

Instance Management Problems in the Role Model of Hozo

Kouji Kozaki, Satoshi Endo, and Riichiro Mizoguchi

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

Abstract. For knowledge (instances model) representation based on ontology and its use, it is desirable to understand phenomena in the target world as precisely as possible. The theory of ontology should reflect the understanding of them and provide a fundamental framework to manage the behavior of instances adequately. Hozo is known as an ontology-development tool with an ability to deal with roles and their instances. Although Hozo allows users to represent roles better than other existing tools, the underlying theoretical foundations are still unclear and there is some room for improvement concerning the generality of how to deal with instances of roles. Especially, establishment of the instance management method of role concept based on the ontological theory is an important subject. It is mainly concerned with handling of appearance and disappearance of role instances. This article discusses a refinement of role theory through investigation of problems such as distinction between constituent roles and general roles and what kind of constituent roles are needed in what situations are also discussed in detail.

Keywords: Ontology, instance management, role theory, ontology modeling.

1 Introduction

Ontology has been used as the basis of knowledge systems in various domains, and its utility is recognized more widely day by day. An ontology provides “an explicit specification of conceptualization” [1] underlying any knowledge representation (an instance model), and it is one of the important roles to keep the consistency and reusability of knowledge by describing them based on the ontology. Many researchers study ontological theories intended to contribute to building a well-founded ontology. Especially, theory of roles is one of the critical topics. The world is full of roles. Roles have various characteristics such as anti-rigidity [2], dynamics [3], context dependency, and so on. For example, by dynamics, we mean that the role which a person plays can be change dynamically according to the focal context. (e.g., a man may be a teacher in a school and a husband in a married couple.) We have been investigating these characteristics of roles and how to deal with them on computer systems as accurately as possible. As a result, we have developed an ontology development/use tool, named Hozo, based on fundamental consideration of roles [4]. However, although

some researchers discuss ontological theories of roles, there remains some room for investigations of instance management problems such as the counting problem [5], appearance/disappearance of instances of roles, dynamic change of roles which players play, and so on. It is important to establish an ontological theory for instance management of roles so that we can capture their behavior and manage them in a sound manner. In addition, there is no tool which implements role theory besides Hozo in the world.

This paper discusses these instance management problems related to roles in a generic framework, and we refine our role model to build and manage instances of roles appropriately. Furthermore, we think that they are general problems in role theories and are not limited to the role model in Hozo. The next section summarizes three instance management issues of roles which are discussed in the following sections. Section 3 discusses our previous role model and dependency of an instance of a role on its context. Section 4 discusses the main issue we investigate in this paper, and we propose distinction between *constituent role* and *post role* to solve the instance management concerned with promotion. Section 5 gives some discussions about identities of roles under consideration. Related work is discussed in Section 6, followed by concluding remarks.

2 Instance Management Problem of Roles

A role is defined as “a name of entity which changes according to contexts” or “an entity that is played by another entity in a context” [4]. For example, when an instance of a man is playing a teacher role in a school, it means that he is a teacher. And the man would be regarded as a teacher in the school and as a husband in his marital relationship. In this example, *teacher* and *husband* are roles, and they differ from so-called natural types (e.g. human).

These characteristics of roles can explain various behaviors of an instance in the real world. The following shows typical ones:

(1) Dependency of an instance of a role on its context

We assume *an instance of human*, Taro, who is a *teacher* in Osaka high school. If the school is closed down, he stops being a teacher in the school while he is still an instance of human. This example says the existence of an instance of a *role* (e.g. teacher) is dependent on that of its *context* (e.g. school).

(2) Continuity of instances of roles

We assume Jiro is an associate professor of Osaka University. When he promotes from associate professor to full professor, his post changes while he remains to be a teaching staff in the university. It implies that some instances of roles continue to exist while others change when the player changes roles to play.

(3) Identity of instances of roles

When the Prime Minister of Japan changes from Abe to Fukuda, we can regard that they play the same role as the head of the Japanese Government. However, we also can recognize they play different roles (e.g. 90th and 91th Prime Ministers). The details of these issues and its solution are discussed in following sections.

3 Role Model

3.1 Fundamental Schema of Our Role Model

The fundamental scheme of our roles at the instance level is the following (see the lower diagram in Fig. 1.):

“In Osaka high school, Taro plays teacher role-1 and thereby becomes teacher-1”

This can be generalized to the class level (see the upper diagram in Fig. 1)

“In schools, there are persons who play teacher roles and thereby become teachers.”

By **play**, we mean that something “acts as”, that is, it contingently acts as according to the role (role concept). By **“teacher”**, we mean a class of dependent entities which roughly correspond to persons who are playing teacher roles and which are often called *qua individuals* [5]. Here, we introduce a couple of important concepts to enable finer distinctions among role-related concepts: **role concept**, **role holder**, **potential player** and **role-playing thing**. In the above example, these terms are used as *“In a context, there are potential players who can play role concepts and thereby become role holders”*.

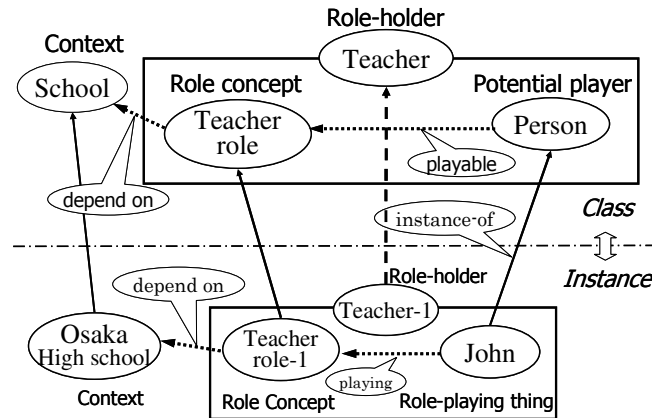


Fig. 1. Fundamental scheme of a role concept and a role holder

By **context**, we mean a class of things that should be considered as a whole. A context includes entities and relations. **Role concept** is defined as a concept whose entities are played by some entity within a context. So, it essentially depends on the context. By **potential player**, we mean a class of things which are able to play an instance of a role concept. In many cases, basic concepts (natural types) can be used to denote classes of **potential players**. When an instance of potential player is playing the instance of role concept, we call the instance a **role-playing thing**. In this example, we say a person can play an instance of a teacher role. In particular, Taro is actually playing a specific teacher role *teacher role-1*. By doing so, he/she is associated with the instance *teacher-1*, an individual teacher **role holder**. A role-holder class is a class of dependent entities like teacher-1. As such, it is neither a specialization of a potential player class (e.g., person) nor that of a role concept class (e.g., teacher role),

but an abstraction of a composition of a role-playing thing and an instance of role concept, as is shown in Figure 1, which is the heart of our Role model. The link from Teacher-1 to Teacher is a broken arrow rather than a solid one like instance-of link to show the relation is not completely same as instance-of relation in Fig. 1. Our model and tool do not allow people to directly instantiate role holder classes because the individual role holder as a dependent entity to be instantiated inherently requires first an instance of a potential player class and of a role concept class. Then, when the playing link is asserted, it virtually acquires the properties of the potential player and the role concept. This is why role holders are dependent entities.

All the concepts introduced here are core of our role model and contain rich implications which are elaborated in [4]. The above shows that we divide the conventional notion of “Role” into two kinds: role concept and role holder in our model. Therefore, our model of roles does not have the concept of “Role” explicitly. In particular, it is understood conventionally that a role existing at the instance level must be something being played by something, since people understand the role instantiation and the action of playing the role as happening at the same time. In contrast, in our model a role concept can exist at the instance level without being played, since it depends only on its context and not on its player. While the concept of role is the target of the ontological research on roles, at the same time, this term has been the source of confusion, since it hides the difference between role concept and role holder.

3.2 Hozo’s Representation Our Role Model

Figure 2 shows the correspondence between the model and the corresponding Hozo representation. Because Hozo is based on frames, the representation is rather straightforward. Additionally, we discussed theoretical solid foundation and formal definitions of our role theory in previous work [4]. In the paper, we discussed the solid foundation of role model and presented its semantics using OWL to clarify its formal definitions. The details of role representation model using OWL and SWRL are discussed in [6]. Hozo also can export ontologies in OWL.

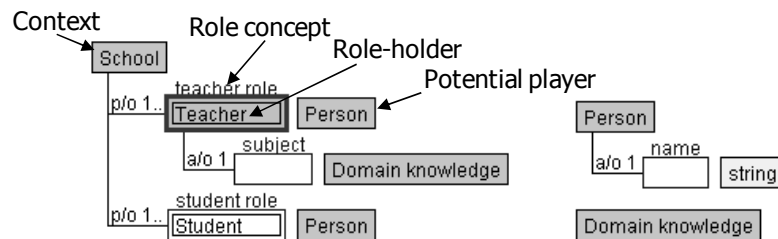


Fig. 2. Hozo’s representation our role model

Let us explain Hozo’s representation conventions by using the example shown in Fig. 2. In Hozo each concept defined as a class is represented in a rectangle like *School* and *Person*. Each class is defined by specifying its parts (denoted by “p/o”) and/or attributes (denoted by “a/o”) as slots. *School* is here defined as an entity composed of teachers and students where *teacher role* and *student role* are role concepts

played by individuals specified by the rectangle at the far right, instances of *Person* in this case. As shown in Figure 2, the key idea of class definition in Hozo is that all concepts, which can theoretically be parts of something, are defined independently of the possible wholes they belong to, and each class as a whole is defined by specifying the roles whose parts play. In other words, all the class definitions in Hozo are reciprocal, in the sense that a whole (School) is defined in terms of its parts (Person) playing their own roles, and at the same time, the roles (teacher role) played by the parts (Person) are defined there under the context of the whole (School).

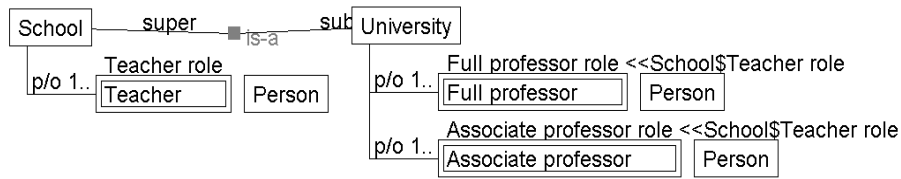


Fig. 3. Representation of the hierarchy of role concepts

Is-a (super-sub) relations between basic concepts are represented by *is-a* links as shown Fig. 3. In this example, *University* is defined as a sub concept of *School*. Sub concepts inherit all role concepts from their super concepts, and sometimes they specialize inherited role concepts to define the role concepts in their context. Fig. 3 shows two role concepts (*full*) *professor role* and *associate professor role* which are defined in the context of *University*. They are sub concepts of *teacher role* in *School*. The relationships between these role concepts are represented by describing the super concepts on the right of role concepts with double angles “<<” as shown in Figure 3. It means the hierarchy of role concepts is analogue to the hierarchy of basic concepts because all role concepts are defined within the basic concepts as their contexts.

3.3 Instances of Role Concepts

The example of the teacher discussed in section 2-(1) can be elaborated and generalized in the following manner. Firstly, if *Osaka High School* does not exist, the instance of the teacher role never exists. In general, any instance of a role concept cannot exist without an instance of its context. This dependency applies to all types of role concepts. Secondly, a vacancy in a teacher post arises when the instance of the *teacher role* is not played. Such a vacancy supports the existence of the role concept. Furthermore, it means that the role concept has two states: played and not played. It can exist in the un-played state because some values of some properties including those of the essential properties of the role concept (for example, in the case of the *teacher role*, *subject*, *class*, and so on) can be determined independently of whether it is played or not. But *name* or *age* of the *teacher* cannot be determined until someone plays it. Thirdly, Taro is no longer a teacher when the teacher position he fills disappears, when he quits the *teacher role*, or when he dies. In general, an individual role holder disappears in the following cases: an instance of the role concept disappears, an instance of the player stops playing the role or an instance of the player disappears. This is understood because that an individual role holder is dependent on the

individuals of a role concept and of its player as far as the playing relation is valid as discussed in Section 3.1. This observation suggests that the identity (ID) of the individual of the role holder is a function of the IDs of the role concept (ID_{Role}) and of the player (ID_{Player}). That is, $ID_{Role\ holder} = f(ID_{Role}, ID_{Player})$ in which both arguments are mandatory for $ID_{Role\ holders}$, and in which “f” is bijective (surjective and injective).

Here, we generalize the characteristics of instances of role-related concepts. An instance model specifies the interdependencies between classes and individuals, especially concerning the appearance and extinction of individuals. It appears as indispensable for the concrete application of ontologies, and for a clarification of the nature of role instances. In the following, **R** denotes a role concept, **C**¹ the contexts it depends on, and **P** is a concept considered as the potential player of **R**.

(A) Dependence of instances of role concepts on their context

An instance of **R** exists if (and only if) an instance of **C** is instantiated. When the instance of **R** ceases to exist, so does the instance of **R**.

(B) Dependence of instances of role concepts on their players

An instance of **R** is dealt with as a defective instance by itself. When the instance of **R** as constituents of **R** is played by an instance of **P**, **R** is concretized to be a complete instance corresponding to **R**.

(C) Extinction of a role holder

A role holder of **R** is composed of both instances of **R** and **P** by combining all of their slots. Let r and p denote instances of **R** and **P**, respectively. Then, there are three cases in which the individual role holder disappears: (1) p disappears, (2) r disappears and (3) p stops playing r .

4 Instance Management Problem Concerned with Promoting

4.1 Instance Management Concerned with Promotion

For example, we consider an *associate professor role* and a *full professor role* which are defined in a university. In this example, an associate professor Jiro in Osaka University is represented as follows:

Jiro plays an instance of the associate professor role and thereby becomes an associate professor (role holder).

Now, we assume a case where Jiro is promoted from associate professor to full professor. It means that Jiro stops to play the instance of the *associate professor role*, then plays the instance of the *full professor role* and thereby becomes a full professor (role holder). In this example, when Jiro plays the instance of associate professor role or full professor role then, he also plays *teaching staff role* in the university at the same time. Because “full professor role *is-a* teaching staff role” and “associate professor role *is-a* teaching staff role” as shown Figure 3), the semantics of *is-a* relation tells us that Jiro stops to play the instance of teaching staff role at the very moment when he changes the role to play. This is because one has to stop to play the current role

¹ Our role model also supports role concepts depending on multiple contexts [4].

when he/she starts to play the new role. In other words, the continuity of playing the *teaching staff role* is damaged. Obviously, this example model does not capture the behavior of the instances in the real world accurately at all. In the real world, Jiro has been the same teaching staff not only when he is the associate professor but also the full professor.

4.2 Solution

The problem discussed in the previous section is caused by the confusion of *teaching staff role* between *full professor/associate professor roles*. The former means that the player of the role belongs to the university as a staff member, and the latter means a role (post) that its player performs in the university.

Therefore, this problem can be solved by distinguishing between *teaching staff role* and *professor/associate professor role*, and defining them separately in the university as shown in Fig. 4.

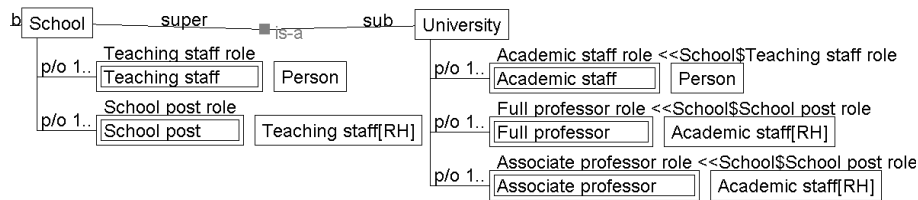


Fig. 4. Class definition to solve section 2-(2)

In the school context, *teaching staff role* and *school post role* are defined. The former is a role concept which means the player belongs to the school as teaching staff. An instance of *person* plays the teaching role and thereby becomes a teaching staff role holder. And the latter means a post which a *teaching staff role holder* holds in the school. In the *university* context, which is a sub concept of *school*, *academic staff role* is defined as a subclass of *teaching staff role*, and *full professor/associate professor roles* are defined as subclasses of *school post role*. In comparison with Figure 3, the super concept of *full professor role* and *associate professor role* in Fig.4 is a *school post role*. And their potential player is a *university staff role holder* rather than *person*. This class definition can solve the instance management problem concerned with promotion discussed in section 3.1 as follows.

Firstly, when Jiro is an associate professor of Osaka University, he plays an instance of an *academic staff role* in Osaka University and thereby becomes an *academic staff role holder* (referred to as RHi). Then, RHi plays an instance of *associate professor role* and thereby becomes the associate professor (role holder). Next, when Jiro is promoted from associate professor to full professor, he (RHi) stops to play the *associate professor role* and plays a *full professor role* while he does not stop to play the *academic staff role*, since he comes back to not an ordinary *person* but RHi which is a role holder of *academic staff role* when he stops to play the *associate professor role*. This problem is thus resolved successfully.

4.3 Constituent Role and Post Role

The instance management problem concerned with post promotion can be generalized to a problem which occurs under following three conditions: 1) there are more than two role concepts in a context which has a common super class, 2) a player stops to play a role instance of one of them, then he/she plays a role instance of another, and 3) the player remains to participate in the same context during the change of roles. The solution to this problem is to distinguish between the two role concepts: a role concept which means its player participates in the context and role concepts which mean roles (posts) that their players occupy perform in the context. We call these two role concepts **constituent role** and **post role**, respectively. Role holders of *constituent roles* are used to denote classes of potential players of *post role*.

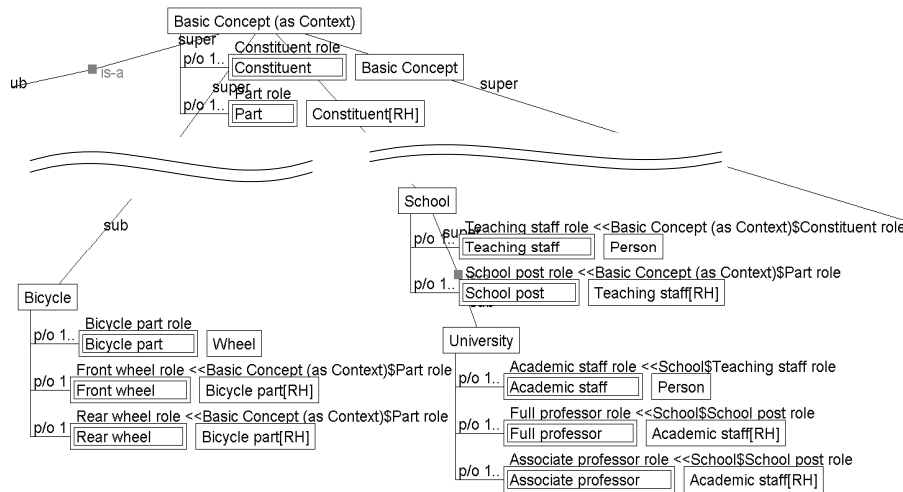


Fig. 5. Class definitions of Constituent role and Part role

In the example discussed section 4.2, *teaching staff role* in a school and *academic staff role* in a university are constituent roles which means their players are participating in the organization (school and university). And, *school post role* in a school, *associate professor role* and *full professor role* are *post roles* which mean roles (posts) are performed in the organization by role holders of the *constituent roles*.

We can find the same problem concerning artifacts. For example, suppose that a front wheel in a bicycle is replaced by a rear wheel in the same bicycle. The replaced wheels change its role while keeping participation in the context. To capture this case adequately, we should define *bicycle part role* as a constituent role and some post roles (e.g. *front wheel role*, *rear wheel role*) played by role holders of the *bicycle part roles*. Figure 5 shows examples of them discussed in this section.

4.4 Context Dependency of Constituent Roles

We can recognize constituent roles according to their contexts. In this section, we discuss contexts on which *constituent role* depends through some examples.

We consider two roles in a company: a *sales member role* in a sales department and *personnel officer role* in a personnel department. Here, we assume a case where Mr. Kimura moves from the sales department to the personnel department in a company. It means that the instance of *person* (Mr. Kimura) stops to play the instance of the *sales member role* then plays the instance of the *personnel officer role*. In this example, because it is natural to regard that he has been an employee of the company, we should define an *employee role* as constituent role whose role holder can be potential players of the *sales member roles* and the *personnel officer roles* (Fig. 6.).

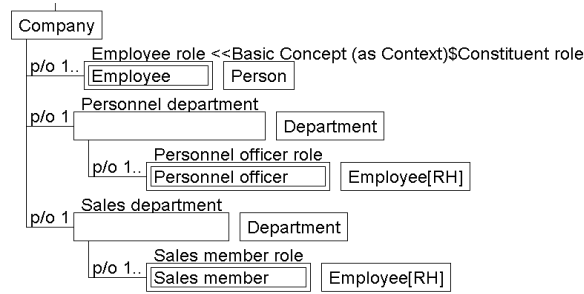


Fig. 6. Constituent roles in company

Furthermore, we assume a case where Mr. Kimura is promoted to the manager of the personnel department. It means that he stops to play the instance of the *personnel officer role* then plays an instance of *manager role* in the personnel department while he remains to be a member of the department. To capture this change in an instance model accurately, *staff member roles* should be defined as constituent role and their role holders should be potential players of *personnel officer roles* and the *manager role* in the personnel department (Fig. 7.).

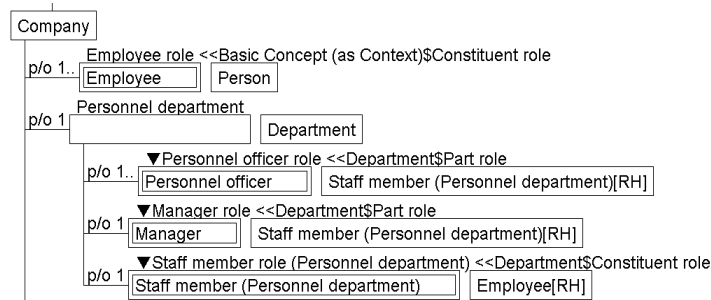


Fig. 7. Constituent roles in personnel department

These examples show that *constituent role* should be defined according to the context in which the player continues to participate in the context when he/she changes roles.

4.5 Ontological Consideration of Constituent Role and Post role

In the previous section, we discussed that *constituent roles* and *post roles* should be distinguished to manage behavior of instances. The discussion was done from engineering point of view. From philosophical point of view, on the other hand, we can extend our theory of *constituent roles* and *post roles* to all type of entities which have unity. For example, we consider a table composed of a top board and four legs. *Component roles* of the table are defined as *constituent role*, and a *top board role* and *leg roles* are defined as *post roles*.

In the real world, just a set of one board and four sticks cannot be regarded as a table when the table is not constructed using them yet. It means they play only the instances of *component roles* (constituent roles) while the instances of the *top board role* and *leg roles* (post roles) are not played. In brief, such a status in which players play only the constitution roles of a context is considered just a set of players of the context. In order to make a thing as an independent whole, its properties and functions must be realized by its components, that is, *post roles* must be played by appropriate *component roles*.

In this example, when the table is constructed using the board and sticks, they play the *post roles* (e.g. the *top board role* and *leg roles*) and thereby its properties and functions as an independent whole are realized. Then, it is regarded as not a just set of components in an instance model but an independent whole which has properties and functions as a table.

On the basis of this consideration, we can understand that a members' club also has *constituent roles* and *post role* even if it has only one kind of membership. If the members of a club have only *constituent roles* without *post roles*, it means a just set of persons. To define members' club properly, *post roles* such as *membership roles* should be defined.

As mentioned above, all type of entities which have unity have *constituent roles* and *post roles* from a philosophical point of view. While we can omit to define constituent roles as an engineering approximation in the cases where instance management problems discussed in section 4.1 do not occur, we cannot omit definitions of *post roles* otherwise.

5 Identity of Roles Instances

When the *Prime Minister of Japan* changes from Abe to Fukuda, we can regard that they play the same role as the *head of the Japanese Government*. In our role model, it means the player of an instance of a *head role* in *Japanese Government* changes from Abe to Fukuda and thereby *Japanese Prime Minister role holder* also changes. In this example, while the instance of the *head role* has kept the same identity, the *Japanese Prime Minister role holder* Abe and the *Japanese Prime Minister role holder* Fukuda have different identities because the identity of role holders are generated using the

identity of the role concept and its player. It can explain the sameness and difference between the two Japanese Prime Ministers.

Next, we consider *Diet member roles* as another example. If the number of Diet seats is 480, we can regard there are 480 instances of *Diet member roles*. Whichever instance of the *Diet member roles* is played by a player, the identity of the *Diet member role holder* is not influenced because they have the same right in the *Diet*. However, when 480 persons play the instances of *Diet member roles*, these instances of roles should be distinguished. This suggests that the identities of *Diet member roles* have weaker identity than *head role* in *Japanese Government*.

Therefore, we are considering definitions of some kinds of identities such as **strong identity** and **weak identity**. The strong identity can identify the uniqueness of an instance, and the weak identity only can argue a fact which an instance is different from others. While these definitions are under consideration, they would contribute to an instance management of roles.

6 Related Work

Guarino and his colleagues aim to establish a formal framework for dealing with roles [2,4,5]. Gangemi and Mika introduced an ontology for representing contexts and states of affairs, called D&S, and its application to roles [7]. Their research was concerned with formalities and axioms of an ontology. In contrast, rather than formalizing role concepts, our goal has been to develop a computer environment for building ontologies. Our notions of role concepts share a lot with their theory of roles, especially context-dependence and specialization of roles. According to their theory, our framework can be reinforced in terms of axioms.

Our notions differ from their work on other two points: the dynamics of a role, and the 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, the framework by Guarino and colleagues deals with time-dependency explicitly. Secondly, we distinguish *role concepts* from *role holder concepts* [8,9]. On the basis of this distinction, we have developed a tool for properties and relations on roles. Masolo et al. introduced a new kind of entity, called qua-individuals, to solve the counting problem [5]. According to them, qua-individuals would be created each time an entity is classified by a role. So if a person plays two roles, the qua-individuals of the person would be created twice, and he/she would be counted three times as a person and the two roles. Qua-individuals seem to be slightly similar to role holder, but it is unclear how to create their instances and identities, while the notion of role holder does not produce such problems that qua-individual would cause.

7 Conclusions and Future Work

In this paper, we have discussed three problems related to instance management of roles. To solve these problems, we have refined our role model based on an ontological

theory of role. The main contribution is the decomposition of role concepts into *constituent roles* and *post roles*. It can explain the continuity and context dependency of instances of roles when the players change their roles. However, the problems and their solutions are never arbitrary. While the idea for solutions looks simple, its generality is high enough. The problem discussed in this paper appears whenever several role concepts are defined in a same context. The proposed theory can be applied to such cases independently of the role player is human being or things, and the unique solution is obtained. As the result, we explained our role model and its implementation in Hozo can solve these instance management problems appropriately. Because constitution roles and part roles which we introduced in this paper can be represented based on the role model using Hozo, we think their solid foundation and formal definitions are certified.

As future work, we plan to further investigate problems related to appearances/disappearances of roles, an investigation of kinds of identity, a version management of instances of role concepts and their players.

References

1. Gruber, T.: A translation approach to portable ontologyspecifications. In: Proc. of JKAW 1992, pp. 89–108 (1992)
2. 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, pp. 527–534 (1998)
3. Masolo, C., Vieu, L., Bottazzi, E., Catenacci, C., Ferrario, R., Gengami, A., Guarino, N.: Social Roles and their Descriptions. In: Proceedings of the 9th International Conference on the Principles of Knowledge Representation and Reasoning (KR 2004), pp. 267–277 (2004)
4. Mizoguchi, R., et al.: A Model of Roles within an Ontology Development Tool: Hozo. J. of Applied Ontology 2(2), 159–179 (2007)
5. Masolo, C., Guizzardi, G., Vieu, L., Bottazzi, E., Ferrario, R.: Relational roles and qua-individuals. In: Boella, G., Odell, J., van der Torre, L., Verhagen, H. (eds.) Proceedings of the 2005 AAAI Fall Symposium Roles, an Interdisciplinary Perspective: Ontologies, Languages, and Multiagent Systems, Arlington, VA, Technical Report FS-05-08, pp. 103–112. AAAI Press, Menlo Park (2005)
6. Kozaki, K., Sunagawa, E., Kitamura, Y., Mizoguchi, R.: Role Representation Model Using OWL and SWRL. In: Proc. of 2nd Workshop on Roles and Relationships in Object Oriented Programming, Multiagent Systems, and Ontologies, Berlin, July 30-31 (2007)
7. Gangemi, A., Mika, P.: Understanding the Semantic Web through Descriptions and Situations. In: Meersman, R., Tari, Z., Schmidt, D.C. (eds.) CoopIS 2003, DOA 2003, and ODBASE 2003, vol. 2888, pp. 689–706. Springer, Heidelberg (2003)
8. Kozaki, K., et al.: Development of an Environment for Building Ontologies which is based on a Fundamental Consideration of “Relationship” and “Role”. In: PKAW 2000, Sydney, Australia, pp. 205–221 (December 2000)
9. Kozaki, K., et al.: An Environment for Building/Using Ontologies Based on a Fundamental Consideration of “Role” and “Relationship”. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473, pp. 213–218. Springer, Heidelberg (2002)