Unification of Sorts Among Local Ontologies for Semantic Web Applications

NWE NI TUN and SATOSHI TOJO
School of Information Science
Japan Advanced Institute of Science and Technology
1-1 Asahidai, Tatsunokuchi, Ishikawa 923-1292
JAPAN
{

Abstract:- Even a single domain can contain a tremendous number of local ontologies with semantic heterogeneity, thus one of the basic problems in the development of techniques for the semantic web is the integration of ontologies. This paper deals with the problem of unification among sorts for the sake of ontology integration and coherent information retrieval in semantic web applications. In this context, a knowledge representation formalism, in order to unify two arbitrary sorts into one, is important. We provide a sorted signature for ontology specification regarding subsumption consistency in a sort hierarchy. In our approach, taxonomies are structured by the set-theoretical inclusion of identity conditions through inheritance. An identity condition (IC) can judge whether two individuals of a sort are identical or not. First, we address some fundamental semantic relations based on the IC sets of sorts. Next, we introduce the *own* IC of a sort, that is, an IC originated by that sort, for our sort unification framework. We deal with the equality of sorts by using sameness and mutual substitution between their own ICs, concerning the views of terminology and subsumption level between local ontologies. We also illustrate a semantic integration model among independent worlds.

Key-Words:- Unification of Sorts, Semantic Heterogeneity, Order-sorted Logic, Taxonomy Integration, Ontology Mapping, Semantic Web

1. Introduction

One of the basic problems in the development of techniques for the semantic web is the integration of ontologies. Indeed, the web incorporates a variety of information sources, and in order to extract information from such sources, their semantic integration and reconciliation is required. In this paper, we deal with a situation in which we have various local ontologies, developed independently from each other. We are required to extract information as a mean of integrating these local ontologies. Thus, the main purposes of this paper are to provide a knowledge representation system for ontologies and to define heuristics for ontology mapping and integrating (see Section 3).

Most of the work carried out on ontologies for the semantic web concerns which ontology language to use for integration on the basis of local ones. For example, the Ontology Inference Layer (OIL) [5, 13] proposes to use an expressive Description Logic (DL) form of ontology language for information exchange in the semantic web.

There are a tremendous number of local ontologies on the web, and for efficient information retrieval in response to a query, we often need to integrate them. A few examples will illustrate the scenario. A web marketplace like $Amazon^{®}$ [1] may need to combine publications from multiple publishing sites into its own. A web portal like $Google^{®}$ [2] may need to proceed a query through multiple ontologies in order to fulfill the user's requirements. A stock exchange agent like $Nasdaq^{®}$ [3] may need to analyze the dynamic statistics of market changes based on several databases across different schemas.

If all ontology engineers and domain experts agreed on a universal standard, the unification of sorts might be not necessary. However, the web has evolved without central editorship and consensus. Since the correspondence between any two communities is inevitably fuzzy even for a single domain, a knowledge representation formalism which deals with the semantic heterogeneity among data repositories as well as with semantic mapping between ontologies is critical.

According to [1], ontological properties are generally classified as:

- *sortals*: No part of the entity has the property. e.g., person, desk, university, book
- *Non-sortals*: Part of the entity has the property. e.g, water, air, gold, red, small, level

The sortal properties are represented as the primitive concepts of a knowledge model and non-sortal properties

[1] http://www.amazon.com

[2] http://www.google.com

[3] http://www.nasdaq.com

become the relations of concepts. In our approach, ontologies are structured by an order-sorted calculus [12] in which concepts are defined as sorts and relations are treated as predicates. We create a correspondence between ontologies and database schemas by interpreting each sort as a class.

The organization of this paper is as follows. In Section 2,we present the specification of ontologies and ontological databases with respect to the sorted signature Σ. In Section 3, we state the semantic relations among sorts concerning their IC sets. Then, we discuss a framework of sort unification based on *own* ICs regarding terminological differences and subsumption levels among local ontologies in independent worlds. In Section 4, we compare our knowledge representation formalism with other related techniques. Finally, in Section 5, we summarize our contribution and discuss our future research.

2 Specification of Ontologies in Order-Sorted Calculus

Order-sorted logic is a first-order predicate logic with many ordered sorts, which leads to the efficient reasoning methods for structural knowledge that ordinary first-order logic lacks. The advantages of order-sorted logic are as follows.

- All formulas with sorted terms can be checked more or less syntactically whether a logic program is *well-typed* [4] or not.
- Sorting reduces the search space of the inference system. We could see the advantage of reasoning methods on sorted predicates and knowledge derivation with multiple knowledge bases in [8].

We also agree the consistency of subsumption relation in a sort hierarchy is important for those advantages [10].

2.1 Overview of Order-Sorted Calculus and Multiple ICs

In [12], we provided the signature and semantics of an order-sorted calculus to formalize taxonomies in sorted logic. Here, we represent an overview of sorted calculus as a short reference for later specification.

The *alphabet* of a first-order language, $\mathcal{L}$, is defined by a tuple $(\mathcal{S}, \sqsubseteq, \mathcal{C}, \mathcal{V}, \mathcal{F}_n, \mathcal{P}_n)$ with a set of *rigid* (γ) and *non-rigid* (λ) sorts including the greatest sort $\top$, subsumption relations ($\sqsubseteq$) on sorts, constants, variables, n-ary functions and predicates respectively. Thereafter, a *sorted signature* $\Sigma = (\mathcal{S}, \sqsubseteq, \Delta)$ is designated where Δ is a set of formulas with sorted terms ($\mathcal{T}$) such that $c : s \in \mathcal{T}_s$, if $t \in \mathcal{T}_s$ and $s \sqsubseteq s'$, then $t \in \mathcal{T}_{s'}$. We give the semantics of sorts and predicates in terms of a possible world because the interpretation of sorts can differ from one world to the next. Thus, we attach w in each interpretation. However, all ontologies given on Σ share the alphabet $\mathcal{L}$.

[4] A program is well-typed iff each argument position of a function or predicate is of an appropriate type.

As we have discussed in [10, 11], a sort can have more than one IC through subsumption relation. We call that concept *multiple ICs* and introduce it as an IC set for each sort. In a formal way, the IC set of a sort s existing in world w is $\mathcal{I}_w(s) = \{\iota_1, \iota_2, ...\}$ where $\iota \in \mathcal{F}_1$ is a unary function which provides the unique identifier for each individual of a sort. For example, $\mathcal{I}_w(person) = \{DNA, fingerprint\}$ where $fingerprint(john : person)$ returns a unique identifier for a specific fingerprint pattern of *person* “john”.

A unary function ι is an *identity condition* (IC) iff:

$$\forall x, y[x = y \leftrightarrow \iota(x : s) = \iota(y : s)]. \qquad (1)$$

We classify ICs into *local ICs* and *global ICs* with respect to the possible world semantics, given a Kripke frame. An IC ι is

- a *Global IC* iff $[\![\iota(a : s)]\!]_w = [\![\iota(a : s)]\!]_{w'}$ for any a in wRw', and
- a *Local IC* iff it is not a global IC.

The difference is that a global IC can retain its value through all accessible worlds. For example, *fingerprint* is a global IC and *studentID* is a local IC. The ICs are intertwined with *rigidity* that can decide the scope of essentiality for a sort. Formally, a sort s is *rigid* iff $\models \Box\forall x[(x : s) \rightarrow \Box(x : s)]$ (+**R**); otherwise *non-rigid* ($-$**R**). For example, “person is rigid” means every person is a person in every world, and “student is non-rigid” means every student is not necessarily a student in every world. Thus, we divide sorts into two categories: rigid sorts ($\Gamma \subseteq S$) and non-rigid sorts ($\Lambda \subseteq S$) such that $\Gamma \cap \Lambda = \emptyset$ and $\Gamma \cup \Lambda = \mathcal{S}$.

2.2 Specification of Formal Ontologies

Now, we give the specification of ontologies with respect to the sorted signature Σ.

Definition 1 (Ontology) *An ontology $\mathcal{O}$ is a tuple $(\mathcal{S}, \sqsubseteq, I, \Delta)$ where I is a set of identity conditions on $\mathcal{S}$.*

Definition 2 (Local Ontology) *A local ontology $\mathcal{O}_L = (\mathcal{S}_w, \sqsubseteq_w, I_w, \Delta_w)$ is an ontology designed in a specific world $w \in \mathcal{W}$.*

For any sort $s \in \mathcal{S}_w$, there is an IC set $\mathcal{I}_w(s)$ that consists of *carried ICs* inherited from parent sorts, as well as *own IC* originated by itself. We discuss *own ICs* in the next section. The subsumption relation ($\sqsubseteq_w$) between any two sorts is consistently defined by IC inheritance.

$$s \sqsubseteq_w s' \quad \text{iff} \quad \mathcal{I}_w(s') \subseteq \mathcal{I}_w(s) \qquad (2)$$

Example 1 *Suppose $\mathcal{O}_{l_1} = (\mathcal{S}_{w_1}, \sqsubseteq_{w_1}, I_{w_1}, \Delta_{w_1})$ is a local ontology in world w_1 for the university domain. The specification of $\mathcal{O}_{l_1}$ is as follows.*
$\mathcal{S}_{w_1}$ ={professor, staff, student, human, university, course, $\top$}
$\sqsubseteq_{w_1} = \{professor \sqsubseteq_{w_1} human, student \sqsubseteq_{w_1} human \sqsubseteq_{w_1} \top, university \sqsubseteq_{w_1} \top, course \sqsubseteq_{w_1} \top\}$
I_{w_1} ={fingerprint, professorID, studentID, courseID, universityID} such that
$\mathcal{I}_{w_1}(human) = \{fingerprint\}$,
$\mathcal{I}_{w_1}(professor) = \{fingerprint, professorID\}$,
$\mathcal{I}_{w_1}(student) = \{fingerprint, studentID\}$,
$\mathcal{I}_{w_1}(university) = \{universityID\}$,
$\mathcal{I}_{w_1}(course) = \{courseID\}$.
$\Delta_{w_1} = \{belong\_to, teach, study, advise\}$ in the forms of
$\exists x, y$ belong_to(x:human, y:university),
$\exists x, y$ teach(x:professor, y:course),
$\exists x, y$ study(x:student, y:course),
$\forall x$ advise(supervisor_of(x:student):professor, x:student).

There may be a number of databases (or knowledge bases) which apply the local ontology in a specific world. We call them *ontological databases*.

Definition 3 (Ontological Database) *Given a local ontology $\mathcal{O}_l$, an ontological database is a tuple $\mathcal{O}_{DB} = (\mathcal{O}_l, \mathcal{D}_w, [\![\ ]\!]_w)$ where $\mathcal{D}_w$ is the domain of $\mathcal{O}_{DB}$ such that $\mathcal{D}_w \subseteq \mathcal{U}$ ($\mathcal{U}$ is a universal set of individuals) and $[\![\ ]\!]$ is the interpretation of $\mathcal{O}_L$ in world w.*

In any ontological database $\mathcal{O}_{DB}$, the following interpretations are consistent.

$$if s \in \mathcal{S}_w \quad \text{then} \quad [\![s]\!]_w \subseteq [\![\mathcal{D}]\!]_w \tag{3}$$
$$s \sqsubseteq s' \quad \text{iff} \quad [\![s]\!]_w \subseteq [\![s']\!]_w \tag{4}$$

Next, we consider a global ontology for all local ontologies given a well-defined Kripke frame.

Definition 4 (Global Ontology) *A global ontology in world w is a tuple $\mathcal{O}_g = (\Gamma, \sqsubseteq, I, \Delta)$ in which all sorts are rigid and $\mathcal{O}_g \subseteq \mathcal{O}_l$ in every accessible world $w_i \in \mathcal{W}$ such that wRw_i.*

Example 2 *Consider an ontological model given in a well-defined Kripke frame $\langle \mathcal{W}, \mathcal{R} \rangle$ where $\mathcal{W} = \{w, w_1, w_2, w_3\}$ with $\mathcal{R} = \{wRw_i\}, 1 \leq i \leq 3$. By Definition 4, there is a global ontology ($\mathcal{O}_g$) in world w which is applied to every local ontology ($\mathcal{O}_l$) in world $w_i \in \mathcal{W}$. For any rigid sort $\gamma \in \Gamma_w, \mathcal{I}_w(\gamma) = \mathcal{I}_{w_i}(\gamma)$.*

Additionally, the following set-theoretical inclusion is satisfied among all ontological databases in the above example.

$$[\![\gamma]\!]_w \supseteq [\![\gamma]\!]_{w_i} \quad and \quad [\![\mathcal{D}]\!]_w \supseteq [\![\mathcal{D}]\!]_{w_i} \tag{5}$$

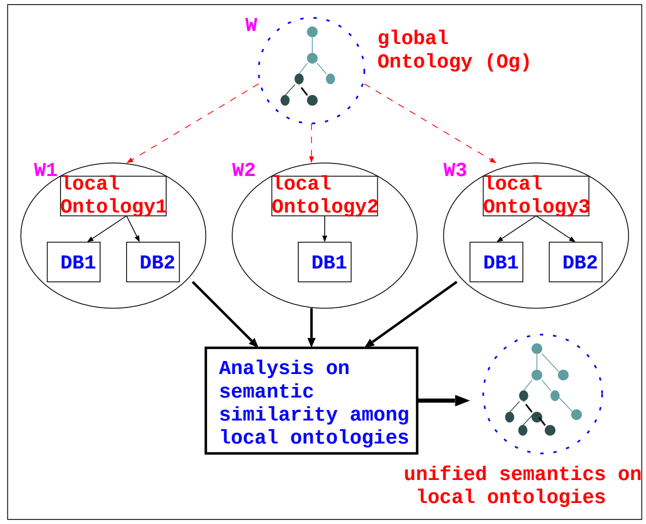

Figure 1. An Example Application Model for Ontology Integration

The upper part of Figure 1 illustrates a model of ontologies and related databases locally designed in different worlds. If the accessibility relations ($\mathcal{R}$) are known among worlds $w \in \mathcal{W}$, the concept of a global ontology is very useful for ontology mapping and integration process. However, ontologies may be designed in each local world without any accessibility relation between each other. Therefore, equation (5) may not be applicable in such a knowledge model. We need to consider another solution to integrate semantic heterogeneity among local ontologies with *unknown* accessibility relations to each other. The next section presents a detailed discussion of ontology integration given a Kripke frame $(\mathcal{W}, \mathcal{R})$ with $\mathcal{W}\{w_i\}$, $1 \leq i \leq n$, and $\mathcal{R} = \{w_i R w_i\}$.

3 Framework of Sorts Unification

Concerning ontology mapping and integrating among independent worlds, three kinds of semantic relation can be generally identified as follows.

- **Disjoint Relation:** There is no intersection between the IC sets of two given sorts $s_i \in \mathcal{S}_{w_i}$ and $s_j \in \mathcal{S}_{w_j}$.

$$s_i \asymp s_j \quad \text{iff} \quad \mathcal{I}_{w_i}(s_i) \cap \mathcal{I}_{w_j}(s_j) = \emptyset \tag{6}$$

- **Overlap Relation:** There is an intersection between the IC sets of two given sorts $s_i \in \mathcal{S}_{w_i}$ and $s_j \in \mathcal{S}_{w_j}$.

$$s_i \cong s_j \quad \text{iff} \quad \mathcal{I}_{w_i}(s_i) \cap \mathcal{I}_{w_j}(s_j) \neq \emptyset \tag{7}$$

- **Equality Relation:** There is an equivalent relation between the IC sets of two given sorts $s_i \in \mathcal{S}_{w_i}$ and $s_j \in \mathcal{S}_{w_j}$.

$$s_i \equiv s_j \quad \text{iff} \quad \mathcal{I}_{w_i}(s_i) = \mathcal{I}_{w_j}(s_j) \qquad (8)$$

There are two possible issues for the above relations especially with regard to equation (8).

1. *Terminological Difference:* How do we make the heuristics defined for each relation to be applicable if the names (or terms) of the sorts and their ICs do not exactly match? For example, *staff* and *employee* are terminologically different but semantically the same as the person who is employed in an organization.

2. *Subsumption level:* The subsumption level of a sort among local ontologies can not always be the same. For example, $\mathcal{I}_{w_i}(professor) \neq \mathcal{I}_{w_j}(professor)$ where $professor \sqsubseteq_{w_i} staff \sqsubseteq_{w_i} human$ and $professor \sqsubseteq_{w_j} faculty \sqsubseteq_{w_j} employee \sqsubseteq_{w_j} person$.

Our solution is based on the notion of *own* IC because the own IC of a sort is created only by that sort and it is very limited. The role of *own* IC overcomes the problem caused by the subsumption level.

Definition 5 (Own IC) *Let $\iota^{+o} \in \mathcal{I}_w(s)$ be an own IC of sort s iff $\iota^{+o} \notin \mathcal{I}_w(s')$ where $s' \not\sqsubseteq_w s$.*

Example 3 *According to Definition 5, the IC sets of $\mathcal{O}_{l_1}$ in example 1 becomes*
$\mathcal{I}_{w_1}(human) = \{fingerprint^{+o}\}$,
$\mathcal{I}_{w_1}(prof) = \{fingerprint, professorID^{+o}\}$,
$\mathcal{I}_{w_1}(student) = \{fingerprint, studentID^{+o}\}$,
$\mathcal{I}_{w_1}(university) = \{universityID^{+o}\}$,
$\mathcal{I}_{w_1}(course) = \{courseID^{+o}\}$.

Normally, each sort is identified by a single own IC, for example, $computer\{product\_key^{+o}\}$, $book\{ISBN^{+o}\}$, $homepage\{URL^{+o}\}$, etc. However, some sorts can have more than one own IC, such as $person\{fingerprint^{+o}, iris\_pattern^{+o}\}$. In the former case, we focus on the terminological difference and show the sameness of own ICs for the equality relation. Each individual of a sort could have the same IC value as the next own IC, if those ICs are semantically equivalent. In the latter case, the IC values of each own IC can be totally different since the semantics of each own IC is quite different. However, the own ICs of sorts may be able to substitute mutually, if they are own ICs of the same sort.

In order to show the *sameness* and *mutual substitution* between two own ICs, we use an approach in which the own ICs of sorts s_i and s_j are temporarily shared by each sort interchangeably. Before that, the system must be sure that $\iota_j \notin \mathcal{I}_{w_i}(s_i)$ and $\iota_i \notin \mathcal{I}_{w_j}(s_j)$. In this approach, we assume that necessary data sets for ι_i^{+o} in w_j concerning with $[\![s_j]\!]_{w_j}$ and ι_j^{+o} in w_i concerning with $[\![s_i]\!]_{w_i}$ are possibly to be created.

Let each $\iota_i{}^{+o}$ and $\iota_j{}^{+o}$ be own ICs of two arbitrary sorts s_i and s_j from different local ontologies.

Definition 6 (Sameness of own ICs) *There is the sameness of own ICs between $\iota_i{}^{+o}$ and $\iota_j{}^{+o}$:*
$\iota_i{}^{+o} \doteq \iota_j{}^{+o}$ *iff* $\forall x : s_i\ [\iota_i{}^{+o}(x) = \iota_j{}^{+o}(x)]$ *and* $\forall y : s_j\ [\iota_j{}^{+o}(y) = \iota_i{}^{+o}(y)]$.

Definition 7 (Mutual Substitution of Own ICs) *There is a mutual substitution between $\iota_i{}^{+o}$ and $\iota_j{}^{+o}$:*
$\iota_i{}^{+o} \bowtie \iota_j{}^{+o}$ *iff* $(\ \forall x_i, y_i : s_i\ [\iota_j{}^{+o}(x_i) = \iota_j{}^{+o}(y_i) \leftrightarrow x_i = y_i]$ *and* $\forall x_j, y_j : s_j\ [\iota_i{}^{+o}(x_j) = \iota_i{}^{+o}(y_j) \leftrightarrow x_j = y_j]\ )$.

Now, we modify the equality relation ($\equiv$) of the two sorts in equation 8 concerning with their own ICs.

Definition 8 (Equality of Sorts) *Sorts s_i and s_j have equality of sorts by either a sameness or a mutual substitution relation between their own ICs. In a formal way,*

- *Equality of Sorts by Same Own ICs($\equiv$):*
  $s_i \equiv s_j$ *iff* $\iota_i{}^{+o} = \iota_j{}^{+o}$.

- *Equality of Sorts by Sameness ($\dot{\equiv}$):*
  $s_i \dot{\equiv} s_j$ *iff* $\iota_i{}^{+o} \doteq \iota_j{}^{+o}$.

- *Equality of Sorts by Mutual Substitution ($\overset{\bowtie}{\equiv}$):*
  $s_i \overset{\bowtie}{\equiv} s_j$ *iff* $\iota_i{}^{+o} \bowtie \iota_j{}^{+o}$.

Both $\dot{\equiv}$ and $\overset{\bowtie}{\equiv}$ are the variations of the equality relation ($\equiv$) based on the correspondence between their own ICs. After having equality, the IC set of sorts become the form $\mathcal{I}_{w_i}(s_i) = \mathcal{I}_{w_j}(s_j) = \{\iota_{ij}^{+o}, ...\}$ for the sameness relation ($\doteq$) where ι_{ij}^{+o} is a transformed own IC of ι_i^{+o} and ι_j^{+o}, and $\mathcal{I}_{w_i}(s_i) = \mathcal{I}_{w_j}(s_j) = \{\iota_i^{+o}, \iota_j^{+o}, ...\}$ for the equality of sorts with mutual substitution relation ($\bowtie$).

Regarding the above heuristics, we consider the following cases in our sort unification algorithm. Every case is labeled in Figure 2.

1. *Any two arbitrary sorts with same sort names and same own IC names.*

2. *Any two arbitrary sorts with same sort names and different own IC names.*

3. *Any two arbitrary sorts with different sort names and same own IC names.*

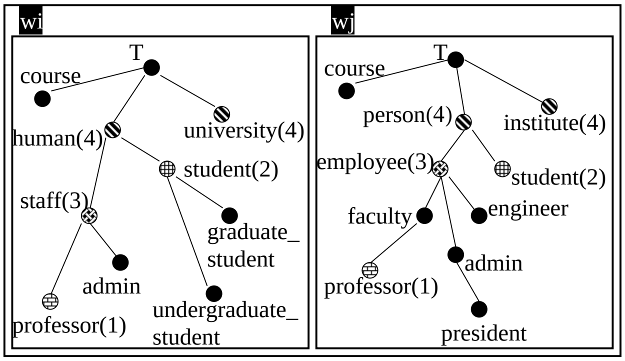

Figure 2. General Cases in Sorts Unification

4. *Any two arbitrary sorts with different sort names and different own IC names*. In this case, both sameness and mutual substitution relations should be detected.

Algorithm: Unification of two sorts by their own ICs

Input
s_i with ${\iota_i}^{+o} \in \mathcal{I}_{w_i}(s_i)$, s_j with ${\iota_j}^{+o} \in \mathcal{I}_{w_j}(s_j)$
Output
$s_i \equiv s_j$ or $s_i \dot{\equiv} s_j$ or $s_i \overset{\bowtie}{\equiv} s_j$ or $s_i \not\equiv s_j$
Begin
1. n $\leftarrow$ pattern_match(s_i,s_j);
2. m $\leftarrow$ pattern_match(${\iota_i}^{+o}$,${\iota_j}^{+o}$);
   Note:{1=True/0=False}
3. If (n=1 and m=1) Then caseno $\leftarrow$ 1;
4. If (n=1 and m=0) Then caseno $\leftarrow$ 2;
5. If (n=0 and m=1) Then caseno $\leftarrow$ 3;
6. If (n=0 and m=0) Then caseno $\leftarrow$ 4;
7. Switch (caseno)
   Case 1,3: $s_i \equiv s_j$;
   Case 2 :If (${\iota_i}^{+o} \doteq {\iota_j}^{+o}$) Then $s_i \dot{\equiv} s_j$;
   Else $s_1 \not\equiv s_j$;
   EndIf.
   Case 4 :If (${\iota_i}^{+o} \doteq {\iota_j}^{+o}$) Then $s_i \dot{\equiv} s_j$;
   Else If (${\iota_i}^{+o} \bowtie {\iota_j}^{+o}$)
   Then $s_i \overset{\bowtie}{\equiv} s_j$;
   Else $s_1 \not\equiv s_j$;
   EndIf
   EndIf
   SwitchEnd
8. If ($s_i \dot{\equiv} s_j$) Then $\mathcal{I}_{w_i}(s_i) \leftarrow \{{\iota_{ij}}^{+o}, \ldots.\}$;
   $\mathcal{I}_{w_j}(s_j) \leftarrow \{{\iota_{ij}}^{+o}, \ldots.\}$;
   EndIf
9. If ($s_i \overset{\bowtie}{\equiv} s_j$) Then $\mathcal{I}_{w_i}(s_i) \leftarrow \{{\iota_i}^{+o}, {\iota_j}^{+o}, ..\}$;
   $\mathcal{I}_{w_j}(s_j) \leftarrow \{{\iota_i}^{+o}, {\iota_j}^{+o}, ..\}$;
   EndIf
End.
Remark: If a sort has multiple own ICs in each world, the system may pick up one own IC of them in randomly for the sort unific_ ation process.

The complexity of the sort unification algorithm depends on the number of individuals related to each sort in the given ontological databases. Let the maximum number of individuals for sort s_i be n and the maximum number of individuals belongs to sort s_j be m. Assume that $m \leq n$. The time complexity to check the sameness or mutual substitution between the given own ICs of s_i and s_j is $O(n)$. Suppose the maximum number of sorts in each local ontology ($\mathcal{O}_l$) is k. Also assume that $k \leq n$. Thus, the time complexity to unify the semantics of any two local sorts is $O(n^2)$.

Example 4 *Let* $\mathcal{O}_{l_2} = (\mathcal{S}_{w_2}, \sqsubseteq_{w_2}, I_{w_2}, \Delta_{w_2})$ *is a local ontology in world* w_2 *for the university domain. The specification of* $\mathcal{O}_{l_2}$ *that distinguishes it from* $\mathcal{O}_{l_1}$ *is as follows.*
$\mathcal{S}_{w_2}$=*{professor, faculty, employee, admin_staff, president, student, person, institute, course,* $\top$*}*
$\sqsubseteq_{w_2} = \{professor \sqsubseteq_{w_2} faculty \sqsubseteq_{w_2} employee \sqsubseteq_{w_2} person, president \sqsubseteq_{w_2} admin\_staff \sqsubseteq_{w_2} person, student \sqsubseteq_{w_2} person, person \sqsubseteq_{w_2} \top, institute \sqsubseteq_{w_2} \top, course \sqsubseteq_{w_2} \top\}$
I_{w_2}=*{iris_pattern, facultyID, employeeID, professorID, adminID, presidentID, enrollID, instituteID, courseID}* *such that* $\mathcal{I}_{w_1}(person) = \{iris\_pattern^{+o}\}$,
$\mathcal{I}_{w_2}(employee) = \{iris\_pattern, employeeID^{+o}\}$,
$\mathcal{I}_{w_2}(faculty) = \{iris\_pattern, employeeID, facultyID^{+o}\}$, $\mathcal{I}_{w_2}(professor) = \{iris\_pattern, employeeID, facultyID, professorID^{+o}\}$,
$\mathcal{I}_{w_2}(admin\_staff) = \{iris\_pattern, employeeID, adminID^{+o}\}$, $\mathcal{I}_{w_2}(president) = \{iris\_pattern, employeeID, adminID, presidentID^{+o}\}$,
$\mathcal{I}_{w_2}(student) = \{iris\_pattern, enrollID^{+o}\}$,
$\mathcal{I}_{w_2}(institute) = \{instituteID^{+o}\}$,
$\mathcal{I}_{w_2}(course) = \{courseID^{+o}\}$.
$\Delta_{w_2} = \{belong\_to, teach, study, advise, manage\}$ *where* $\exists x, y$ *manage(x:president, y:institute).*

After the process of sort unification between $\mathcal{O}_{l_1}$ and $\mathcal{O}_{l_2}$ by consulting with their related ontological databases, the equality relations between labeled sorts in Figure 2 can be identified as follows.

1. $professor \equiv professor$ with $professorID = professorID$ in case 1 by Definition 8.

2. $student \dot{\equiv} student$ with $studentID \doteq enrollID$ in case 2 by Definition 6.

3. $staff \equiv employee$ with $employeeID = employeeID$ in case 3 by Definition 8.

4. $university \dot{\equiv} institute$ with $universityID \doteq instituteID$ in case 4 by Definition 6.

5. $human \overset{\bowtie}{\equiv} person$ with $fingerprint \bowtie iris\_pattern$ in case 4 by Definition 7.

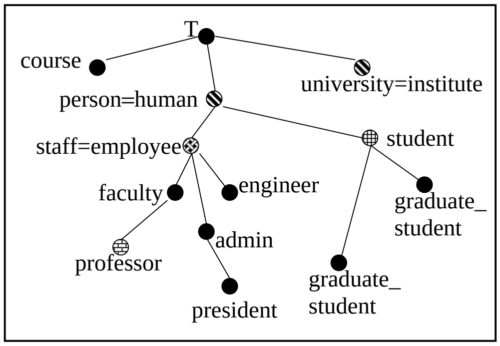

Figure 3. Unified View on the Local Ontologies of w_i and w_j

After the unification of sorts, each local ontology $\mathcal{O}_l$ is necessarily updated by equation (2). The following heuristic is applied in the later ontology *integration* process as regards subsumption relation.
If $(s \sqsubseteq_{w_i} s''\ and\ s \sqsubseteq_{w_j} s'\ and\ s' \sqsubseteq_{w_j} s'')$

$$Then\ (s \sqsubseteq_{w_{ij}} s'\ and\ s' \sqsubseteq_{w_{ij}} s'') \quad (9)$$

Definition 9 (Integrated Ontology) *Suppose there are a local ontology and its related ontological databases in each world* $w \in \mathcal{W}$ *such that* $\mathcal{O}_{l_w} = (\mathcal{S}_w, \sqsubseteq_w, I_w, \Delta_w)$ *with* $\sum_{i=1}^{n} \mathcal{O}_{DB_i}(\mathcal{O}_{l_w}, \mathcal{D}_w, [\![\ ]\!]_w)$. *For any two worlds* $w_i, w_j \in \mathcal{W}$ *where* $i \neq j$, *there is an integrated ontology* $\mathcal{O}^{+}_{L_{ij}} = (\mathcal{S}_{w_{ij}}, \sqsubseteq_{ij}, I_{w_{ij}}, \Delta_{w_{ij}})$ *such that* $\mathcal{O}^{+o}_{l_{ij}} = \mathcal{O}_{l_i} \cup \mathcal{O}_{l_j}$. *Additionally,*

$$For\ any\ s \in \mathcal{S}_{w_{ij}},\ [\![s]\!]_{w_{ij}} = [\![s]\!]_{w_i} \cup [\![s]\!]_{w_j}. \quad (10)$$

Therefore, we can extract all relevant information for a query through an integrated ontology among local worlds. The unified view of the local ontologies in worlds w_i and w_j is shown in Figure 3. Our unification framework can also define the accessibility relations among worlds.
If $(s_i \equiv s_j\ and\ [\![s_i]\!]_{w_i} \subseteq [\![s_j]\!]_{w_j})$

$$Then\quad s_i, s_j \in \Gamma\ and\ w_i R w_j \quad (11)$$

The equation (11) mentions that two given sorts $s_i \in \mathcal{S}_{w_i}$ and $s_j \in \mathcal{S}_{w_j}$ can be identified as *rigid* sorts if they have not only sort equality but also domain inclusion through worlds. Consequently, they navigate the accessibility relation between their worlds in order to construct a Kripke frame.

Initially, a Kripke frame $(\mathcal{W}, \mathcal{R})$ can be defined for the independent worlds only in a reflexive form where $wRw \in \mathcal{R}$ for any $w \in \mathcal{W}$. After the unification process, other accessibility relations between them such as transitive, Euclidean, etc., are known to complete the given Kripke frame.

4 Related Work

We have argued that mapping between concepts (or sorts) is essential for the integration of local ontologies. Prior related research can be considered from two different points of view. One perspective is the representation of ontologies and reasoning among them. The subject "what features are important in representing ontologies for reasoning" is addressed here. **OBSERVER** [4] is a project that uses ontologies to allow queries against heterogeneous sources. It applies ontologies represented in *Description Logic* (DL) to replace terms in user queries with formal concepts. In the case of the mapping process, the semantic relationships (synonyms, hyponyms, and hypernyms) between concepts are predefined.

One of the projects in the framework of Semantic Web, **SHOE** [6, 7], uses HTML tags as its extensions and provides its own representation formalism for ontologies and reasoning based on semantic search. SHOE emploies DL to define ontologies with ontology extension. **Ontobroker** [14] is similar to SHOE in many respects. It is based on frame-based logic. Although their work is comprehensive for semantic search and information integration based on ontologies, their weakness is a lack of heuristics to identify semantic relations between ontological concepts.

The next perspective concerns the semantic integration of database schemas. It addresses the issue of relating ontologies with existing database schemas [5] for the purpose of information integration. The aim of the integration process is the development of a global schema which integrates and subsumes the local schema in such a way that users are provided with a uniform and correct view of the federated databases. Recent works in this field use a varety of heuristics to find mapping [2, 9]. A machine learning technique, support vector machine, is used to automate the ontology mapping process in [3] by deriving a similarity between concepts based on their extensions.

In this paper, we have proposed a knowledge representation formalism in order-sorted logic concerning heterogeneous schemas. The advantages of this work compared to other attempts are:

- providing efficient reasoning capabilities in semantic integration because of the formalization of ontologies in terms of *order-sorted logic*;
- presenting two approaches: *a global ontology* for a well-defined Kripke frame, and *a sort unification* framework with prior unknown accessibility relations among worlds.

[5] Schemas are definitions that specify the structure of data and are the result of a database design phase.

- providing sort unification utilities as the heuristics for the ontology integration and mapping among independent worlds;

5 Conclusion

Our main contribution is to show that the specification of ontologies given on a sorted signature Σ can be effectively exploited to boost ontology integration using a sort unification algorithm. In our future work, we are planning to address the following problems:

- We need to accelerate our unification algorithm and extend it to a full functional ontology integration system for semantic globalization.

- We need to consider the specification of ontology reuse that enhances the achievement of a global schema.

Acknowledgement

The authors acknowledge with thanks the support by Fostering Talent in Emergent Research Fields, Special Coordination Funds for Promoting Science and Technology, Ministry of Education, Culture, Sports, Science and Technology, Japan.

References:

[1] C. Welty and N. Guarino. Supporting Ontological Analysis of Taxonomic Relationships. *Data & Knowledge Engineering*, vol.39, pp.51-74, Elsevier 2001.

[2] D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder. *The Chimaera Ontology Environment*. In Proceedings of *The 17th National Conference on Artificial Intelligence (AAAI)*, Austin, TX, pp.1123-1124, 2000.

[3] D. Zhang and W. S. Lee. *Web Taxonomy Integration using Support Vector Machine*. *WWW 2004, May 17-22, New York, USA, ACM 1-58113-844-X/04/0005*, 2004.

[4] E. Mena, V. Kashyap, A. Illarramendi, and A. Sheth. *Domain Specific Ontologies for Semantic Information Brokering on the Global Information Infrastructure*. *In Nicola Guarino, Editor, Formal Ontology in Information Systems*, IOS press, 1998.

[5] I. Horrocks, D. Fensel, J. Broekstra, S. Decker, M. Erdmann, C. Goble, F. V. Harmelen, M. Klein, S. Stabb, R. Studer, and E. Motta. *The Ontology Inference Layer OIL*. *http://www.ontoknowledge.org/oil*, 2000.

[6] J. Heflin and J. Hendler. *Semantic Interoperability on the Web*. *In Extreme Markup Languages 2000, http://www.cs.umd.edu/projects/plus/SHOE/pubs/extreme2000.pdf*, 2000.

[7] J. Heflin and J. Hendler. *Towards the Semantic Web: Knowledge Representation in a Dynamic, Distributed Environment*. *Doctoral Dissertation Thesis, http://www.cs.umd.edu/projects/plus/SHOE/pubs/heflin-thesis-orig.pdf*, 2001.

[8] K. Kaneiwa and R. Mizoguchi. Ontological Knowledge Base Reasoning with Sort-Hierarchy and Rigidity. In Proceedings of *Knowledge Representation (KR2004)*, 2004.

[9] N. F. Noy and M. A. Musen. *PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment*. In Proceedings of *The National Conference on Artificial Intelligence (AAAI)*, Austin, TX, pp.450-455, 2000.

[10] N. N. Tun and S. Tojo. *Consistency Maintenance for Ontological Knowledge Updating*. In Proceedings of *IEEE International Conference on Information Reuse and Integration (IRI-2004), pp.388-393*, 2004.

[11] N. N. Tun and S. Tojo. *Inheritance of Multiple Identity Conditions in Order-Sorted Logic*. In Proceedings of *17th Australian Joint International Conference on Artificial Intelligence (AI 2004), Springer, LNAI 3339: Advances in Artificial Intelligence, pp.1187-1193*, 2004.

[12] N. N. Tun and S. Tojo. *Consistent Hybrid Knowledge Representation in Order-Sorted Logic*. In Proceedings of *International Conference on Computational Intelligence (ICCI 2004), IJCI Press, pp.349-353*, 2004.

[13] S. Decker, D. Fensel, F. Van Harmelen, I. Horrocks, S. Melnik, M. Klein, and J. Broekstra. *Knowledge Representation on the Web*. In Proceedings of *The 2000 Description Logic Workshop (DL 2000)*. http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-33, pp.89-97 2004.

[14] V. Richard Benjamins and Dieter Fensel. *The Ontological Engineering Initiative*. *In Nicola Guarino, Editor, Formal Ontology in Information Systems, pp. 287-301, IOS Press*, 1998.