Role Representation Model Using OWL and SWRL

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

Abstract. Role is very important in ontology engineering. Although OWL has been available for ontology representation, consideration about roles is not enough. It can cause to decrease semantic interoperability of ontologies because of conceptual gaps between OWL and developers. To overcome this difficulty, this paper presents some consideration for dealing with roles using OWL.

1. Introduction

Ontology is one of the key technologies for realization of the Semantic Web. To represent web-ontologies, OWL and SWRL have been published as a W3C Recommendation. Although there are many tools for ontology development in OWL and SWRL, few of them provide a higher-level framework for conceptualization of the target world with fundamental discussion. That can cause to decrease semantic interoperability of ontologies because developers need to devise idiosyncratic patterns for building their own ontologies for themselves and such patterns will lack compatibility with others. In this research, we focus on roles [1, 2, 3, 5, 7] as one of the common and typical semantic primitives in ontology development, and investigate representation model for dealing with characteristics of roles in OWL and SWRL justified by fundamental consideration. It contributes to increasing semantic interoperability of roles by providing an infrastructure for role representation.

This paper is organized as follows. Section 2 clarifies roles treated in this paper and summarizes characteristics of roles as requirements for representation model. Section 3 evaluates some examples of role representation. After that, role representation model in this research is presented. Section 4 describes some related work. And, Section 5 concludes this paper with description about some future work.

2. Characteristics of Roles (Requirements)

2.1 Roles in Our Model

In this section, we summarize fundamental schema of our role model proposed in previous work [10]. The fundamental scheme of roles at the instance level is the following (see the lower diagram in Fig. 1): *"In Osaka high school, John plays*

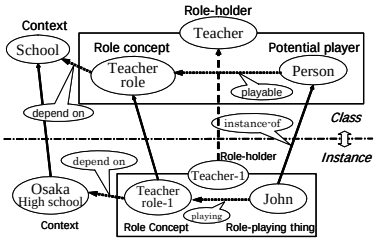


Fig. 1. Fundamental schema of a role concept and role holder

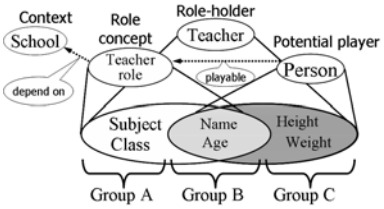


Fig. 2. Conceptual framework of a role, player and role-holder

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**[1] and thereby become **teachers**"*. By play, we mean "act as", that is, it contingently acts as the role (role concept). By "**teacher**", we mean a class of persons who are playing teacher roles.

We introduced a couple of important concepts to enable finer distinction 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**."* By **context**, we mean a class of things to be considered as a whole. It includes entities and relations. R**ole concept** is defined as a class of concepts which are played by something within a context. By **potential player**, we mean a class of things which are able to play an instance of a role concept. In many cases[2], **basic concepts** (natural types) can be used as potential player class. In this example, we say a person can play an instance of *teacher role*. And, when a person (an instance of person class) is actually playing a *teacher role*, he/she thereby becomes an individual *teacher **role-holder***. This means the conventional concept, **player**, is divided into two: One is **potential player** (a role-playable thing) which at the class level, means a class of entities which can play a role of interest and the other is a **role-playing thing**, which is an entity playing the role at the instance level. At the same time, the conventional player link is divided into two kinds: one is playable link (class level) and the other is playing link (instance level). Role holder class is an abstraction of a composition of role-playing thing and an instance of role concept, as is shown in Figs. 1 and 2.

Fig.2. shows the conceptual framework of role model we have proposed. There are two kinds of properties: those of teacher role and person. All of the properties are divided into three groups. Properties of group A are determined by the definition of the role concept itself independently of its player. The second group B is shared by both of the role concept and the potential player. And, the last group C is what the role concept does not care about. Generally, a role concept is defined by describing its properties of group A together with some from group B which are shared by a potential player but come originally from the role concept. Its potential player class is defined by itself context-independently and is used as a constraint of the potential player of the role concept. And, the role-holder is defined as a result of the above two definition operations and eventually includes all of three kinds of properties. Therefore, the individual corresponding to a teacher role holder is the compound of these two instances and totally dependent on them.

---

[1] When we mention a particular role, we put "role" after its name.

[2] In some case, role holders can be used as potential player class. (see section 2.2 (5))

2.2 Requirements for Role Representation Model

In the following items, we summarize characteristics of role concepts. They are referred to as criteria for evaluating role representation models in the next sections.

***(1) Context dependency*:** Context dependence is critical to describe that roles cannot be determined without their contexts and entities change their roles to play according to changes of their contexts.

***(2) Identity of Role concepts*:** Identities of roles is needed to discuss whether two roles are the same or not. For example, when a person is reinstated in his former position, is the role he/she is playing now the same as the one he/she has played? Identities of the roles may answer this question. Furthermore, it enables to represent a vacant post by an individual of a role which is left un-played.

***(3) Distinction between role concepts and role-holders*:** The distinction between role concepts and role-holders is represented the conceptual model discussed in section 2.1. It solves counting problem described in (11).

***(4) Part-whole relation associated with roles and players*:** An object, which is recognized as a whole thing for its part thing(s), can be conceptualized from at least two aspects. From one aspect, the whole thing consists of constituents which build up it (e.g. wheels of bicycle). From the other, it has a conceptual structure which determines roles played by the constituents (e.g. a steering wheel and driving wheel of bicycle). Hence, in some cases, the whole thing is described as a composition of the role-playing thing(s) and a set of the roles. This is similar to a description of a crystal as a composition of "constituents" in the crystal structure and the "crystal structure" without the constituents. By a context for roles in this paper, we mean a conceptual structure. It corresponds roughly to particular patterns of relationships connected with roles by Sowa [9], associations or collaborations in UML.

***(5) Compound role concepts*:** 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 according to interdependency. Such a relation between roles is discussed in other researches as "requirement" [7] and "roles can play roles" [8]. In our terminology, "role-holders can play roles". Such a role concept depends on multiple contexts. For example, teacher can be recognized not only as a staff member of a school but also as a person who teaches students. So, teacher role is interpreted as a compound of school staff role and teaching agent role. So, here, we can identify two kinds of roles according to the complexity of their context dependencies: primitive role concepts and compound role concepts. The former has a single context-dependency and latter has multiple context dependency. A school staff role in school context and a teaching agent role in teaching action context are primitive role. A teacher role is a compound role of them. In order to deal with compound role concepts, here we introduce **Role Aggregation**, which is based on decomposition of the compound role and determination of essential context for it [10].

***(6) An individual plays multiple roles*:** An individual can play multiple roles at the same time. For example, an instance of *Person* may play a *Teacher Role* and a *Husband Role* at the same time.

***(7) Individuals of role concepts*:** Individuals of role concepts have the following two characteristics. (a) They cannot exist if individuals of their contexts do not exist

because roles are externally founded [3,7]. (b) Because roles are dynamic [7], they have two states: played and un-played. (c) They have their own identities independently of their states and their players and are regarded as defective instances until played by some individuals.

***(8) Individuals of role-holders*:** An individual of a role-holder is composed of individuals of a role concept and its player. 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, $ID_{Role\text{-}Holder}=f$ (ID_{Role}, ID_{Player}) in which both arguments are mandatory for $ID_{RoleHolder}$, in which "*f*" is bijective (surjective and injective)

***(9) Disappearance of individuals of role-holder*s:** In connection with (8) and (9), individuals of role-holders disappear when (a) its player disappears, (b) its role disappears and its player quits playing the role.

***(10) Solution of counting problem*:** For example, the number of passengers taking a certain means of transportation in one week may be greater than the number of individual persons traveling with that means during the same period [4,11]. This problem, so-called Counting Problem, can be solved by separately counting identities of individuals (players) and that of Passenger role-holders which are recognized every time that the players play Passenger role. For example, when we need to count the number of passengers, we use the $ID_{Role\ holder}$, and when we need to count the number of persons, we use ID_{Player} instead of $ID_{Role\text{-}Holder}$.

***(11) Players of compound role concepts*:** Individuals of roles as constituents of a compound role need to be played together by the same individual.

3. Representation of Roles Using OWL and SWRL

3.1. Examples of Role Representation

Fig.3 shows three examples of role representation model in OWL. In this section, we evaluate each of them according to the requirements for dealing with characteristics of roles discussed in Section 2.2. Here, we refer a Teacher Role, which depends on a School as its context, exampled in Section 2.1. Table 1 shows the result of comparison among the examples. (d) and (d)+ in the table shows representation ability of the proposed models explained in the next section.

Example 1 (in Fig.3-a): A role as a Teacher is dealt with in *teacherOf* property. This property may represent the role which is determined in e.g. "teacher-student relation". In this model, however, context dependency of role concepts is implicit (see (1) in the former section). That causes a critical problem because the context dependency relates essentially to other characteristics.

Example 2 (in Fig.3-b): This model represents a context of Teacher explicitly. However, the role is dealt with still in a property. That can complicate management of identity of roles in its instance model (see (2)). For example, it is difficult to describe that, after some person quits his/her job as a Teacher, other person fills the same post as the Teacher. Moreover, this model can not represent state of the role concept: played or un-played (see (7)-b). It means, for example, a vacant post can hardly be represented and identified.

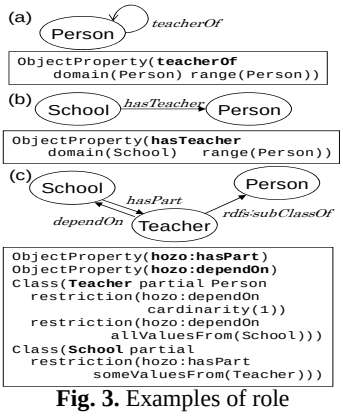


Fig. 3. Examples of role representation models

Table 1. Comparison of role representation models

Characteristics of Role Concepts and Role-Holders	*(a)*	*(b)*	*(c)*	*(d)*	*(d)+*
(1) Context dependency	-	OK	OK	OK	OK
(2) Identity of Role concept	-	-	OK	OK	OK
(3) Distinction between role concepts and role-holders	-	-	-	-/OK	OK
(4) Part-whole relation associated with roles and players	-	-/OK	-/OK	-/OK	OK
(5) Compound role concepts	-/OK	-/OK	-/OK	OK	OK
(6) An individual plays multiple roles	OK	OK	OK	OK	OK
(7) Individuals of role concepts	-	-	-	-/OK	OK
(8) Individuals of role-holders	-	-	OK	-/OK	OK
(9) Disappearance of individuals of role-holders	OK	OK	-/OK	-/OK	OK
(10) Solution of counting problem	-	-	-	OK	OK
(11) Players of compound role concepts	-/OK	-/OK	-/OK	-	OK

OK:represented, -:not represented, -/OK:represented partially

Example 3 (in Fig.3-c): The *hasPart* property in this model means School consists of Teacher(s) here. And a restriction on *dependOn* property in Teacher class expresses that a Teacher depends on School as its context. This model is superior to the above two models because their problems can be solved in this model. However, a Teacher is classified into a Person in confusion between role concepts, role-holders and basic concepts (see (3)). Hence, according to the semantics of *rdfs:subClassOf*, an instance of a Teacher and its player (an instance of Person) are required to be one and the same instance. That causes the player cannot stop to be an instance of a Teacher without stopping to be an instance of a Person, i.e., deletion of an instance of a Teacher brings with deletion of an instance of a Person (see (9)). Furthermore, in this model, the Counting Problem cannot be solved because it is necessary for solution of the problem to distinct role-holders from role concepts (see (3) and (10)).

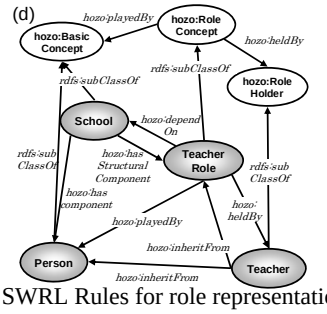
3.2. Role Representation Model

In Fig. 4(d), we represent our role model in OWL for dealing with characteristics of roles with fundamental consideration as faithfully as possible. We define some properties and classes which are indicated by namespace “hozo:”. For example, hozo:BasicConcept class, hozo:RoleConcept class and hozo:RoleHolder class express basic concepts, role concepts and role-holders respectively. hozo:playedBy property represents a relation between classes of role concept and classes of potential player. This property indicates role-playable thing discussed in 2.1. When a relation between an instance of role concept and player is represented as hozo:playedBy property, the property means a playing relation between them. hozo:RoleHolder class represents a role holder.

Fig. 4(d)+ gives rules which are applied into classes and properties in this role representation. Theey are written in a human-readable style and can be implemented in SWRL rules like an example under the table. They are not only applied to instance models for inference, but also implying our policies on using the classes and properties in this section for describing characteristics of roles. For example, Rule-03 and 04 require that we describe a role concept with two properties (*hozo:dependOn* and *hozo:hasStructuralComponent*) among the role and a class as its context.

In the following items, we evaluate this model also with reference to characteristics of roles summarized above. The evaluations are shown in Table 1 as (d) and (d+). In

(d)



(d)+ : SWRL Rules for role representation model

	Antecedent	Consequent
Rule-01	hozo:playedBy(?x, ?y) ∧ hozo:hasStructuralComponent(?z, ?x)	hozo:hasComponent(?z, ?y)
Rule-02	hozo:playedBy(?x, ?y) ∧ hozo:hasComponent(?z, ?y)	hozo:hasComponent(?z, ?x)
Rule-03	hozo:structuralComponent(?x, ?y)	hozo:dependOn (?y, ?x)
Rule-04	hozo:dependOn (?x, ?y)	hozo:structuralComponent(?y, ?x)
Rule-05	hozo:playedBy(?x, ?y)	hozo:heldBy(?x, ?z)
Rule-06	hozo:heldBy(?x, ?y)	hozo:playedBy(?y, ?x)
Rule-07	hozo:playedBy(?x, ?y) ∧ hozo:heldBy(?x, ?z)	hozo:inheritFrom(?z, ?x)
Rule-08	hozo:inheritFrom(?x, ?y) ∧ hozo:RoleConcept(?y)	hozo:heldby(?y, ?x)
Rule-09	hozo:aggregateOf(?x, ?y) ∧ hozo:playedBy(?x, ?z)	hozo:playedBy(?y, ?z)
Rule-10	hozo:aggregateOf(?x, ?y) ∧ hozo:playedBy(?y, ?z)	hozo:playedBy(?x,?z)
Rule-11	aggregateOf(?x, ?y)	differentFrom(?x, ?y)

```
Example : Rule-01
    Implies( Antecedent( hozo:playedBy(I-variable(x) I-variable(y))
                         hozo:hasStructuralComponent(I-variable(z) I-variable(x)))
             Consequent( hozo:hasComponent(I-variable(z)) I-variable(y)))
```

```
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 cardinality(1))
    restriction(hozo:playedBy maxCardinality(1)))
     restriction(hozo:heldBy maxCardinality(1)))
Class(hozo:RoleHolder partial
    restriction(hozo:inheritFrom cardinality(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. 4. Role represntation model in Hozo

column (d), the model is evaluated only within the description of this model in OWL. And, in column (d+), it is done within the descriptions in OWL and SWRL.

***(1)Context dependency*:** The definition of hozo:RoleConcept has a restriction on this property to have exactly one *hozo:dependOn* property. It represents all role concepts depend on another class as their context.

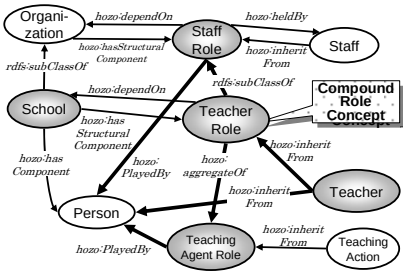
***(2) Identity of Role concepts*:** Roles are conceptualized as classes and are categorized into hozo:RoleConcept as its subclasses. One of the major contributions by treating roles as not properties but classes in the syntax of OWL is to make management of identities of roles easier.

(3) Distinction between role concepts and role-holders: In the same way as roles, role-holders are categorized into hozo:RoleHolder. And, they are discriminated from each other explicitly. Role-Holders are described with *hozo:inheritFrom*. This property is used for representing that a role-holder inherits definitions from role concept or its player. But the property does not imply inheritance of identity, and in that respect *hozo:inheritFrom* differs from *rdfs:subClassOf*.

***(4) Part-whole relation associated with roles and players*:** Part-whole relation is represented by two properties (*hozo:hasStructuralComponent* and *hozo:hasComponent*) and rules for reasoning on them (Rule-01,02 in Table 2).

***(5) Compound role concepts*:** Fig. 5 shows an extended model of the one in Fig. 4 for role aggregation. In this example, a Teacher Role is a compound role concept defined by aggregation of a Staff Role and a Teaching Agent Role. A compound role concept is described by role aggregation using two properties: *rdfs:subClassOf* and *hozo:aggregateOf*. The latter means that, a role concept in its range inherits some properties from one in its domain.

***(6) An individual plays multiple roles*:** This can be represented by several instances of role concepts are connected with one and the same individual through *hozo:playedBy*.




```
ObjectProperty(hozo:aggregateOf
    domain(hozo:RoleConcept)
    range(hozo:RoleConcept))
Class(Teaching Agent Role partial hozo:RoleConcept
    restriction(hozo:playedBy
                allValuesFrom(Person)))
Class(Teacher Role partial Staff Role
    restriction(hozo:dependOn
                allValuesFrom(School))
    restriction(hozo:playedBy
                allValuesFrom(Person))
    restriction(hozo:aggregateOf
                allValuesFrom(
                     Teaching Agent Role)))
```

Fig. 5. Representation model for Role Aggregation in Hozo

***(7) Individuals of role concepts*:** (a) By a restriction on cardinalities of *hozo:dependOn* in hozo:RoleConcept and Rule-03 and 04 in Table 2, it is described that an individual of a role concept exists always accompany with an instance of its context. (b) Two states of a role: played or un-played are distinguished by whether the role has *hozo:playedBy* or not. (c) Individuals of role concepts are identified as instances of hozo:RoleConcept.

***(8) Individuals of role-holders*:** An instances of a role holder exists iff *hozo:playedBy* holds between an instance of its role concept and one of its player. And, the instance of role-holder inherits the definitions from instances of the role concept and the player by *hozo:inheritFrom* (Rule-05~09 in Table 2).

***(9) Disappearance of individuals of role-holders*:** Restrictions on properties of hozo:RoleHolder and Rule-05~09 described in (8) mean also that iff their onditions are not fulfilled, an individuals of role-holder cannot exist.

***(10) Solution of counting problem*:** An instance of a player and one of a role-holder are distinguished by their own identities. Counting Problem can be solved by counting them separately.

***(11) Players of compound role concepts*:** Rule-10~12 in Table 2 describe that a player of a compound role must also play other roles connected with the compound role by *hozo:aggregateOf* simultaneously.

4. Related Work

W3C has started up Semantic Web Best Practice and Deployment Working Group for providing typical semantic primitives of ontologies as Ontology Engineering Patterns. For example, in the draft on simple part-whole relations[3], it is described that "It is important to realize that making, e.g. Engine a subclass of e.g. CarPart means that all engines are car parts - which is simply not true". These problems show exactly why we need to discriminate roles from the others for development of ontology. In the role representation model presented in this paper, CarPart and Engine are regarded respectively as a role concept and as a player of the role. In this way, the role representation model contributes to assure semantic interoperability of roles.

Guarino and his colleagues aim to establish a formal framework for dealing with roles [3, 7]. And Gangemi and Mika introduce an ontology for representing a context and states of affairs, called D&S, and its application to roles [2]. Our notions of role

[3] http://www.w3.org/2001/sw/BestPractices/OEP/SimplePartWhole/

concepts share a lot with their theory of roles; that is, context-dependence, specialization of roles, and so on. But our role model differs from their work on other two points. Firstly, we focus on context-dependence of a role concept. 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. Secondly, we distinguish role concepts and role holders.

5. Conclusion and Future Work

In this paper, we have discussed characteristics of roles and developed a role representation model using OWL and SWRL. The model covers major important aspects of roles and is available to represent role and their characteristics. It not only contributes to semantic interoperability of ontologies but also affords clues for solving problems caused by confusion of roles, the counting problem and so on.

As future work, we plan to investigate some other characteristics of roles such as instance management of roles (when it is created or deleted?), categories of roles, and so on. The instance management is the most serious among them. In order to clearly understand playable, playing, depend-on relations, we need to investigate when and how the related instances appear and disappear in what interdependence.

References

1. Fan, J., Barker, K.,et al.: Representing Roles and Purpose. In Proc. of the International Conference on Knowledge Capture (K-Cap2001), pp.38–43, Victoria, B.C., Canada (2001)
2. Gangemi, A., Mika, P.: Understanding the Semantic Web through Descriptions and Situations, ODBASE 2003, Catania, Italy (2003)
3. Guarino, N.: Concepts, attributes and arbitrary relations, Data & Knowledge Engineering 8 (1992) 249-2615.
4. Guizzardi, G. “Agent Roles, Qua Individuals and The Counting Problem”, in Software Engineering of Multi-Agent Systems, vol. IV, Springer-Verlag, 2006.
5. Guizzardi, G., “Ontological Foundations for Structural Conceptual Models”, Telematica Instituut Fundamental Research Series, Universal Press, The Netherlands, 2005
6. 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, USA (2005)
7. Masolo, C., Vieu, L., Bottazzi, E., et al.: Social Roles and their Descriptions. In Proc. of the 9th International Conference on the Principles of Knowledge Representation and Reasoning (KR2004), pp.267–277, Whistler, Canada (2004)
8. Steimann, F.: On the Representation of Roles in Object-oriented and Conceptual Modeling, Data & Knowledge Engineering 35 (2000) 83-106
9. Sowa, J. F. 1988. Using a lexicon of canonical graphs in a semantic interpreter, in Relational models of the lexicon (1988)
10. 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, pp. 136-143, USA (2005)
11. Wieringa, R. J., Jonge W. de, Spruit P. A.: Using Dynamic Classes and Role Classes to Model Object Migration. Theory and Practice of Object Systems 1 (1995) 61-83