Organizing *Role-concepts* in Ontology Development Environment: Hozo

Eiichi Sunagawa, Kouji Kozaki,
Yoshinobu Kitamura and Riichiro Mizoguchi

AI-TR-04-1
Received 26 Aug. 2004

AI Technical Report

Artificial Intelligence Research Group
The Institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka, 567-0047 Japan
Tel: +81-6-6879-8416, Fax: +81-6-6879-2123
{sunagawa, kozaki, kita, miz} @ei.sanken.osaka-u.ac.jp
http://www.ei.sanken.osaka-u.ac.jp/

Abstract:

In this research, we aim to enhance descriptive quality of *role-concept* in our ontology development system Hozo. Based on some fundamental theories of ontology, we can define a "concept" of a role which an object plays in a context, although that is treated as one kind of "property" in most of the ontology development systems. When constructing an ontology, discrimination of role-concepts from the other concepts helps to develop a theoretically sound ontology which describes definitions of concepts properly based on ontological theories. So, we have been developing a framework for definition of role-concepts in Hozo. In this paper, we extend the framework for the specification of a hierarchy of role-concepts such as teacher, husband, commodity, etc. differentiated from the other concepts such as human, apple, etc., and we present how to treat the relations between these two kinds of hierarchies of concepts.

1. Introduction

The more Semantic Web attracts attention, the more importance of ontology increases. Software agents and people can find information about many domains on the web. And, on each domain or several domains, ontology can describe what concepts exist in the target world. Many tools for ontology development have been developed to date. On the other hand, fundamental theories of ontology based on philosophy have been investigated exhaustively. However, not many of the ontology development tools provide higher level of constructs for ontology description compliant with the fundamental theories of ontology. So, it is still difficult to develop a theoretically sound ontology which describes definitions of concepts properly based on ontological theories. We have developed an environment for building/using ontologies based on a fundamental consideration of *role* and *relationship* [1, 2]. In this research, we aim to enhance descriptive quality of *role-concept* in our ontology development system Hozo.

It is indispensable for building ontology to describe properties (or attributes) of concepts. Ontology languages for the Semantic Web provide some mechanisms to define *property* for representing roles, relations, attributes, etc., and most of ontology development tools support them. While those properties have different semantics according to the theories of ontology, most of the tools do not support users to develop an ontology which reflects distinctions among them. Even if the users contrive to describe distinction among the kinds of properties in some way, the descriptions tend to be ad hoc and not be able to define concepts rightly or naturally in the same way as people understand them. To make it easier to develop an ontologically-sound ontology, we need a tool with higher level constructs which help us to differentiate properties and provide a framework for representing such differentiation.

In this paper, we focus on *role-concepts* and discuss a way to organize and define them consistently. Through those considerations, we intend to design a framework for enhancement of descriptive quality of role-concepts in Hozo. Based on some fundamental theories of ontology, we can recognize a kind of concept; so called "role-concept", in this paper, which is defined as a concept of a role which an object plays in a context. When constructing an ontology, discrimination of role-concepts from the other concepts helps to develop a theoretically sound ontology which describes definitions of concepts properly based on ontological theories. The definition of role-concepts and how to treat them have been investigated exhaustively in different research areas: knowledge representation, description logics, knowledge engineering, object-oriented and conceptual modeling, multi-agent systems, sociology and philosophy and so on. Also in a research area of ontology engineering, the role-concept is a significant topic. Masolo et al. discuss a social role after survey of the literature in those areas [3] and Guarino brings up some problems related role-concepts [4]. A Role-concept is closely related with ontology primitive.
Therefore, it necessary for the Semantic Web on which various ontologies are developed to investigate a general framework consistent in the treatment of role-concepts. In this background, we have been developing a framework in Hozo for ontology development which can describe role-concepts. And, we extend it for designing a specification of a hierarchy of role-concepts such as *teacher, husband, commodity, etc*. differentiated from the other concepts such as *human, apple, etc.*, and we present how to treat the relations between these two kinds of hierarchies of concepts. Here, we should emphasize that this paper never imposes a methodology for universal organization of role-concepts. We just aim to develop and propose an effective tool for organizing role-concepts. So, we see that to inquire essential identity of a role-concept and to design an ontology specifically are outside the scope of our research goal.
Section 2 explains the definition of a role-concept and its related concepts in Hozo. Section 3 introduces our ontology building/using environment Hozo and describes its previous framework for treatment of a role-concept. Section 4 describes how to extend our framework for a specification of a hierarchy of role-concepts and how to realize *aggregation of role-concepts*. They are distinctive characteristic of role-concept. Section 5 discusses hot to treat role-concepts in instance models. Section 6 shows an example of the translation of role-concepts in Hozo into OWL. Section 7 focuses on related work and section 8 concludes this paper.

2 Role-concepts in Hozo

2.1 Key concepts

With citing work by Charles S. Peirce, John Sowa introduced the *firstness*, the *secondness* and the *thirdness* of concepts [5]. The firstness can be roughly defined as a concept which can be defined without mentioning other concepts. Examples include iron, a man, a tree, etc. In a similar, the secondness can be defined as a concept which cannot be defined without mentioning other concepts. Examples include wife, teacher, child, etc. We call one kind of the *secondness* type except artifacts[1] ***role-concepts*** in this paper. The thirdness links the firstness and the secondness. Examples include paternity, brotherhood, etc.

Based on these theories, we identified three categories for a concept. They are a *role-concept*, a *basic-concept* and a *role-holder*. A ***role-concept*** represents a role which an object plays in a specific context and it is defined with other concepts. On the other hand, a ***basic-concept*** does not need other concepts for being defined. An entity of the basic concept that plays a role-concept is called a ***role-holder***.

A role-concept in Hozo is defined with reference to the following three conceptual elements.

- ***Role-holder:*** An entity of a basic-concept that plays the role
- ***Class-constraint:*** constraint on a class which an instance playing the role is classified into
- ***Context:*** A concept which the role is recognized through a relation with

The class-constraint usually[2] refers to the *basic-concept* which is defined elsewhere. When an instance that satisfies the class-constraint plays the role, it becomes the role-holder. For example in "a school" context, some people play the role as a teacher ("a teacher-role"), and others play "a student-role". People who play "a teacher-role" and "a student-role" are called respectively "teachers" and "students" which are *role holders*.

Our aim here is to differentiate role-concepts from basic-concepts clearly in their semantics and notations. It is advisable for a computer model and an ontology behind it to correspond to the real world as truly as possible. Then, assume that <human *is-a* animal> and <teacher *is-a* person> has been describes in some ontology. If an instance of "human" class is deleted, that also means deletion of an instance of "animal" class according to the definition of *is-a* relation. This operation in a computer model corresponds to an event that someone dies and changes from an animal to a body in the real world, because that instance doesn't hold an identity as an animal. The second description, however, doesn't mean similarly. Because, in the real world, an event that some teacher quits his/her job doesn't means his/her death. Therefore, deletion of an instance of "teacher" doesn't much deletion of an instance of "person". At this point, these two *is-a* relations have different semantics. The former is defined more strictly based on the fundamental theory of ontology than the latter, and is used in our system "Hozo". We intend to provide a framework to cope with such a difference clearly. Therefore, we discriminate them and do not adulterate a hierarchy of role-concepts with basic-concepts.

We never claim that, if <teacher *is-a* person> is described in some ontology, a computer model based on it conflicts with the real world. That description may have no trouble with in a model which represents confined context such personnel affairs of school. However, it will be hard to reuse for another system and to integrate with another ontology. Additionally, we do not proscribe treatment of a concept of "teacher" as a subclass of "person". In some model, that treatment may simplify and ease its process. We suggest that consideration of role-concepts in ontology development make consistent policy about that difference of semantics of *is-a* relations. And it contributes to ensuring reusability and versatility of an ontology.

[1] For example, we can accept "to use a box as a desk" without difficulty. However, it just describes that some box can perform such a function of a desk should be discriminated from the conceptualization of a desk itself. Although a desk is one of the *secondness* types of concepts in the sense that it can not exist if no person recognize so in some context, it is designed and produced as an object which plays a desk-role as its intrinsic property. And this role is exactly an identity of a desk. Hence, an artifact itself is not a role-concept differently from a role-concept as an object which occasionally carries out its function.

[2] Ideally, the class constraint refers to basic concept. Practically, however, we can allow users to refer to role-holder when they represent role concept as is discussed in sections 4 and 5.

2.2 Roles treated in the *part-of* relation and the *participant-in* relation

Hozo represents role-concepts in the two kinds of a relation with a context [1, 2]. They are the *part-of* relation and the *participant-in* relation. The former imposes a role as the part which is required to play it within the *whole-concept*. For example, <teacher *part-of* school> represents that "a teacher (role-holder)" composes and belongs to "a school (context)" which is an educational institution. It is defined in the description of the *part-of* relation that "a person (class-constraint)" can play "a teacher-role" and then become "a teacher". The latter imposes a role as the participant which to play it within the *relation-concept*. For example, <brother *participant-in* brotherhood> represents that "a brother participates in a brotherhood". It is defined in the description of the *participant-in* relation that "a man" can play "a brother-role" and then become "a brother". Note that, by a relation-concept, we mean not a concept which can be represented in a relational form but a concept which is recognized with conceptualization of a relation (e.g. enmity, causality, marital relation).

These two kinds of relation are associated with each other. We have defined "a brother" as a role-holder which participates in "a brotherhood" relation. However, we can also define it as a role-holder which compose "a brother" and describe <brother *part-of* brothers>. In most cases, participants in a *relation-concept* can be also seen parts of *whole-concept* which is recognized with collective conceptualization of the all participants. Hozo supports to treat this association.

2.3 Roles in other research areas

Role-concept and its representation have been discussed in different fields. In knowledge representation and description logics, a role-concept is regarded as a binary relation, which some concept acts upon others. Sowa defined roles more strictly and proposed that roles are concepts connected to "patterns of relationships" [6]. On the other hand, Guarino defines roles as properties, which are anti-rigid [7]. In knowledge engineering, CommonKADS introduces knowledge roles such as a hypothesis [8]. They are represented based on notion of a task ontology [9]. In object-oriented and conceptual modeling, especially UML, roles describes how objects behave or how an association will behave in a particular situation in collaboration, which can explain the static structure and the dynamic behavior of a system [10]. In multi–agent system, roles are defined according to action of agents and interaction among agents [11, 12]. Such notation is related with generic functionality of agents in a society of agents and deontic characterization such as permissions and obligations. In sociology and philosophy, discussion focuses on roles of human actors [13, 14]. Tuomera defined roles in terms of sets of tasks and rights, and described an agent plays a role when it uses rights in order to achieve tasks [14].In linguistics, roles in a sentence are recognized as *thematic roles* such as "agent" and "instrument" from the both viewpoints of syntax and semantics [15]. Moreover, *relational nouns* such as "mother" have been investigated (e.g., in [16]). This is similar to the relational treatment in knowledge representation. In cognitive semantics, roles are related to mental representation. Fauconnier discussed roles with reference to a mental space [17]. His theory treats roles and their players. Furthermore, his theory notes also roles in linguistics (e.g. counterfactuals, metaphors).

3 Hozo and role-concepts in a hierarchy of basic concepts

3.1 Hozo and an ontology editor in it

This section reviews our previous framework. Our research goal is to improve on it for enrichment of representational power so that users can treat role-concepts according to our policy of organizing them.

We have developed "Hozo" [1, 2], which is an environment for building ontologies based on fundamental ontological theories. Hozo is composed of "Ontology Editor", "Onto-Studio", "Ontology Server" and "Ontology Manager" (in Fig.1). Ontology Editor provides users with a graphical interface, through which they can browse and modify ontologies. This system manages properties between concepts in the *is-a* hierarchy. The users can define role-concepts also. This topic is explained in next section. Onto-Studio is based on a method of building ontologies, named AFM (Activity-First Method) [18], and it helps users design an ontology from technical documents. Ontology Server manages the built

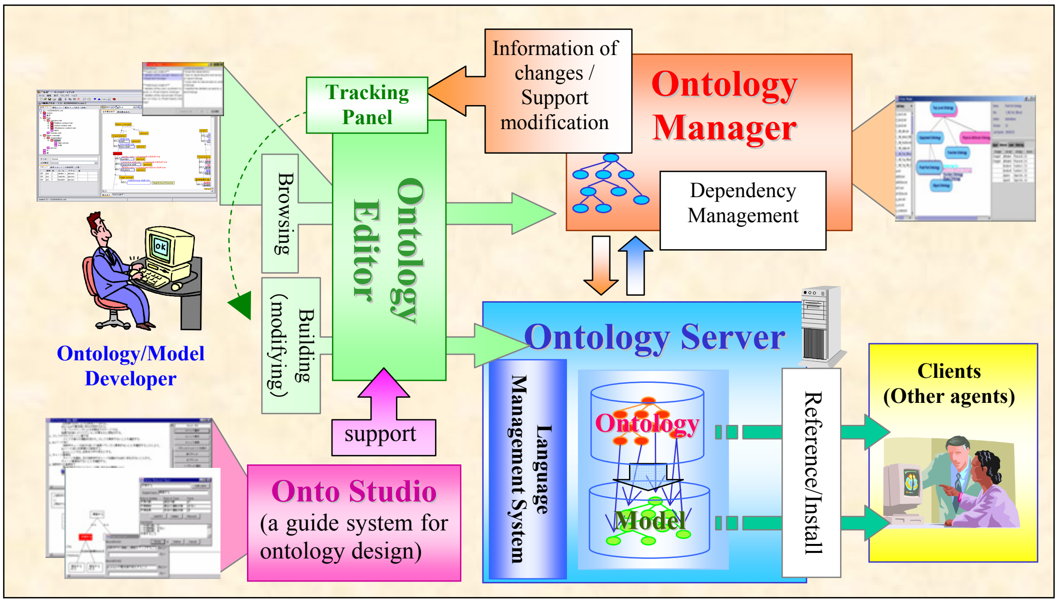


Fig. 1. Hozo: an Environment for Building/Using Ontologies

ontologies and models. Ontology Manager [19, 20] helps users for distributed ontology development. It manages ontologies as components of a target ontology based on their dependencies. And when a change of some ontology influences others, it supports modification of influenced ontology for keeping its consistency.

Because the architecture is implemented in Java and the Ontology Editor is an applet, it can work as a client through Internet. Hozo manages ontologies and models considering who its developer is. Models are built by choosing and instantiating concepts in the ontology and by connecting the instances. Hozo also checks the consistency of the model using the axioms defined in the ontology. The ontology and the resulting model are available in different formats (Lisp, Text, and XML/DTD) that make it portable and reusable.

3.2 Role-concepts in a hierarchy of basic-concepts

An ontology editor in Hozo provides users with a graphical interface, through which they can browse and modify ontologies by simple mouse operations. Its interface mainly consists of the following 3 panels (Fig. 2): *navigation panel, definition panel and browsing panel*. The last two panels are located centrally in this tool. They display the definition of the concept that is selected in the *is-a* hierarchy browser, and allows users to edit it. It is composed of a *browsing panel* and a *definition panel*.

The browsing panel shows an ontology and users edit it mainly on this panel. It has two display modes. One displays it graphically by using nodes and links. Each node represents a whole-concept or a relation-concept which has slots representing its part-concepts or participant-concepts by such a manner that is shown in Fig.3 (a). A diagram of a part-concept is composed of three parts. Each of them represents 1) a *role-concept*, 2) a *class constraint*, and 3) a *role holder*. A symbol besides a link connecting a whole-concept and a part-concept denotes kinds of relation ("p/o" denotes *part-of* relation, "p/i" denotes *participant-in* relation and "a/o" denotes *attribute-of*) and a numeral represents the number of part-concepts (or attributes). Fig.3 (b) shows a person, who is referred in the *class constraint*, plays "a teacher role", and the wheel becomes a *role holder* "a teacher". The other mode displays an ontology under editing in RDF(s).

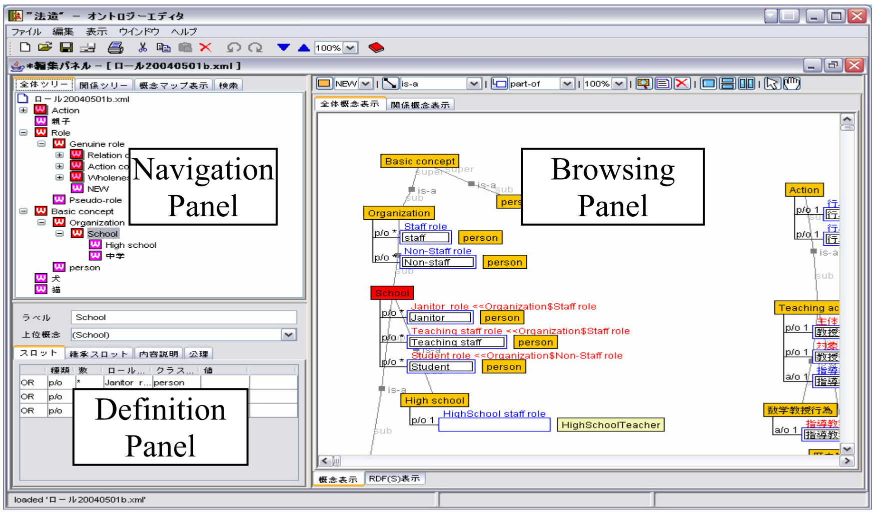
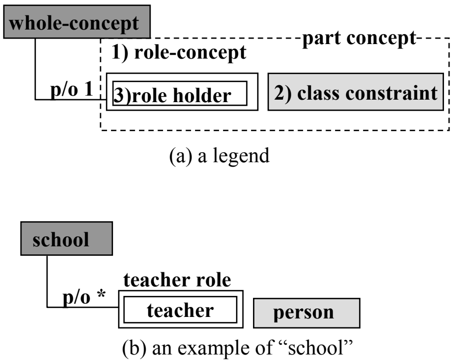


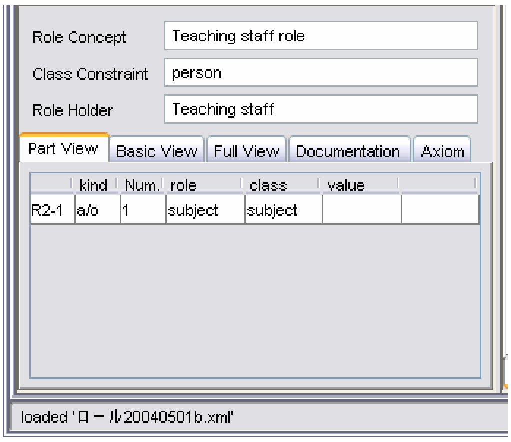
Fig. 2. A snapshot of Ontology Editor



(a) a legend

(b) an example of "school"

Fig. 3. A legend of part-of relation


Role Concept: Teaching staff role

Class Constraint: person

Role Holder: Teaching staff

	kind	Num.	role	class	value	
R2-1	a/o	1	subject	subject		

loaded 'ロール20040501b.xml'

Fig. 4. The definition panel for part-concepts

Role-concepts shown in the browsing panel are defined in detail in the definition panel. This panel is based on our theory of a relation among a role-concept, a basic-concept and a role-holder. That displays definition of part-concepts or participant-concepts, shown in Fig.4. At the top of the panel, the label of a role-concept, a class-constraint and a role-holder are shown. At the bottom of this panel, definitions of these three concepts are shown on a tabbed panel. Users can switch the following three views to read and edit the definition.

- **Part view:** The panel displays definition of the role-concept. It allows users to *add a new definition* and *constraints on properties inherited from a basic concept*.
- **Basic view:** The panel displays definition of a *basic concept* referred to in the class constraint. It allows users only to *select inheritable properties to the role-concept*.
- **Full view:** The panel displays the definition of the *role holder*. It allows users *only to read* the definitions.

Furthermore, users can document the concept or the slot in natural language and describe axioms.

4 Organizing role-concepts according to their context dependence

We aim to enhance descriptive quality of role-concepts in Hozo. In our previous framework described in section 3, role-concepts are dealt with in a basic concept-centered view. So, users cannot easily represent/grasp relations among role-concepts because their definitions are scattered around in the respective related "whole" concepts which give the context of the roles. In this paper, we present a framework for organizing role-concepts in a hierarchy not from the view point of basic concepts but role-centered view. In the following sections, first, we explain some kinds of role-concept from the view point of its context dependency. And next, we illustrate specification of a hierarchy of role-concepts with an example. Finally, we describe an aggregation of role-concepts.

4.1 Categories of role-concepts according to their dependence

For building an ontology, it is important to discriminate among role-concepts. As described in section 3, role-concepts are represented in the form of part or participant. However, when concepts act as various contexts, they contain more semantics. They are regulated in their context dependency. Therefore, they can be organized according to categories of context and formalities of the dependencies. Although we do not cover them in detail yet, we consider that there are, at least, the following kinds of a role-concept:

- Task context role: fault role, symptom role, fault hypothesis role [9]
- Domain context role: Tower-top component role (naphtha)

  They are role-concepts which recognized through a relation between task concepts and domain concepts. A task context role is generally applicable in a problem solving system, and defined according to their total tasks, their activities and so on. A domain context role describes the target world and independent of the purpose of use at some level.
- Function context role: steering wheel role, draw pump role, brake lump role

  It is a role as an object which carries out some function. By "function", we mean result of teleological interpretation of behavior under a goal. In principle, an artifact is designed and produced as an object playing this role. Consideration relevant to this kind of a role-concept is described in foot note 1.
- Action context role: weapon role, learner role, walker role

  It is a role recognized related to the specification of an action such as doer, tool and so on.

Furthermore, we are discussing other kinds of role-concepts. Some of them are role-concepts which do not strictly depend on contexts. For example;

- Capacity context role: club member role, king role, expert role

  It is a role whose career has some qualification.
- State context role: newcomer role, veteran role, rolling agent role

  It is a role from the view point of state.

In addition, kinds of role-concepts are organized differently according to the use model. Furthermore, some role-concept is conceptualized in several viewpoints. For example, "husband" can be recognized as both of a part of "marital couple" and a participant in "marital relationship". So, it is left as our future work to show the guideline of categorizing role-concepts.

4.2 Role-concepts in their hierarchy

In the previous framework provided in Hozo, role-concepts are described only in a hierarchy of basic-concepts. However, users cannot completely organize role-concepts in the framework because of its basic-concept centered view. So, in this research, we provide another framework which can describe a hierarchy of role-concepts. Its mechanism helps ontology developers to associate a role-concept which is used in the description of basic-concepts with a hierarchy of role-concepts that shows in what context they are essentially recognized. Therefore, the framework contributes to ease to deal with a role-concept.

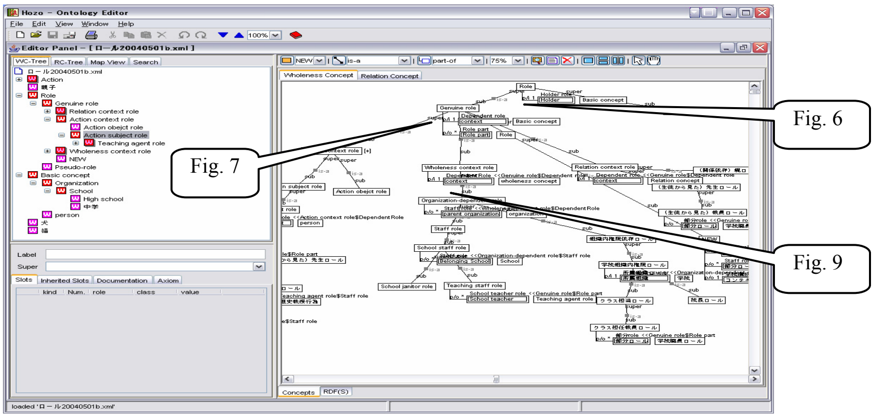


Fig. 5. A total picture of a hierarchy of role-concept constructed with "Hozo"

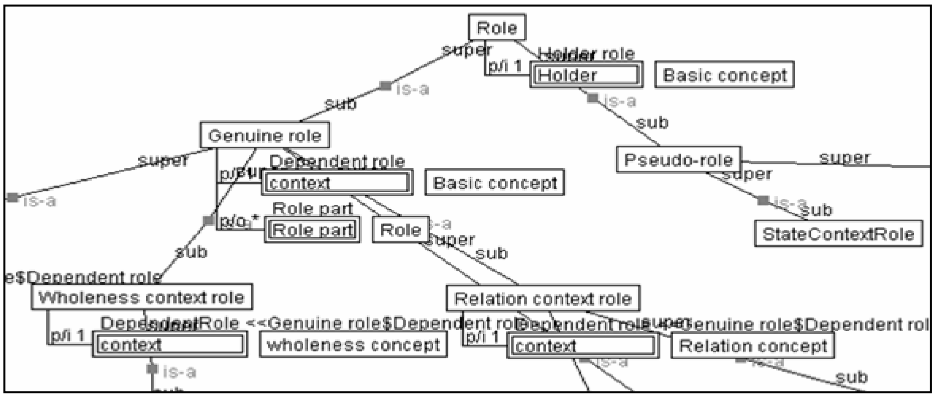


Fig. 6. A definition of "role" in the hierarchy of role-concepts

Because role-concepts are dealt with in the same way as basic concepts in the construction of an ontology, we only have to improve and reuse our previous framework in order to construct a hierarchy of role-concepts.

In this section, we just show what we aim through an example of a hierarchy of role-concepts in Hozo (in Fig.5). It organizes some parts of role-concepts in school. With this example, we explain where to look for enhancement of its descriptive quality. Those viewpoints mean also why, in an ontology, we should build a hierarchy of role-concepts discriminated from general properties. However, we haven't finished designing the framework at the level of a development tool. This consideration will be reflected for researching what points to pay attention to when constructing a hierarchy of role-concepts.

Definition of a role-concept

Because a role-concept is recognized as a concept defined in a context, role-concepts should be organized in a hierarchy based on the theory of context dependency. So, in this example (in Fig.6), role-concepts are classified into two categories: genuine role, which is recognized in a given context, and pseudo role which is context free. General role-concept has *participate-in* relation with "holder". It is a basic concept which can carry the role-concept. And, genuine role has other two relations: *participate-in* relation with "context" and *part-of* relation with "role-part". The former describes in what context the role-concept is recognized. The latter is explained in section 4.3 associated with role-aggregation.

Is-a relation in a hierarchy of role-concept

Generally, *is-a* relation is defined between role-concepts which have same category of context dependency. And, super-concept and sub-concept are determined according to concepts as their context (in Fig.7). For example (in Fig.9), "school-staff role" is recognized in "school" context and "staff" in

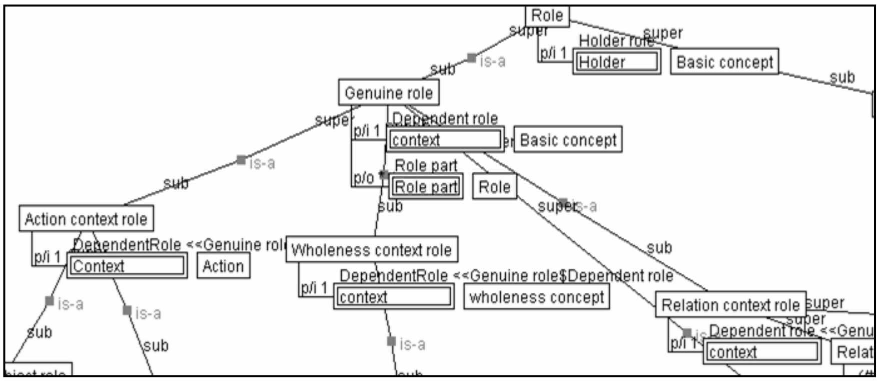
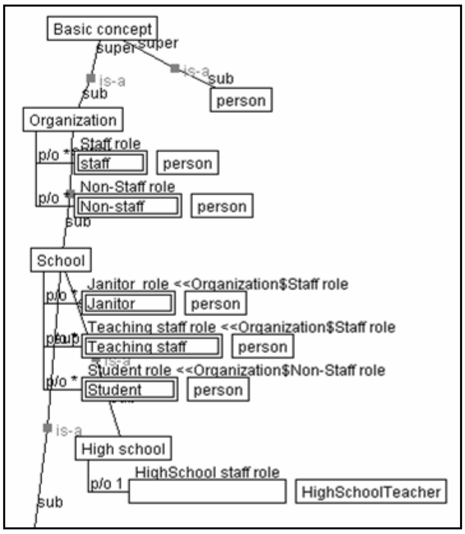


Fig. 7. Organization of role-concepts according to their context



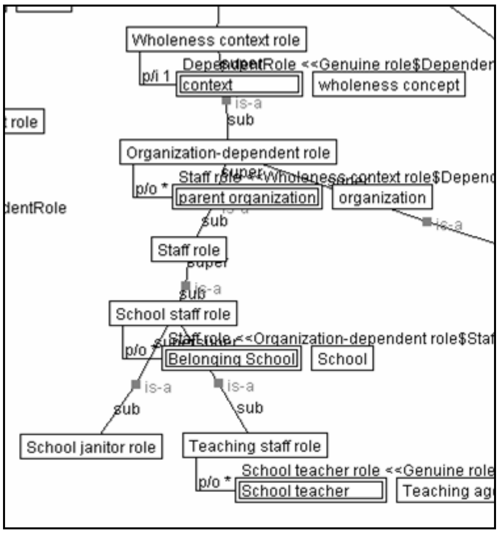


Fig. 8. A hierarchy of basic concepts **Fig. 9.** An example of intermediate role-concepts

"organization" context. Then, <"school staff role" *is-a* "staff role"> is determined according to <"school" *is-a* "organization">. On the other hand, a pseudo role does not have context dependency by strict definition. Although we have some categories of pseudo roles, we only discuss how to define *is-a* relation among them without going into details of them in this paper.

Reference to organization of basic concepts

Some parts of a hierarchy of role-concepts can be organized semi-automatically in synchronized with the construction of a hierarchy of basic concepts. In this example, "high school student role" is defined as a subclass of "student role". This description is based on that, in a hierarchy of basic concepts, "high school student role" as a part of "high school" overrides "student" as a part of "school".

Definition of intermediate concepts

For organizing role-concepts appropriately, it is indispensable to define role-concepts which cannot be described in a hierarchy of basic concepts. In this example (Fig.8, 9), "janitor role" and "teaching staff role" as parts of "school" override "staff role" as a part of "organization". So, in a hierarchy of role-concepts, we can hold <"teaching staff role" *is-a* "staff role"> and <"janitor role" *is-a* "staff role">. On the other hand, according to <"school" *is-a* "organization">, we can describe <"school staff role" *is-a* "staff role">. In this case, "school staff role" is defined as a super-class of "teaching staff role" and "janitor role" by their context dependencies.

Definition of "role-part" for aggregation of role-concepts
Actually, theory of context dependency of a role-concept is more complicated than discussed above. Many of the role-concepts are recognized in several contexts. Those are described in the definition of "role" which has "part-role" as its part (in Fig.6). More details of this topic are discussed in the next section.

4.3 Aggregation of role-concepts

In this section, we present how to organize role-concepts which depends on several contexts. Actually, many of the individuals play a role-concept which depends on several contexts rather than a single context. For example, "Teacher" role is recognized not only as a part of "school" (wholeness context) but also as an agent of "teaching action" (action context). In such a case, we cannot organize role-concepts simply according to their contexts. So, we need to extend our framework.

We assume that if a role-concept is made of several context dependencies, we can choose the most essential context among them. From this point of view, we started to define primitive role-concepts which depend on only one and the essential context one by one. Secondly, we define role-concepts which have several context dependencies as sub-concepts of primitive role-concepts which depend on the same essential context to them. This is based on an ontological theory that an essential attribute is inherited through *is-a* relation. At the same time, in the definition of the role concepts, the other context dependencies are described as participating concepts through *part-of* relations. We call "aggregation of role-concepts" as aggregation of several primitive role-concepts in such a manner.

Here, let us investigate "teacher role" as an example of "aggregation of role-concepts". Firstly, we can find some kinds of "teacher role" in several contexts. For example, there are "school staff role" who belongs to a school, "class teacher role" who teaches students in his/her class, "teaching doctor role" who had got a doctorate, "novice teacher role" who are a beginner in teaching, and so on. And, when we pay attention to a relation between a teacher and a student in school, we can notice another "teacher role". It is "teacher role (for students)" which depends on a relation between a teacher and a student. We assume that this "relational-concept-depend teacher role" is the same one which is treated as "teacher property" in most of the ontology development systems.

Next, let us organize these role-concepts according to their context dependencies. We can extract several primitive role-concepts from these role-concepts regarded as some kind of "a teacher role" mentioned above. For example "school staff role" depends on a *wholeness-concept* "school" as an organization the teacher belongs to, "teaching agent role" depends on an *action-concept* "teach" as an action which he/she plays, "in charge of a class role" depends on a *task-concept* "class" as the task of which he/she is in charge, "teaching qualification role" depends on a *qualification-concept* as a qualification which he/she needs to have to be a teacher, "beginner teacher role" depends on a *state-concept* "beginner" as a state in which he/she is, "teacher role (for students)" depends on a *relation-concept* "teacher-relation" through which he/she associated with students, and so on. We get the taxonomy (in Fig.10) of role-concepts after definition and organization of primitive role-concepts which

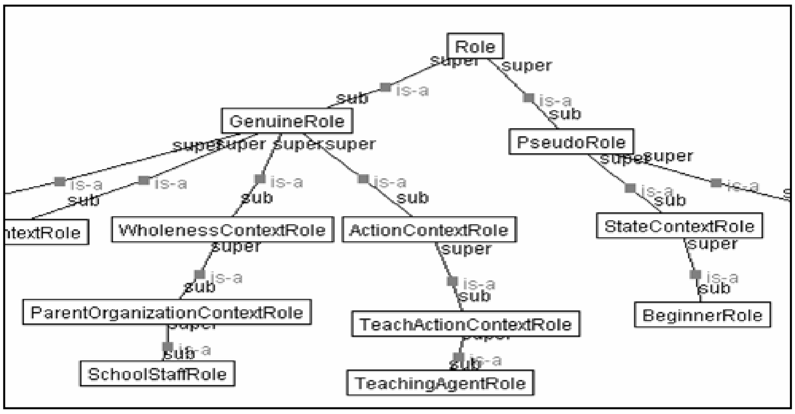


Fig. 10. A hierarchy of primitive role-concepts

depend on the each context.

Here, let us focus on “teaching staff role” and “class teacher role”. Both of them seem to have aggregated several context dependencies. So, after analyzing, we can determine structures of their aggregations as below. Bracketed role-concepts are determined as an essential one.

- Teaching staff role= <school staff role> + teaching agent role
- Class teacher role= <in charge of a class role> + teaching agent role

Needless to say, “teaching agent role” has to be defined somewhere as a primitive role.

Thus, “a teaching staff role” is defined as sub-concepts of “a school-staff role” which is its primitive role. And it has “a teaching agent role” as its part. In this way, we can represent aggregation of role-concepts and organize role concepts.

5 Instances of role-concepts

5.1 What is an instance of a role-concept?

Here, we discuss an instance of a role-concept with an example. Because the role concept is different from classes of the basic concepts, careful investigation is needed. Although we need further detailed investigation on “identify” of instances of a role-concept, we describe our current understanding of the characteristics of the role-concept.

(a) States of an instance of a role concept

An instance of a role concept has the following two states;

(1) *Only the role defined in the role concept is instantiated (realized) without an instance of a basic concept which plays the role (denoted Ri’)*

(2) *An instance of a basic concept plays the instantiated role (denoted Ri)*

Ex: When Mr. P becomes a teaching staff of a school, the post of a teaching staff shifts which should have been instantiated already from open to occupied by him.

To represent such a situation in the instance model, it is necessary to make instances of “person” and “teaching staff role”. First, the instance of role concept is undetermined about who will play it (Ri’). It is just defined as a part of the wholeness concept (school). And, when Mr. P gets a job as the teaching staff, an instance of “teaching staff role” changes its state from (Ri’) to (Ri). It is dealt with as a part of the definition of a teaching staff role-holder. That is represented in Fig.11.

(b) Dependency of instances of role concepts on “context”.

Unlike basic concepts, definition of a role concept is dependent on a basic concept which specifies the context. The existence of an instance of a role concept Ri’ depends on the existence of an instance of the role-context concept (denoted Ci). Thus, the instantiation of Ri’ is done if (and only if) Ci.exists. When Ci has disappeared, so does Ri’. Disappearance of Ri, however, does not caues Disappearance of Ci.

Ex: If the school is closed, Mr. P cannot be a teaching staff belonging to the school any longer.

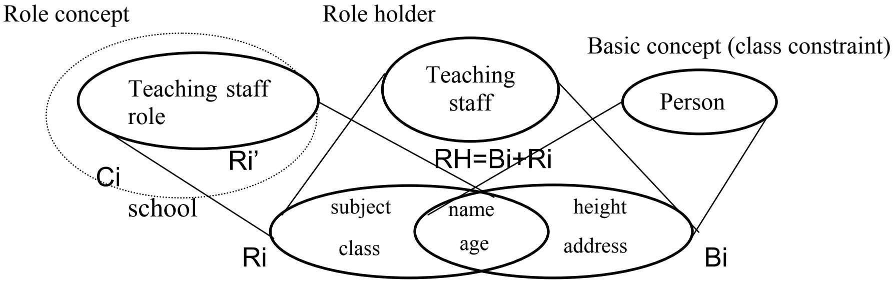


Fig. 11. An example of instances of a role-concept, a basic concept and a role

Because “teaching staff role” (Ri’) is defined in a *part-of* relation of “school”(Ci), this role depends on “school” as its context, an instance of “teaching staff role” cannot be recognized without an instance of “school”.

(c) Dependency of instances of a role holder on “basic concept”

Ri’ is dealt with as a defective instance by itself because values of some properties of Ri are determined by a basic conceptBi which holds (plays) Ri’. And, an instance of the role holder is defined when a basic concept holds Ri’ which then becomes Ri.

Ex: Even if the school is established and teaching programs charted, some properties of a teaching staff might not be fixed yet. For example, although *subject*, *class*, etc. can be fixed, *name*, *age*, *sex*, *years of experience* as a teaching staff cannot before determining the person who occupies the position.

What people have in their mind when they think of an instance of “teaching staff” is a teaching staff role holder, which is a state of a person holding an instance of “teaching staff role” (Ri) (in Fig.5). Ranges or values of some properties of this role holder are fixed in the definition of “school”. The others are not fixed unless some basic concept holds an instance of “teaching staff role” [1]. So, a “teaching staff (role holder)” depends on the basic concept “person” as the holder.

(d) Extinction of a role holder

A role holder is recognized as the summation of instances of a role-concept and a basic concept as its holder. Here, they are denoted Ri and Bi. Then, there are three cases in which a role holder is disappear:

1. *Bi has disappeared. And, Ri becomes to Ri’.*
2. *Ri’ has disappeared. And, Bi exists.*
3. *Bi has stopped holding Ri’. And, Bi exists. Ri is shifted to Ri’.*

Ex: There are 3 cases in which Mr. P is not recognized as a teaching staff any longer. They are 1) when Mr. P has died, 2) when the post as a teaching staff is deleted because of restructuring or closing the school and 3) when Mr. P quits his job as a teaching staff.

A role holder “teaching staff”, which is Mr. P holding “teaching staff role”, is deleted 1) when an instance of “person” (Mr. P) is deleted, 2) an instance of “teaching staff role” is deleted and 3) an instance of “person” stops holding an instance of “teaching staff role”.

5.2 An instance of an aggregate role-concept.

To extend our framework for treating an instance of an aggregate role-concept, we apply the theory of *part-of* relation to the consideration in section 5.1. Generally, when an instance of a whole-concept is created, instances of its all part-concepts are also created. So, it can be said that:

When an instance of an aggregated role-concept is created, an instance of its “role-part” is also created. And, when a basic concept holds an aggregated role-concept, it must hold also an instance of its “role-part”.

Ex: Mr. P teaches information processing in his school. Then, he is recognized as a teaching staff so long as not only he belongs to the school but also he teaches information processing.

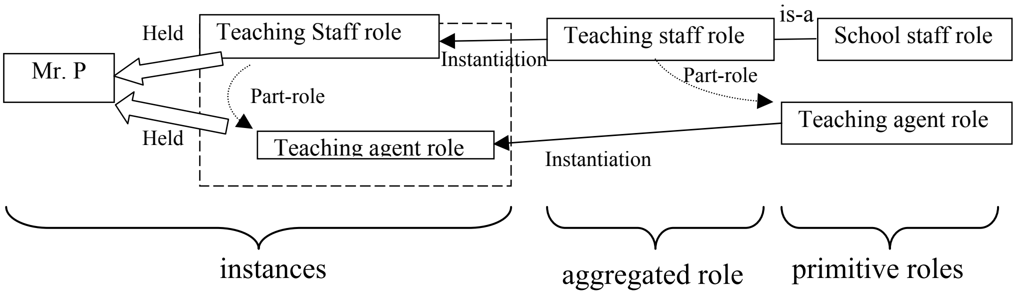


Fig. 12. An instance of an aggregate role-concept

"Teaching staff role" is defined by aggregation of "school staff role" which is a subclass of a wholeness context role and "teaching agent role" which is a subclass of an action context role. The former is located as a super-class of "teaching staff role", and the latter is as "role-part". Then, for holding "teaching staff role", both instances of "school staff role" and "teaching agent role" must be created. And, when Mr. P holds both of them, he is recognized as a role holder of "teaching staff" (in Fig.12).

Furthermore, the theory of *part-of* relation can also apply to the deletion of an instance of aggregate role-concept. If each role holder of primitive role-concept is deleted (in section 5.1), the instance of aggregate role-concept is also deleted. In addition, although we cannot generally define appropriate procedures for all the possible cases of role disappearance, we can cope with it using axioms for each case.

6 How to translate into OWL format

Here, we outline how to translate Hozo the ontologies into OWL format for building the semantic web. Firstly, we define "Wholeness Concept class" (<hozo:WholenessConcept>) and "Relational Concept class" (<hozo:RelationalConcept>) as subclass of the class element of OWL (<owl:Class>). The basic concepts defined in Hozo are expressed as subclass of these two classes. The part-of and participant-in relation are defined as "has Part property" (<hozo:hasPart>) and "has Participant property" (<hozo:hasParticipant>) which are sub-properties of owl property (<owl:ObjectProperty>). And part-concepts of the wholeness concept are defined as property restrictions by using <owl:Restriction>. In the same way, we define "Role Concept class" (<hozo:RoleConcept>) and "Role Holder class" (<hozo:RoleHolder>) as subclass of the class element of OWL (<owl:Class>). Hierarchy of role concepts are defined by subclass of the "Role Concept class" (<hozo:RoleConcept>) , and they are referred in the definition of "Role Holder class" (<hozo:RoleHolder>). And furthermore, the "Role Holder class" (<hozo:RoleHolder>) is use by "has Part property" (<hozo:hasPart>) as < rdfs:range>. Fig.13 shows a translation of an example "teacher role" discussed in section 4. We are considering extending OWL classes to define role concepts and role holders, which support semantics of role concepts mentioned in section 3.

(a) The definition of Basic Concept

```
<owl:Class rdf:ID="school">
   <rdfs:subClassOf rdf:resource="#organization" />
   <rdfs:subClassOf><owl:Restriction>
         <owl:onProperty rdf:resource="hozo:hasPart" />
         <owl:someValuesFrom rdf:resource="#school staff" />
      </owl:Restriction></rdfs:subClassOf>
   <rdfs:subClassOf><owl:Restriction>
         <owl:onProperty rdf:resource="hozo:hasPart" />
         <owl:someValuesFrom rdf:resource="#janitor" />
      </owl:Restriction></rdfs:subClassOf>
   <rdfs:subClassOf><owl:Restriction>
         <owl:onProperty rdf:resource="hozo:hasPart" />
         <owl:someValuesFrom rdf:resource="#student" />
      </owl:Restriction></rdfs:subClassOf>
   ...
</owl:Class>
```

(b) The definition of Role Holder

```
<owl:Class rdf:ID="teaching staff">
   <rdfs:subClassOf rdf:resource="hozo:RoleHolder" />
   <rdfs:subClassOf><owl:Restriction>
         <owl:onProperty rdf:resource="#Role" />
         <owl:someValuesFrom rdf:resource="#teaching staff role" />
     </owl:Restriction></rdfs:subClassOf>
   ...
</owl:Class>
```

(c) The definition of Role Concept

```
<owl:Class rdf:ID="school staff role">
   <rdfs:subClassOf rdf:resource="#staff role" />
   <rdfs:subClassOf> <owl:Restriction>
         <owl:onProperty rdf:resource="#Holder" />
         <owl:someValuesFrom rdf:resource="#person" />
      </owl:Restriction></rdfs:subClassOf>
   <rdfs:subClassOf> <owl:Restriction>
         <owl:onProperty rdf:resource="#context" />
         <owl:someValuesFrom rdf:resource="#school" />
      </owl:Restriction></rdfs:subClassOf>
   ...
</owl:Class>

<owl:Class rdf:ID="teaching staffRole">
   <rdfs:subClassOf rdf:resource="#school staff role" />
   <rdfs:subClassOf> <owl:Restriction>
         <owl:onProperty rdf:resource="#hasRolePart" />
         <owl:someValuesFrom rdf:resource="#teaching agent role" />
      </owl:Restriction></rdfs:subClassOf>
   ...
</owl:Class>
```

Fig. 13. An example of "Teacher Role" translated into OWL format

7 Related work

A theory of roles in ontology engineering is developed also by Guarino [3, 4]. He and his colleagues develop a formal theory for represent specifically roles. His research is concerned with formalities and axiom of an ontology. In contrast, our goal is to develop an effective tool for ontology development. Our notions share a lot of his theory in many aspects of characterization of roles; that is, context dependency, specialization of roles, and so on. By his theory, our framework can be reinforced in terms of axioms and so on. However, roughly speaking, our notion differs from his work on two points; that is, dynamics of a role, and clear discrimination of a role from its player (role holder). Firstly, we focus on context dependency of role-concepts and its categories. So, time dependency of role-concepts is treated implicitly in our framework. As opposed to this, his framework allows to represent it explicitly. Secondly, we distinguish role-concepts and role-holders. On the basis of this distinction, we propose a tool for editing properties and relations on roles, such as an aggregation of role-concepts.

Although there are many ontology development tools and framework for the Semantic Web, for example OntoEdit [21], KAON [22], WebODE [23], Protégé [24] and so on. In some of them relations and properties are differentiated from each other. However, they are differentiated according not to their semantics but to their description form. Most of the tools are based on a frame-based knowledge representation language with an additional functionality for writing axioms. Hozo is similar to them in that sense, but is different from them in some respects:

1. Distinction of role-concepts from properties based on their ontological semantics
2. Discrimination among a role-concept, a role holder and basic concept is done to treat “role” property
3. Hierarchy of role-concepts according to their context dependencies from ontological point of view

8 Conclusion and future work

In this paper, we came up with a framework for organizing role-concepts in their hierarchy according to their context dependencies. Then, we defined instances of role-concepts and investigated their feature. The definitions of the role-concepts can be translated into statements in OWL. In conclusion, our framework provides a layer in which developers can construct ontologies with high quality description of role-concept and a mechanism for setting it in current linguistic expression. Further investigation on the state-dependent role-concepts mentioned in Section 4.1 remains as future work.

References

1. Kozaki, K., et al: Development of an Environment for Building Ontologies which is based on a Fundamental Consideration of "Relationship" and "Role", *Proc. of the 2000 Pacific Rim Knowledge Acquisition Workshop (PKAW2000)*, pp.205-221, Sydney, Australia, December, 2000.
2. K. Kozaki, Y. Kitamura, M. Ikeda, and R. Mizoguchi: Hozo: An Environment for Building/Using Ontologies Based on a Fundamental Consideration of Role” and “Relationship”, *Proc. of the 13th International Conference Knowledge Engineering and Knowledge Management (EKAW2002)*, pp.213-218, Sigüenza, Spain, October 1-4, 2002.
3. C. Masolo et al: Social Roles and their Descriptions. *Proc. of the Ninth International Conference on the Principles of Knowledge Representation and Reasoning (KR2004)*, pp.267-277, Whistler, Canada, June 2-5, 2004.
4. Guarino, N.: Some Ontological Principles for Designing Upper Level Lexical Resources, *Proc. of the First International Conference on Lexical Resources and Evaluation*, Granada, Spain, May 28-30, 1998.
5. John F. Sowa: Top-level Ontological Categories. *International Journal of Human and Computer Studies*, Volume 43, Issue 5-6, Nov/Dec, 1995, pp.669-685.
6. John F. Sowa: Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks/Cole Publishing Co., Pacific Grove, CA, 2000.
7. Guarino, N. and Welty, C.: Evaluating Ontological Decisions with OntoClean, Communications of the ACM 45(2):61-65, 2002.
8. Schreiber, G. et al.: Knowledge Engineering and Management – The CommonKADS Methodology, Cambridge, MA: MIT Press, 2000.

9. Mizoguchi, R., Vanwelkenhuysen, J. and Ikeda, M.: Task Ontology for Reuse of Problem Solving Knowledge, *Proc. of Knowledge Building & Knowledge Sharing 1995(KB&KS'95) (2nd International Conference on Very Large-ScaleKnowledge Bases)*, pp.46-59, Enschede, The Netherlands, 1995.
10. http://www.uml.org/
11. Zambonelli, F., Jennings, N. R. and Wooldridge, M.: Developing Multiagent Systems: the Gaia Methodology, ACM Transactions on Software Engineering and Methodology 12(3): 317-370, 2003.
12. Pacheco, O. and Carmo, J.: A Role Based Model for the Normative Specification of Organized Collective Agency and Agents Interaction, Journal of Autonomous agents and Multi-Agent Systems 6(2):145-184. 2003
13. Biddle, B. J.: Role Theory. Expectations, Identities, and Behaviors, New York: Academic Press, 1979.
14. Tuomera, R.: The importance of us: a philosophical study of basic social notions, Stanford University Press, 1995.
15. Fillmore, C. J.: The case for case, in Universals in Linguistic Theory, Bach, E. and Harms, R., eds., Holt, Rinehart, and Winston: New York, 1968.
16. Barker, C.: Possessive Descriptions, Stanford: CSLI.
17. Fauconnier, G.: Quantification, Roles and Domains, in Meaning and Mental representation, Eco, U. et al. eds. Indiana, 1988.
18. Mizoguchi, R., Kozaki, K., Sano, T., and Kitamura, Y.: Construction and Deployment of a Plant Ontology, *Proc. of 12th International Conference on Knowledge Engineering and Knowledge Management*, pp.113-128, Juan-les-Pins, French Riviera, October, 2000.
19. E. Sunagawa, K. Kozaki, Y. Kitamura, R. Mizoguchi: Management of Dependency Between Two or More Ontologies in an Environment for Distributed Development, *Proc. of the International Workshop on Semantic Web Foundations and Application Technologies (SWFAT2003)*, pp.35-41. Nara, Japan, March 12, 2003.
20. E. Sunagawa, K. Kozaki, Y. Kitamura, R. Mizoguchi: An Environment for Distributed Ontology Development Based on Dependency Management, *Proc. of the 2nd International Semantic Web Conference (ISWC2003)*, pp.453-468, Florida, USA, October 20-23, 2003.
21. Y. Sure, J. Angele, S. Staab.: OntoEdit: Multifaceted Inferencing for Ontology Engineering, *Journal on Data Semantics*, LNCS 2800, Springer, 2003, pp. 128-152.
22. Maedche, S. Staab.: KAON: The Karlsruhe Ontology and Semantic Web Meta Project, *Künstliche Intelligenz. Special Issue on Semantic Web 3/2003*, pp. 27-30.
23. Corcho O, Fernández-López M, Gómez-Pérez A, Vicente O.: WebODE: an integrated workbench for ontology representation, reasoning and exchange, *Proc. of 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW'02)*, pp: 138-153. Sigüenza, Spain, October 1-4, 2002.
24. John H. Gennari, et al: The Evolution of Protégé: An Environment for Knowledge-Based Systems Development http://www.smi.stanford.edu/pubs/SMI_Reports/SMI-2002-0943.pdf