarchy of basic concepts (Fig.3-a, b).

- 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. 5. Categories of role concepts

3.2 Aggregation of Role Concepts

Because some roles are conceptualized from several viewpoints and depend on several contexts, it is difficult to organize them simply according to their contexts. For example, a **Teacher** is recognized not only as a **Teaching Agent** but also as a **School Staff**. Such a role needs to be played together with other roles. In other words, some role will automatically become un-played if its player stops to play any one of the other roles. This kind of relation between roles cannot be explained only with a well-known characteristic of roles: an entity can play multiple roles simultaneously. Some other researchers discusses it as "requirement" [7] or "roles can play roles" ("role-holders can play roles" in our terminology) [3].

We differentiate role concepts into two kinds: a ***primitive role concept*** and a ***compound role concept*** which have single-context and multiple-context dependences, respectively. Primitive role concepts can be organized simply and easily according to the categories of their contexts as described in 3.1. In order to deal with compound role concepts, we devise the idea of ***Role Aggregation***. It is represented in both hierarchy of basic concepts and role concepts. And, the two representations have the same semantic information on role aggregation. Fig.6 shows hierarchies extracted from the hierarchies shown in Figs. 3-c and 4-f in order to focus on role aggregation.

Two central purposes of role aggregation are decomposition of a compound role concept and clear decision on its essential context[6]. To summarize an outline of role aggregation, we here organize an example of a compound role concept which depends on two contexts. At the start, the essential context is chosen among the two contexts after investigating and decomposing the context dependence of the role concept. Assume that a **Teacher Role** depends on two 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**. They depend on each of those contexts respectively.

In our framework as described in 2.2, we can constrain on a class whose instances may play certain roles. 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 compound role concept is classified into a role concept which depends on an essential context of the compound role. Role aggregation is represented by using *is-a* relation

---

[6] The decision on essential contexts of compound roles enables to organize them in an *is-a* hierarchy. We do not discuss and conclude what the essential context should be in general. Based on the relativity of essence, we think essences of concepts are decided by the developers intended as far as the decision is consistent in the while ontology.

and *part-of* relation[7] 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 role concept defined as a part of a compound role concept.

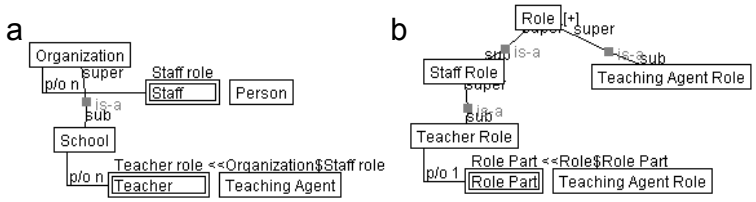

Fig. 6. An example of role aggregation

Our framework for ontology development is based on the consistent policy that an *is-a* hierarchy of concepts are formed according to their essential properties. The policy is indispensable for organizing roles, especially compound roles because, if their essential properties are not determined, they might be just listed in disorder. In this reason, we represent inheritance of dependence on an essential context between roles with an *is-a* relation and discriminate it clearly from the inheritance of other characteristics of the roles. Thus, we did not adopt other methods, say, multiple inheritance, to organize role concepts which might cause an unnecessary disorder to role aggregation.

3.3 Further Considerations for Organizing Roles

After classifications of role concepts according to categories of their contexts (described in 3.1), they are organized in detail. This kind of organizing role concepts is located in a middle layer of a hierarchy of role concepts between top categories of role concepts and aggregated role concepts from the bottom (described in 3.2). Here, we mention three significant points of organizing role concepts.

The first is to organize role concepts according to the aspects of entity playing the roles and manners of its participation into contexts. They are clarified in definitions of the contexts and their categories depend on the definitions. For example, we classified a **Weapon Role** and a **Learner Role** as an **Action-related Role**. And, with investigation of them in more detail, we conclude that the former participates in an action context as an instrument and the latter participates as an agent. Then, we can define an **Action Instrument Role** and an **Action Agent Role** and classify them as subclasses of an **Action-related Role**. In the example of the hierarchy of role

[7] 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. Besides this, the *part-of* relation among roles here represents that a compound role concept (the whole one) can not exists without its role-part(s) (the part one(s)). As far as our discussion, that relation among roles has general common semantics of part-whole relation like transitivity, anti-symmetry and so on.

concepts (in Fig.4-d), a **Staff Role** and a **Non-staff Role** are classified into **Organizational Role** depending on an **Organization** as its context. This classification represent that a **Staff Role** and a **Non-staff Role** are defined as parts of an **Organization** in the hierarchy of basic concepts (in Fig.3-a).

The second is to organize role concepts based on an *is-a* relation between basic concepts as contexts. In general, role concepts related to an *is-a* relation depends on the same category of context. Assume that there are a sub-concept and its super-concept in a hierarchy of basic concepts. A role concept depending on the sub-concept is recognized by specialization of the context of the role concept depending on the super-concept. Then, in a hierarchy of basic concepts, a relation between these role concepts is represented by using overriding. And, in a hierarchy of role concepts, it is represented as an *is-a* relation. In the example of the hierarchy of basic concepts (in Fig.3-d), a **High School Teacher Role** is defined as a part of a **High School** and is recognized by specialization of the context of **Teacher** from a **School** to a **High School**. Then, according to this specialization, in the hierarchy of role concepts (in Fig.4-f,g), <**High School Teacher Role** *is-a* **Teacher Role**> is determined.

The third is also based on an *is-a* relation between basic concepts, but it is shown only in a hierarchy of role concepts. For organizing role concepts appropriately, it is indispensable to define role concepts which cannot be described in a hierarchy of basic concepts. Such role concepts are defined for constraint of contexts as intermediate concepts among role concepts described in a hierarchy of basic concepts. They are used mainly for constraint of contexts and not instantiated directly. We call them ***Abstract Role Concepts*** like an abstract class in an object oriented programming. In the example of the hierarchy of basic concepts (in Fig.3-c), a **Teacher Role** and a **Janitor Role** as parts of a **School** is defined by specializing a **Staff Role** as a part of **Organization**. So, in the hierarchy of role concepts, <**Teacher Role** *is-a* **Staff Role**> and <**Janitor Role** *is-a* **Staff Role**> are held (in Fig.4-d,f). And then, according to <**School** *is-a* **Organization**>, <**School Staff Role** *is-a* **Staff Role**> is described (in Fig.4-d,e). In this case, according to their context dependences, a **School Staff Role** is classified into a **Staff Role** and defined as a super class of a **Teacher Role** and a **Janitor Role** in the hierarchy of role concepts (in Fig.4-e).

4 Instances of Roles

In this section, we discuss what characteristics of instances of role concepts should be represented in their instance model. Instance model provides us with semantics of classes and individuals by specifying their interdependencies concerning their appearance and extinction. It is indispensable for application of ontologies developed with Hozo and clarification of our strategy for treatment of roles to consider the characteristics of the instances of role concepts.

While we have investigated basic issues of role concepts in our previous work [4], it does not include consideration of compound role concepts. So, in this paper, we generalize the framework of role concepts. In the following, **R** denotes a compound role concept, $\mathbf{C_1 \ldots C_n}$ its depending contexts, $\mathbf{R_1 \ldots R_n}$ role concepts aggregated into **R** as its *role-parts* and **P** a concept referred to as the class constraint by **R**. An instance

of **P** can play the role conceptualized as **R**. We explain the framework using an actual example of a **Teacher Role** described in section 3.

(A) States of an instance of a role concept
An instance of **R** has the following two states. (1) Only the role conceptualized as **R** is instantiated (realized). (2) An instance of **P** plays the **R**.

For example, an instance of a **Teacher Role** has two states. One is a teacher role just defined as a part of an instance of **School**. As a vacant position, it is undetermined about who will play it. The other is a role which some person is playing when he/she is recognized as a **Teacher** (role holder).

(B) Dependence of instances of role concepts on their context
An instance of **R** exists only if all instances of $\mathbf{C_1 \ldots C_n}$ are instantiated. When, at least, one of them is deleted, so does the instance of **R**.

For example, a **Teacher Role** is instantiated and a **Teacher** is recognized, on the assumption that a **School** and a **Teaching Action** are instantiated. When the school is closed down or when a teaching class is finished, an instance of a **Teacher Role** is deleted.

(C) Dependence of instances of role concepts on their players
In general, an instance of a role is dealt with as a incomplete instance until an entity plays the role because roles can not behave without its players. When instances of **R1…Rn** as constituents of **R** are played by the same instance of **P**, a role holder of **R** is recognized with being composed by an instance of **R**.

For example, when someone is employed as a staff by a school and he/she teaches, all values or ranges of properties of **Teacher** (role holder) are fixed. Then, a **Teacher Role** can be instantiated and he/she is recognized as a teacher.

(D) Extinction of a role holder
A role holder of **R** is recognized as the summation of both instances of **R** and **P**. Here, they are denoted **Ri** and **Pi**. Then, there are four cases in which the role holder is disappear: (1) **Pi** has been disappeared. (2) **Ri** has been disappeared. (3) **Pi** has stopped playing **Ri**. (4) At least, one of role holders of $\mathbf{R_1 \ldots R_n}$ is disappeared.

For example, there are three cases in which a person is not recognized as a Teacher. They are (1) when he/she has died, (2) when the post he/she filled has disappeared because of closing down his/her school, personnel reduction and so on, (3) when he/she has retired his/her job as a teacher and (4) when his/her teaching class has been finished.

5 Implementation

As an extension of the ontology editor in Hozo, we provide a pane for constructing a hierarchy of role concepts and function to support organizing role concepts.

We add a pane for building and editing a hierarchy of role concepts to the ontology editor in a line of panes for basic concepts provided previously (Fig.7). The pane provides almost the same functions as those of the panes for basic concepts. And, we improve the ontology editor to support organizing role concepts in the strategies described in section 3. Firstly, we extend the framework to define concepts for representation of role aggregation. Secondly, we add a function for keeping consistency between role concepts defined in the hierarchies of basic concepts and those defined in the one of role concepts. This function is based on the fact that some parts of the role concepts defined in both of the hierarchies share the common semantics. For example, if a developer aggregates role concepts in a hierarchy of role concepts, this aggregation is represented automatically also in a hierarchy of basic concepts. And, we provide some wizards for organizing role concepts. They support operation to deal with role concepts and guide ontology developers.

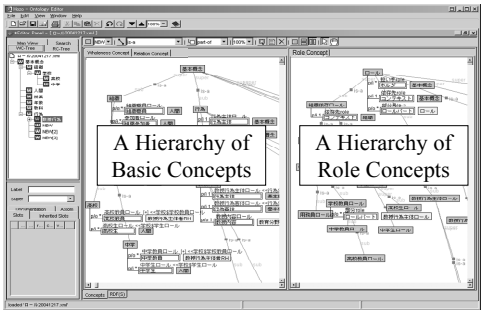

Fig. 7. Panes for building and editing hierarchies of basic concepts and role concepts in Ontology Editor

Besides this, we are investigating how to deal with characteristics of roles in OWL. See Fig.8 and Fig.9. We define *hozo:BasicConcept* 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*. A *hozo:playedBy* property represents a relation between classes of role concepts and classes of their potential players. 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 a role concept and its 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. And, at the present, we are trying to clarify and regulate behaviors of roles with using SWRL (Semantic Web Rule Language) in more detail. The model presented here is utilized for extension of the function for exporting ontologies which are developed with Hozo in OWL.

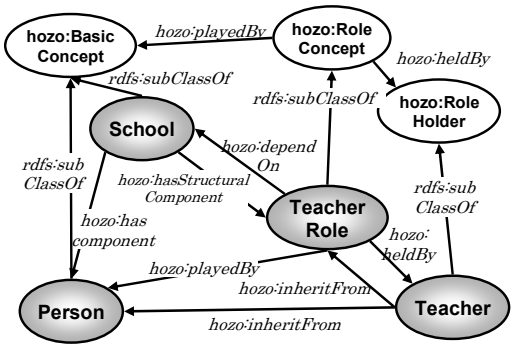

Fig. 8. Role representation in OWL

```
ObjectProperty(hozo:dependOn
    domain(hozo:RoleConcept))
ObjectProperty(hozo:playedBy
    domain(hozo:RoleConcept)
    range(hozo:BasicConcept))
ObjectProperty(hozo:inheritFrom
    domain(hozo:RoleHolder))
ObjectProperty(hozo:hasComponent
    range(hozo:BasicConcept))
ObjectProperty(hozo:hasStructuralComponent
    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)))
   restriction(hozo:heldBy 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:hasStructuralComponent
someValuesFrom(TeacherRole))
    restriction(hozo:hasComponent someValuesFrom(Person)))
```

Fig. 9. Role representation in OWL (Abstract Syntax)

6 Related Work

Guarino and his colleagues aim to establish a formal framework for dealing with roles [2, 7]. And Gangemi and Mika introduce an ontology for representing a context and states of affairs, called D&S, and its application to roles [10, 11]. 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 specialization and requirements as kind of sub-class relations 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 [4, 5]. 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 [12]. 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 [13]. 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 [4, 5]. 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.

7 Conclusion

In this paper, we have discussed a framework for organizing role concepts in a hierarchy according to their context-dependences. Then, we investigated instances of role concepts to give semantics of role-related concepts. Although it was not discussed explicitly, our framework solves the so-called counting problem and universal/individual problem of roles. The definitions of role concepts can be translated into statements in OWL. In conclusion, our framework in Hozo provides a layer in which developers can construct ontologies with high quality description of role concepts and a mechanism for setting it in the current linguistic expression. As

future work, we plan to implement the framework in Hozo and investigate a theory of organizing role concepts (e.g. semantics of *is-a* relation between role concepts).

References

1. Sowa, J. F.: Top-level ontological categories, International Journal of Human-Computer Studies, Vol.43, Issue 5-6, pp.669-685 (1995)
2. Guarino, N.: Some Ontological Principles for Designing Upper Level Lexical Resources. In Proceedings of the First International Conference on Language Resources and Evaluation, pp.527–534, Granada, Spain (1998)
3. Steimann, F.: On the Representation of Roles in Object-oriented and Conceptual Modeling, Data & Knowledge Engineering, Vol.35, Num.1, pp.83-106 (2000)
4. 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), pp.213-218, Sigüenza, Spain (2002)
5. 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 2000 Pacific Knowledge Acquisition Workshop (PKAW2000), pp.205-221, Sydney, Australia (2000)
6. Sowa, J. F.: Knowledge Representation: Logical, Philosophical, and Computational Foundations (2000)
7. Masolo, C., Vieu, L., Bottazzi, E., Catenacci, C., Ferrario, R., Gangemi, 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), pp.267–277, Whistler, Canada (2004)
8. Loebe, F.: Abstract vs. Social Roles - A Refined Top-Level Ontological Analysis. Papers from the AAAI Fall Symposium Technical Report FS-05-08, pp.93-100, Virginia, USA (2005)
9. 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, pp.113-128, Juan-les-Pins, France (2000)
10. Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schneider, L.: Sweetening Ontologies with DOLCE, In Proceedings of the 13th International Conference Knowledge Engineering and Knowledge Management (EKAW2002), pp.166-181, Sigüenza, Spain (2002)
11. Gangemi, A., Mika, P.: Understanding the Semantic Web through Descriptions and Situations. International Conference on Ontologies, Databases and Applications of SEmantics (ODBASE 2003), Catania, Italy (2003)
12. Fan, J., Barker, K., Porter, B., and Clark, P.: Representing Roles and Purpose. In Proceedings of the International Conference on Knowledge Capture (K-Cap2001), pp.38–43, Victoria, B.C., Canada (2001)
13. 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, pp.15-27, Northamptonshire, UK (2004)
14. Kozaki, K., Sunagawa, E., Kitamura, K. and Mizoguchi, M.: Fundamental Consideration of Role Concepts for Ontology Evaluation, In Proceedings of the 4th International EON (Evaluation of Ontologies for the Web) Workshop (to appear) (2006)