A Logical and Ontological Framework for Compositional Concepts of Objects and Properties

Ken KANEIWA

Department of Communication Engineering and Informatics
The University of Electro-Communication
1-5-1 Chofugaoka, Chofu-shi, Tokyo 182-8585, Japan

kaneiwa@uec.ac.jp

Riichiro MIZOGUCHI

Research Center for Service Science
Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan

mizo@jaist.ac.jp

Philip H.P. NGUYEN

Attorney-General's Department, Government of South Australia
45, Pirie Street, Adelaide, SA 5000, Australia

philip.nguyen@sa.gov.au

Abstract In order to formalize complex and compositional concepts, we propose a logical framework based on an upper ontology constructed from the composition of basic concepts such as properties and objects. In particular, ontologically distinct compositions (called ontological compositions) that are not easily defined by using ISA and PART-OF relations are classified into characterizing, temporal, and spatial compositions (e.g., 'red face' and 'today'). In this paper, we precisely model such ontological compositions by using monadic second-order logic; properties and objects are expressed as predicates, and attributes are expressed as predicates of predicates. The proposed approach provides a novel technique for the classification of attributes as higher-order concepts, and it clarifies illegal compositions of properties and objects and the uniqueness of temporal attributes. Moreover, our composition ontology is described by a set of RDF triples using the metamodeling of concepts in RDF Schema.

§1 Introduction

In the field of formal ontology,[22, 14] basic concepts have been introduced and categorized on the basis of philosophical notions. For example, the *sortality* discussed in [8] can determine whether or not a concept is sortal or non-sortal; furthermore, in order to treat the sortality in a logical reasoning system, the notion of sortality is embedded in the syntax and semantics of order-sorted logic.[10, 9, 12] The sortality enables us to distinguish between the following two concepts: property (non-sortal) and object (sortal). Combining two such concepts leads to a more complex concept, e.g., 'red face' and 'tall human' where 'red' and 'tall' are properties and 'face' and 'human' are objects. This is useful for creating new concepts compositionally and infinitely.

However, the existing approaches do not provide us any guide to characterize the compositions of different types of concept (i.e., the combined features of a property and an object). The composition of concepts is syntactically simple, but semantically complex. For example, if the composition 'red face' is expressed by the logical conjunction $red(x) \wedge face(x)$ in FOL, it does not express the nature of the composition, ontologically. In our notion, the composition 'red face' is ontologically legal if the property 'red' as part of the composition is an instance of the attribute 'face color.' Hence, the composition 'red face' must be interpreted through the attribute 'face color' but it is a higher-order entity obtained as a subset of the attribute 'color'.

Moreover, let 'now' be a property and 'date' be an object. Then, 'today' is regarded as the composition of the temporal property 'now' and the temporal object 'date.' In the same manner as the composition 'red face,' the logical conjunction $now(x) \wedge date(x)$ does not ontologically express the composition 'today.' In order to define the composition, temporal attributes such as 'time unit' and 'time reference' must be employed; these attributes are meta properties of temporal objects and properties.[*1]

[*1] In [21] higher-order concepts are discussed: second-order/third-order properties are considered to be properties of first-order/second-order properties.

In this study, we formalize an upper ontology for the compositions of basic concepts by defining their essential features in logic. We focus on compositions of a property and an object (called ontological compositions) together with the simple compositions of two properties and two objects (called logical conjunctions). Composition ontology is essentially different from the analysis of composite terms in natural languages. Our aim is to deal with complex concepts obtained from basic concepts in an upper ontology, but the procedure does not have to be confined to a particular natural language. By following the upper ontology, we logically formalize and axiomatize the compositions of concepts in monadic second-order logic that is more directly expressible than in first-order logic (cf. [15, 18]). The contributions of this study are listed as follows:

1. *Classification of ontological compositions:* Characterizing, temporal, and spatial compositions are classified on the basis of the combined features of objects and properties, and these compositions are characterized by rigidity and dependency.
2. *Formalization of ontological compositions:* The ontological compositions are axiomatized in monadic second-order logic. Specifically, higher-order concepts (as attributes) that are implicitly and semantically included in the compositions are modeled by using second-order predicates and variables.
3. *Properties of ontological compositions:* The formalization of the composition ontology helps determine the legality and illegality of compositions of properties and objects (e.g., the composition 'gold face' is illegal) and the uniqueness of temporal objects and properties.

This paper is organized as follows. In Section 2, we introduce the basic entities and their compositions in an upper ontology. In Section 3, we explain the characterizing compositions of entities, and in Section 4, we discuss the temporal compositions of entities. Section 5 formalizes and axiomatizes the compositions in monadic second-order logic, and shows some propositions relative to the properties of ontological compositions. In Section 6, we provide a set of RDF triples for our composition ontology by using the RDF(S) description in the Semantic Web. Section 7 discusses how our proposed ontology differentiates from, and complements, existing related work, and how it could be applied in reasoning systems. In Section 8, we present our conclusion and outline our future work.

§2 Classification and Composition of Entities

2.1 Basic Concepts

We classify the five basic concepts as a subset of an upper ontology as follows: attribute, property, object, time, and space. According to the definitions in Welty and Guarino's paper,[25] an entity is called *sortal* if it carries an Identity Condition (IC), and it is called *rigid* if it is essential to all its instances in any time, space, belief, or situation, otherwise it is non-rigid. In this ontology, we deal with the distinction between objects and properties that is defined by the notions of sortality and rigidity.

Object: sortal or pseudo-sortal

Property: characterizing (non-sortal and non-rigid)

The objects and properties are both represented by unary predicates without distinction in predicate logic; for example, $human(x)$ and $tall(x)$. In the ontological classification, the object is defined as sortal or pseudo-sortal, and the property is defined as characterizing (i.e., non-sortal and non-rigid). For example, 'human' is an object (sortal), and 'tall' is a property (non-sortal and non-rigid). Moreover, if a property has an instance, then the instance should correspond to an object. For example, if the object 'human' has an instance 'Tom' and Tom is actually tall, then 'Tom' is suitable as an instance of the property 'tall,' for example, $human(Tom)$ and $tall(Tom)$.

In this paper, we distinguish between attributes and properties in such a manner that attributes are regarded as higher-order concepts for properties and objects. Attributes can be defined as meta predicates, or unary predicates of other unary predicates that represent properties and objects. In other words, attributes correspond to meta properties, or properties of "properties and objects." As properties and objects are interpreted by sets of instances, attributes are interpreted by families of sets of instances. For example, 'color' is an attribute as a higher-order entity whose instances (e.g., 'red' and 'yellow') are properties, and they are represented by $color(red)$ and $color(yellow)$. Also, 'species' is an attribute whose instances (e.g., 'human' and 'dog') are objects, represented by $species(human)$ and $species(dog)$.

Attribute: color, sex, weight

Property: red, yellow, male, heavy, 52kg

From this point of view, attributes are second-order predicates, and properties and objects are first-order predicates. Therefore, an instance of an attribute cannot also be an instance of a property. For example, let *heavy* be an instance of the attribute *weight* and let *red* be a property. Then, $red(heavy)$ contains a

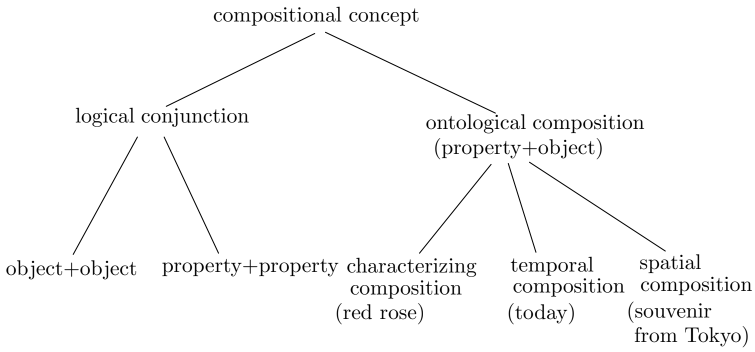

Fig. 1 A composition ontology

type error (rather than being syntactically false).

2.2 Compositions of Objects and Properties

We are concerned with compositions that are generated by combining two basic concepts, such as a property and an object. The composition of two types of concept X and Y is denoted by X+Y; for example, it can be represented by property+object. As two simple compositions of the same entity types, object+object and property+property indicate the composition of two objects and the composition of two properties, respectively. For example, the following represent the concepts 'the fathers who are scientists' and 'the things that are red and long.'

Object+object: father and scientist

Property+property: red and long

These compositions can be simply defined by the logical conjunction (e.g., $human(x) \wedge father(x)$) because they indicate common instances of the same types of concept, i.e., x is an instance of both 'human' and 'father.'

However, we cannot easily handle compositions of different types of concepts by using logical conjunction. This is because the composition of a property and an object (property + object) involves different ontological features that are possessed by the property and the object. As shown in Fig. 1, simple compositions of similar concepts (object+object and property+property) are called logical conjunctions, and the complex compositions of different types of concept (property+object) are called ontological compositions.

Properties are ontologically classified into characterizing properties, temporal properties, and spatial properties. Temporal and spatial properties (e.g., $Sunday(x)$ and $Asia(x)$) are characterizing properties viewed as time and space concepts. On the basis of the nature of the properties, the ontological compositions are subdivided into characterizing compositions, temporal compositions, and spatial compositions. A characterizing composition is constructed by combining a characterizing property and an object. For example, if 'red' is a characterizing property and 'rose' is an object, then 'red rose' is a characterizing composition. A temporal composition is the composition of a temporal property and a temporal or non-temporal object. For example, 'today' is a temporal composition because it consists of the temporal property 'now' and the temporal object 'date.' The property and object indicate a current time point and day intervals, respectively, that is, combining them conceptually expresses the semantics of the entity 'today.' Similarly, the spatial composition is generated by a spatial property and a spatial or non-spatial object. For example, the spatial property 'Tokyo' and the non-spatial object 'souvenir' are combined to form the spatial composition 'souvenir from Tokyo.'

Ontological compositions are useful for creating new concepts compositionally and infinitely. Again, it should be noted that the compositions are syntactically simple, but semantically complex. This is the reason why composition ontology must be carefully designed. The ontological compositions of properties and objects will be exemplified; in particular, the semantics of characterizing compositions and temporal compositions will be discussed ontologically.

§3 Characterizing Compositions

In the field of formal ontology,[11)] time and situation dependencies are proposed as ontological features to classify the following anti-rigid concepts.

Time dependent: child, adult, young, old

Situation dependent: student, teacher, doctor

Time-situation dependent: veteran doctor

Each time-dependent entity is true in a time period and each situation-dependent entity is true in a situation. Moreover, each time-situation dependent entity is true during a specific time period and in a specific situation. The notions of rigidity and dependency will be employed to explain the characterizing compositions.

3.1 Example: red objects

Let us consider the compositions of the property 'red' and the objects 'rose,' 'face,' and 'apple.' According to the notions of formal ontology, the property ‘red’ is anti-rigid, and the objects ‘rose,’ ‘face,’ and ‘apple’ are rigid. On the basis of rigidity and anti-rigidity (such as time and situation dependencies), the compositions ‘red rose,’ ‘red face,’ and ‘red apple’ are ontologically classified as follows.

Red rose: rigid

Red face: situation dependent

Red apple: time dependent

The composition ‘red rose’ is rigid if it is interpreted as a species of the rose. This is because the species cannot be changed at any instant. The composition ‘red face’ is situation dependent if it indicates that someone has blushed or got drunk in a situation. Likewise, the composition ‘red apple’ is time dependent if a green apple has become ripe.

As a trivial case, if a rigid property is combined with a rigid object, then such a composition does not lead to any non-rigidity or dependency. For the composition ‘red rose,’ the interpretation of ‘red’ is limited to the things that are essentially red, that is, the part ‘red’ of ‘red rose’ can be regarded as a rigid property. Thus, the logical conjunction simply defines the rigid composition created by a combination of two rigid concepts.

In contrast, the logical conjunction cannot define the compositions ‘red face’ and ‘red apple’ if they imply blushing faces and ripe red apples, respectively, because of their anti-rigidity. The situation dependency of ‘red face’ originates from the fact that face colors change with situations. This means that faces are not essentially and not always red, but they are red in a particular situation. When someone blushes (or turns pale), the face becomes red (or blue). For interpreting the composition ‘red face,’ the changes in the face colors are importantly limited to several specific colors (e.g., red or blue), i.e., not all colors are accepted for faces. Similarly, the time dependency of ‘red apple’ is explained by the temporal changes in the apple colors that are limited to specific colors of apples. An extended interpretation arises when faces may essentially be ruddy faces or blushing (or drunk) faces and when apples may essentially be red as a species or may turn ripe red during the period of growth.

Red face: essentially ruddy faces and blushing faces

Red apple: a species of apples and ripe red apples

In the extended interpretations, the rigidity and dependency of the compositions ‘red face’ and ‘red apple’ are not uniquely determined.

The property ‘red’ is an instance of the attribute ‘color’ as a higher-order entity. Therefore, the combination of 'red' with the other entity 'face' (resp. 'apple') is constrained by the interpretation of the fused attribute 'face color' (resp. 'apple color') as a restricted (or specific) attribute of 'color.' To discuss such fused attributes, we need to analyze instances of the attributes (e.g., color), some of which are also instances of their restricted attributes (e.g., face color). For example, the color (red, blue, white), shape (circle, box, sphere), quality (rough, hard, soft), age (young, middle, old), and sex (male, female) are attributes that have instances used as properties.

It is important that each attribute has a diversity obtained from the rigidity and dependency of the instances as properties. This diversity indicates the ontological possibility of interpreting the compositions. For example, the diversities of the attributes can be listed as follows:

Color, shape, and quality: rigid, time dependent, and situation dependent

Age: time dependent

Sex: rigid

The diversity of the attribute 'color' is the set of three types of colors: essential colors (rigid), colors exhibited during periods of growth (time dependent), and colors exhibited in particular situations (situation dependent). In the first case, the essential colors are colors by nature. In the second case, colors change during the growth of organic entities. In the third case, a situation induces a color change in an object. Unlike 'color,' the diversities of the attributes 'age' and 'sex' are deterministically time dependent and rigid, respectively.

On the basis of the above description, the diversities of the fused attributes 'face color,' 'apple color,' and 'eye color' are listed as follows:

Face color: rigid or situation dependent

Apple color: rigid or time dependent

Eye color: rigid

These diversities are subsets of the diversity of the attribute 'color' such that each color is rigid, time dependent, or situation dependent. For the composition 'red face,' the diversity of the fused attribute 'face color' is rigid or situation dependent. For the composition 'red apple,' the diversity of the fused attribute 'apple color' is rigid or time dependent. Therefore, 'red face' is rigid or situation dependent and 'red apple' is rigid or time dependent. As another example, the eye colors of humans are essentially considered to be rigid; therefore, the composition 'black eye' is rigid.

The diversity imposes a semantic constraint such that a property in-

attributes	subattributes	diversities	instances
color	face color	rigid	white, black, yellow
		situation dependent	red, blue, white
	apple color	rigid	green, red
		time dependent	green, yellow green, red, dark red
	eye color	rigid	black, brown, blue, green

Table 1 Diversities and instances of attributes

cluded in the composition is legal if it is an instance of the fused attribute. Intuitively speaking, 'red' in 'red face' has to belong to the fused attribute 'face color,' and therefore, it is impossible to fuse every color instance with the object 'face.' Table 1 shows the diversities and instances of the fused attributes. The attributes are subattributes of the attribute 'color.' Hence, each set of instances is limited to some of the colors.

In the above examples, rigid objects are fused with properties. We consider a characterizing property fused with an anti-rigid object, such as the composition 'red teacher' of the characterizing property 'red' and the anti-rigid object 'teacher.' To interpret such a composition, the time and situation dependencies of the objects are analyzed as follows.

Teacher: situation dependent

Novice teacher: time-situation dependent

If such different types of objects are combined with 'red' to be situation dependent, then the compositions have the following dependency combinations.

Red teacher: situation dependent + situation dependent

Red novice teacher: time-situation dependent + situation dependent

The composition 'red teacher' involves double situation dependency, i.e., someone is a red teacher when she/he is a teacher (working in a school) and becomes red in a situation (ashamed or drunk). The composition 'red novice teacher' has time-situation dependency and situation dependency by combining 'red' and 'novice teacher' because 'novice teacher' is true for a particular time period in a situation.

§4 Temporal Compositions

In this section, we discuss the compositions of temporal properties and temporal objects (temporal compositions) corresponding to temporal complex concepts.

4.1 Example: today = now + date

We consider the composition ‘today’ which represents the day including the current time (i.e., now). In a timeline of days, the composition ‘today’ is temporally related to the two temporal compositions ‘yesterday’ and ‘tomorrow.’ In other words, the three compositions ‘yesterday,’ ‘today,’ and ‘tomorrow’ are similar to the temporal properties ‘last time,’ ‘now,’ and ‘next time,’ respectively. Because a time point mapping to now exists in the range of each time unit, ‘now,’ ‘today,’ and ‘this week’ are semantically related to each other. To make it explicit, we employ the specific property ‘now during a time interval’ where there are infinitely many time intervals including the time point ‘now.’ For instance, ‘now during today’ and ‘now in this week’ refer to the same time point, but they are different time intervals. By considering these time intervals, the temporal properties ‘last time,’ ‘now,’ and ‘next time’ are extended to the temporal compositions ‘yesterday,’ ‘today,’ and ‘tomorrow,’ and they are further extended to the temporal compositions ‘last week,’ ‘this week,’ and ‘next week,’ respectively. The modeling of these compositions is performed on the basis of various granularities of time units.

A time ontology (cf. [1)]) defines time concepts and units such as time instants, time intervals, hours, days, weeks, months, and years. It provides a vocabulary for expressing and specifying temporal concepts. The standard time ontology contains the basic temporal concepts but does not semantically define the compositions of temporal properties and objects as a combination of different concepts such as ‘today,’ ‘this week,’ and ‘this month.’

In our ontology, we define the compositions ‘yesterday,’ ‘today,’ and ‘tomorrow’ over day intervals, where ‘today’ is mapped into a day interval containing ‘now’ and ‘yesterday’ and ‘tomorrow’ are mapped into the last and next day intervals, respectively. Let us introduce the following time intervals.

> Time intervals: second intervals, minutes intervals, hour intervals, day intervals, week intervals, month intervals, and year intervals

By using these time intervals, the temporal compositions ‘today,’ ‘this week,’ ‘this month,’ and ‘this year’ are defined as follows.

Today ⇔ now + day intervals

This week ⇔ now + week intervals

This month ⇔ now + month intervals

attributes	concepts	instances
property of characterizing property	color	red,blue,yellow
property of temporal property	time reference	now, last time, next time
property of spatial property	space reference	here, right-side space
property of non-spatio-temporal object	species	human, dog, bird
property of temporal object	time unit	second, minute, hour
property of spatial object	space area	town, city, country

Table 2 Attributes as predicates of predicates

This year $\Leftrightarrow$ now + year intervals

Likewise, the temporal compositions 'yesterday' and 'tomorrow' are defined as follows.

Yesterday $\Leftrightarrow$ last time + day intervals

Tomorrow $\Leftrightarrow$ next time + day intervals

Because the composition 'today' is defined by 'now' and 'day intervals,' removing 'now' from 'today' leaves day intervals in its remaining part. In other words, the day intervals do not focus on a specific day interval, but the present time point of 'now' selects a specific day from the day intervals. For example, if the current day is November 15, 2007, then 'today' is equivalent to 'now' in a day that is selected from a set of dates in a calendar.

Today $\Leftrightarrow$ now + { ..., (2007-12-14),(2007-12-15),(2007-12-16),... }

Similar to the temporal compositions, we can consider the spatial compositions that consist of spatial properties and objects. For example, 'this city' is the composition of the spatial property 'here' (denoting the place where you are) and the spatial object 'city.' Further, 'neighboring city' is the composition of the spatial property 'neighboring' and the spatial object 'city.' However, unlike the temporal compositions, instances of the neighboring cities cannot be uniquely determined because space has two dimensions.

§5 Compositions in Logical Formalization

On the basis of the previous sections, we will establish the logical formalization of characterizing compositions and temporal compositions for the upper ontology.

In order to formalize our ontological compositions, we introduce the notion of monadic second-order logic.[23, 16] Unlike general second-order logic, monadic second-order logic guarantees the sound and complete deduction. The alphabet of a second-order language consists of first-order/second-order variables; first-order/second-order predicates; first-order functions; logical connectives $\wedge$, $\vee$, $\neg$, $\Rightarrow$, and $\Leftrightarrow$; and quantifiers $\forall$ and $\exists$. We express $V_1 = \{x, y, z, \ldots\}$ for a set of first-order variables and $V_2 = \{X, Y, Z, \ldots\}$ for a set of second-order variables. The objects, properties, and compositions correspond to first-order unary predicates that are constants of second-order variables. Attributes (regarded as predicates of predicates) correspond to second-order unary predicates. For convenience, $Prop$, Obj, $Comp$, and Att denote a set of properties, a set of objects, a set of compositions, and a set of attributes, respectively.

As shown in Table 2, the attributes are used to define the meta properties of properties and objects. The properties of properties are divided into properties of non-temporal properties and properties of temporal properties. For example, the attribute 'color' is a property of non-temporal properties, and it has the instances 'red,' 'blue,' and 'yellow,' (which are defined as physical qualities in DOLCE[17]). The attribute 'time reference' is a property of temporal properties, and it has the instances 'now,' 'last time,' and 'next time.' Furthermore, the properties of objects are classified into properties of non-temporal objects and properties of temporal objects. For example, the attribute 'species' is a property of non-temporal objects, and it has the instances 'human,' 'dog,' and 'bird.' The attribute 'time unit' is a property of temporal objects, and it has the instances 'second,' 'minute,' and 'hour.'

5.1 Formalization of Characterizing Compositions

We axiomatize characterizing compositions using second-order predicates and variables. Let $p \in Prop$, $o \in Obj$, $po \in Comp$, $x \in V_1$, and $a_o \in Att$. The composition of a property p and an object o, denoted by po, is defined as a first-order predicate.

Property+object: $\forall x.(po(x) \Leftrightarrow p(x) \wedge o(x) \wedge a_o(p))$

The composition po implies the set of instances x such that x is an instance of po if and only if x is an instance of a property p and an object o, and p is an instance of an attribute a_o. The attribute a_o is a set of properties restricted by the object o. Let $a \in Att$ be an attribute as a property of non-temporal properties, i.e., instances of a are non-temporal properties. The restricted attribute a_o of object o is defined as follows.

Restricted attribute: $\forall X.(a_o(X) \Leftrightarrow a(X) \wedge \exists x.(X(x) \wedge o(x)))$

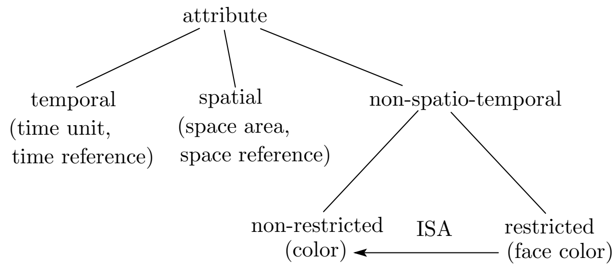

Fig. 2 A classification of non-spatio-temporal attributes

This axiom indicates that a_o is subsumed by an attribute a, and there exists x such that x is an instance of both X and o where the second-order variable X represents a variable over properties. An attribute is restricted if there is an object that restricts the attribute; otherwise, it is non-restricted. Fig.2 shows a classification of non-spatio-temporal attributes on the basis of restrictedness. For example, 'face color' and 'species in the sea' (denoted by $color_{face}$ and $species_{sea}$) are restricted and are subattributes of 'color' and 'species.'

A composition *po* is illegal if it has no instance x in the above axioms. Therefore, if *po* is illegal, then the property p and the object o are not fused as a complex concept. For example, the composition *gold_face* is illegal because the property *gold* is not an instance of the attribute $color_{face}$ restricted by the object *face*. Our axioms lead to the following proposition that recognizes a set of illegal compositions.

Proposition 5.1 (Illegal Compositions)
Let $p \in Prop$, $o \in Obj$, and $po \in Comp$. If there is no restricted attribute a_o of object o such that the property p is an instance of a_o (i.e., $a_o(p)$), then the composition *po* is illegal.

This proposition implies that if any attribute restricted by the object o does not have a property p, the composition *po* is empty. Let $color, color_{face} \in Att$, $red \in Prop$, $face \in Obj$, and $red\_face \in Comp$. Then, an example of the characterizing composition *red_face* of the property 'red' and the object 'face' is as follows.

$$\forall x.(red\_face(x) \Leftrightarrow red(x) \wedge face(x) \wedge color_{face}(red))$$

The composition *red_face* is defined by the set of instances x such that x is an instance of the property *red* and of the object *face*, and *red* is an instance of the attribute $color_{face}$ of the object *face*.

$color_{face}(red) \Leftrightarrow color(red) \wedge \exists x.(red(x) \wedge face(x))$

The property *red* is an instance of the attribute $color_{face}$ if and only if *red* is an instance of the attribute *color*, and there exists x such that x is an instance of both the property *red* and the object *face*. As discussed in Section 3, the fused attribute 'face color' denoted by $color_{face}$ has instances suitable for faces that are a subset of the attribute *color*.

Furthermore, we define the rigidity and dependency of the characterizing compositions (based on [24)]). Let t, t', t'' be first-order variables with regard to time and s, s', s'' be first-order variables with regard to situation. Let ϕ be a first-order unary predicate that denotes an object, property, or composition. Then, the unary predicate $\phi(x)$ is extended to the following binary and ternary predicates in order to state the truth of $\phi(x)$ at a given time or in a given situation.

$\phi(x, t) \Leftrightarrow x$ is an instance of ϕ at time t

$\phi(x, s) \Leftrightarrow x$ is an instance of ϕ in situation s

$\phi(x, t, s) \Leftrightarrow x$ is an instance of ϕ at time t in situation s

By the extended predicates, the rigidity of the composition *po* is defined as follows.

Rigid composition: $\forall x.(po(x) \Leftrightarrow \forall t, s.(o(x, t, s) \Rightarrow po(x, t, s)))$

This axiom indicates that for every situation s and for every time t, if x is an instance of object o at time t in situation s, then x is an instance of composition *po* at time t in situation s. Moreover, three types of dependencies of the composition *po* are defined as follows.

Time dependent composition:

$$\forall x.(po(x) \Leftrightarrow \exists t_1, t_2.(t_1 < t_2 \wedge \forall t.(t_1 \leq t \leq t_2 \Rightarrow po(x, t)) \wedge \exists t'.((t' < t_1 \vee t_2 < t') \wedge \neg po(x, t'))))$$

Situation dependent composition:

$$\forall x.(po(x) \Leftrightarrow \exists s, s'.(s \neq s' \wedge po(x, s) \wedge \neg po(x, s')))$$

Time-situation dependent composition:

$$\forall x.(po(x) \Leftrightarrow \exists t_1, t_2.(t_1 < t_2 \wedge \exists s, s' \forall t.(t_1 \leq t \leq t_2 \Rightarrow (s \neq s' \wedge po(x, t, s) \wedge \neg po(x, t, s'))) \wedge \exists t'.((t' < t_1 \vee t_2 < t') \wedge \neg po(x, t'))))$$

In the relation $t < t'$, the time instances t and t' denote the beginning and end of a time period, respectively.

By restricting 'color,' the restricted attribute 'face color' becomes rigid or situation dependent. Therefore, if the composition 'red face' is rigid, then it is defined as follows:

$$\forall x.(red\_face(x) \Leftrightarrow \forall t, s.(face(x,t,s) \Rightarrow red\_face(x,t,s)))$$

If the composition 'red face' is situation dependent, then it is defined as follows:

$$\forall x.(red\_face(x) \Leftrightarrow \exists s.red\_face(x,s) \wedge \exists s'.\neg red\_face(x,s'))$$

Unlike 'face color,' the restricted attribute 'apple color' becomes rigid or time dependent. Therefore, if the composition 'red apple' is rigid, then it is defined as follows:

$$\forall x.(red\_apple(x) \Leftrightarrow \forall t, s.(apple(x,t,s) \Rightarrow red\_apple(x,t,s)))$$

If the composition 'red apple' is time dependent, then it is defined as follows:

$$\forall x.(red\_apple(x) \Leftrightarrow \exists t_1, t_2.(t_1 < t_2 \wedge \forall t.(t_1 \leq t \leq t_2 \Rightarrow red\_apple(x,t)) \wedge \exists t'.((t' < t_1 \vee t_2 < t') \wedge \neg red\_apple(x,t'))))$$

Finally, the double situation dependency and combinations of time and situation dependencies are simply formalized by the rigidity and dependency of each object o and property p in the composition po. Additionally, the following axiom is supplemented to characterize the composition $po(x)$ to be a subset of $o(x) \wedge p(x)$.

$$\forall x.(po(x) \Leftrightarrow \forall t, s.(po(x,t,s) \Rightarrow o(x,t,s) \wedge p(x,t,s)))$$

This axiom illustrates that each instance of the composition po is also an instance of both object o and property p at every time and in every situation. It can define the composition of a situation dependent object and a situation dependent property such as 'red teacher.'

$$\forall x.(red\_teacher(x) \Leftrightarrow \forall t, s.(red\_teacher(x,t,s) \Rightarrow teacher(x,t,s) \wedge red(x,t,s)))$$

That it, every red teacher is an instance of the object *teacher* and the property *red* such that both *teacher* and *red* are situation dependent. This double situation dependency is defined as follows.

$$\forall x.(teacher(x) \Leftrightarrow \exists s, s'.(s \neq s' \wedge teacher(x,s) \wedge \neg teacher(x,s')))$$
$$\forall x.(red(x) \Leftrightarrow \exists s, s'.(s \neq s' \wedge red(x,s) \wedge \neg red(x,s')))$$

5.2 Formalization of Temporal Compositions

We define temporal compositions on the basis of some of the temporal notions in a calendar logic.[20, 6] We introduce the two attributes $time\_reference$ and $time\_unit$ (in Att) that are properties of temporal properties and of temporal objects. Let $now \in Prop$ be a temporal property (as an instance of the attribute $time\_reference$), and let $second$, $minute$, $hour$, day, $week$, $month$, $year \in Obj$ be temporal objects (as instances of the attribute $time\_unit$).

In our formalization, all the instances of temporal objects are represented by a totally ordered set of non-negative integers. We use the temporal object *second* as the smallest time unit, and similar to [3], we set the origin 0 as January 1 0:00:00 1970. Let n, i_b, i_e be first-order variables with respect to integers. Time intervals in seconds define various time units (as instances of the attribute *time_unit*). For example, the temporal object *day* is defined by the ternary predicate *day_interval*.

$\forall n.(day(n) \Leftrightarrow \exists i_b, i_e.day\_interval(n, i_b, i_e))$

In the following, the temporal object 'day interval' is defined as a ternary predicate $day\_interval(n, i_b, i_e)$, where n is an ordinal number in day intervals, i_b is the beginning of the interval, and i_e is the end of the interval.

$day\_interval(0, 0, 86400)$

$\forall n, i_b, i_e.(day\_interval(n+1, i_e, i_e + 86400) \Leftrightarrow day\_interval(n, i_b, i_e))$

The ordinal number 0 indicates the day interval from second 0 (included) up to second 86400 (excluded), and then the other ordinal numbers greater than zero (i.e., $n > 0$) are defined recursively. Further, the temporal objects *minute*, *hour*, *week*, *month*, and *year* and their intervals are defined in terms of seconds.

We require an expression such that x is a day within a specific month y and y is a month within a specific year z. For two time units X and Y, the binary predicate $X\_within\_Y(x, y)$ (as a variant of a similar concept in [20]) is defined with respect to non-negative integers.

$\forall x, y.(X\_within\_Y(x, y) \Leftrightarrow \exists i_b, i_e, i'_b, i'_e.(X\_interval(x, i_b, i_e) \wedge$
$Y\_interval(y, i'_b, i'_e) \wedge (i'_b \leq i_b) \wedge (i_e \leq i'_e)))$

For example, the binary predicate $hour\_within\_day(x, y)$ is defined as follows.

$\forall x, y.(hour\_within\_day(x, y) \Leftrightarrow \exists i_b, i_e, i'_b, i'_e.(hour\_interval(x, i_b, i_e) \wedge$
$day\_interval(y, i'_b, i'_e) \wedge (i'_b \leq i_b) \wedge (i_e \leq i'_e)))$

Let *yesterday*, *today*, *tomorrow* $\in Comp$ be temporal compositions. We can define the temporal composition *today* by using the temporal property *now* and the binary predicate X_within_Y, as follows.

$\forall x.(today(x) \Leftrightarrow \exists_{=1} u, v, w.(now(u) \wedge second\_within\_minute(u, v) \wedge$
$minute\_within\_hour(v, w) \wedge hour\_within\_day(w, x) \wedge day(x)))$

This means that x is an instance of *today* if and only if there exists exactly one constant of u, v, and w such that u is an instance of *now* and a second within a minute v, v is a minute within an hour w, and w is an hour within a day x. The counting quantifier $\exists_{=n}$ indicates that there are exactly n values. In addition, the temporal compositions *yesterday* and *tomorrow* are defined as the last time and the next time of today.

$\forall x.(yesterday(x) \Leftrightarrow today(x-1))$
$\forall x.(tomorrow(x) \Leftrightarrow today(x+1))$

In a similar manner, the temporal compositions $last\_week$, $next\_week$, $last\_month$, $next\_month$, $last\_year$, and $next\_year$ are defined as the predecessor and successor of the temporal compositions $this\_week$, $this\_month$, and $this\_year$, respectively.

Proposition 5.2 (Time Units)
Let $p \in Prop$ be an instance of the attribute $time\_unit$. Then, there are (finitely or infinitely) many instances of p.

This proposition implies that each time unit represents a set of time intervals for a time granularity. For example, since the temporal object *day* is an instance of the attribute $time\_unit$, it has (infinitely) many time instances. The temporal compositions defined in the above axioms are added as new properties to the instances of the attribute $time\_reference$ because the following uniqueness holds.

Proposition 5.3 (Time References)
Let $\phi \in Prop \cup Comp$ be an instance of the attribute $time\_reference$. Then, there is a unique instance of ϕ.

This proposition indicates that the instances of concepts referring to time are uniquely decided even if they change with time. For example, the temporal composition *today* has a unique instance because *today* is a concept referring to a specific day including 'now.' Similarly, at any time, there is only one instance of 'yesterday' and one instance of 'tomorrow.' The uniqueness of temporal attributes is defined by the following. A property is unique if there is a unique instance of the property in a model; otherwise, it is non-unique. As shown in Fig.3, temporal attributes are classified into unique (time reference) and non-unique (time unit) attributes.

§6 Compositional Concepts in RDF(S)

RDF (Resource Description Framework)[13] is a family of World Wide Web Consortium (W3C) specifications, used as a knowledge-representation language for conceptual description or modeling of information in the Web. RDFS (RDF Schema)[4] is a semantic extension of RDF, used for representing simple RDF vocabularies. For the purpose of the Semantic Web, we provide the

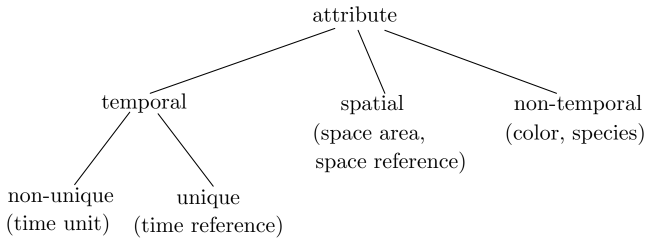

Fig. 3 A classification of temporal and spatial attributes

RDF(S) description of our composition ontology by using the metamodeling of concepts in RDF triples (and it results in a translation into Conceptual Graphs, as discussed in [2)]). A triple of the form $\langle s, rdf{:}type, o\rangle$ indicates that s is an instance of class o, and a triple of the form $\langle s, rdfs{:}subClassOf, o\rangle$ states that s is a subclass of o.

We reduce our definition of the attributes and compositional concepts to a set of RDF triples. Let the attribute classes *uo:non_temporal_attribute*, *uo:attribute*, *uo:non_temporal_attribute* (where *uo*: (standing for 'upper ontology') is a new name space used to introduce our proposed vocabularies) be instances of *rdfs:Class*, and let the classes *red*, *white*, *green* and the meta-classes *color*, *face_color*, *color*, *apple_color* be instances of *rdfs:Class*. An upper ontology of attributes and compositional concepts is constructed by the following RDF triples (as described in Fig 4):

$\langle uo{:}non\_temporal\_attribute, rdfs{:}subClassOf, uo{:}attribute\rangle$
$\langle color, rdf{:}type, uo{:}non\_temporal\_attribute\rangle$
$\langle face\_color, rdfs{:}subClassOf, color\rangle$
$\langle apple\_color, rdfs{:}subClassOf, color\rangle$
$\langle eye\_color, rdfs{:}subClassOf, color\rangle$
$\langle red, rdf{:}type, face\_color\rangle$
$\langle white, rdf{:}type, face\_color\rangle$
$\langle red, rdf{:}type, apple\_color\rangle$
$\langle green, rdf{:}type, apple\_color\rangle$
$\langle black, rdf{:}type, eye\_color\rangle$
$\langle brown, rdf{:}type, eye\_color\rangle$

In the ontology, a class is an instance of a meta-class, e.g., the class *red* is an instance of the meta-classes *apple_color* and *face_color*. In addition, the attributes *face_color*, *apple_color*, and *eye_color* are defined as subclasses of *color*, which in turn is an instance of *uo:non_temporal_attribute*.

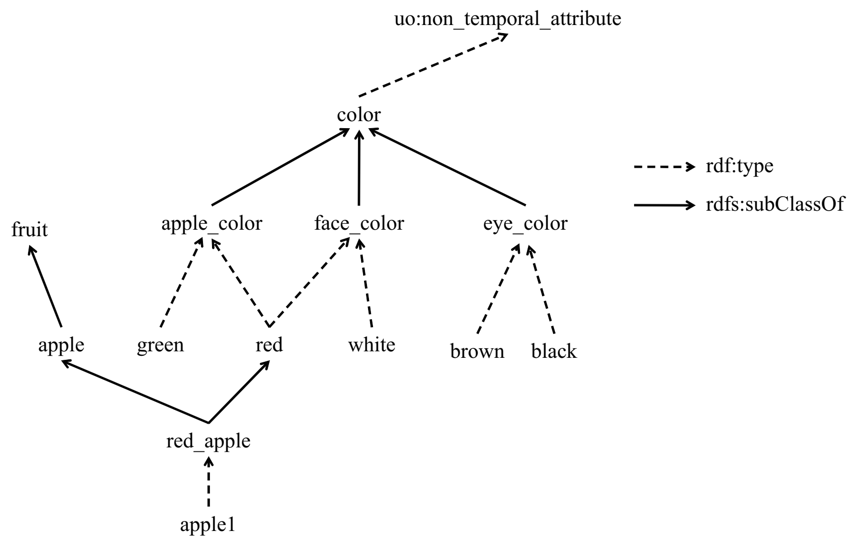

Fig. 4 Compositional Concepts in RDF(S)

Let the resource $apple1$ be an instance of $rdfs{:}Resource$ and let the classes red_apple, $apple$, $fruit$ be instances of $rdfs{:}Class$. Then, the compositional concept red_apple is well typed by the following RDF triples.

$\langle apple, rdfs{:}subClassOf, fruit\rangle$
$\langle red\_apple, rdfs{:}subClassOf, apple\rangle$
$\langle red\_apple, rdfs{:}subClassOf, red\rangle$
$\langle apple1, rdf{:}type, red\_apple\rangle$

We can identify the resource $apple1$ that is an instance of the compositional concept red_apple. In our ontological classification, compositional concepts are defined as subclasses of the two classes of properties and objects. For example, the compositional concept red_apple is a subclass of the two classes red (as a property) and $apple$ (as an object) where red is an instance of the attribute $apple\_color$.

As shown in Fig 5, the attributes $apple\_color$, $face\_color$, and eye_color are classified into rigid, temporal, and situational classes in the following RDF triples.

$\langle rigid\_apple\_color, rdfs{:}subClassOf, apple\_color\rangle$
$\langle temporal\_apple\_color, rdfs{:}subClassOf, apple\_color\rangle$
$\langle rigid\_face\_color, rdfs{:}subClassOf, face\_color\rangle$
$\langle situational\_face\_color, rdfs{:}subClassOf, face\_color\rangle$

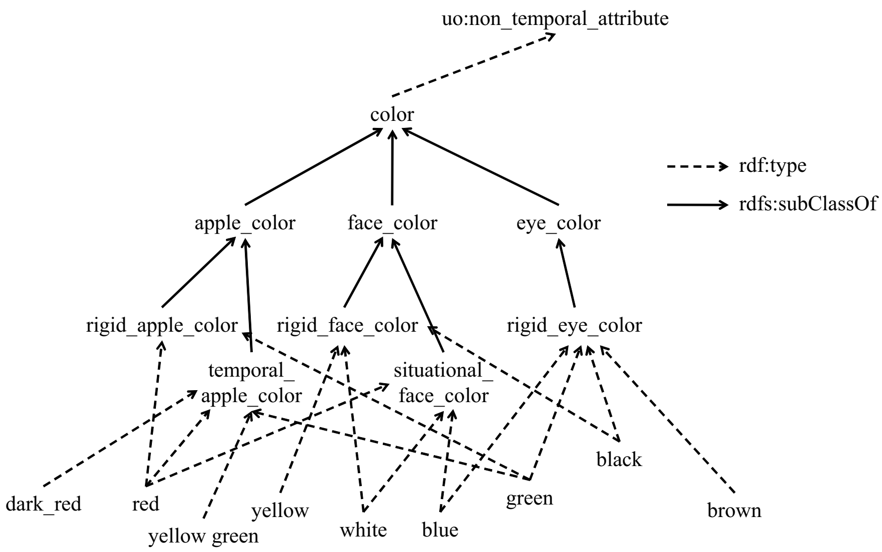

Fig. 5 Rigidity of Compositional Concepts in RDF(S)

$\langle \mathit{rigid\_eye\_color}, \mathit{rdfs}{:}\mathit{subClassOf}, \mathit{eye\_color} \rangle$

$\langle \mathit{red}, \mathit{rdf}{:}\mathit{type}, \mathit{rigid\_apple\_color} \rangle$

$\langle \mathit{red}, \mathit{rdf}{:}\mathit{type}, \mathit{temporal\_apple\_color} \rangle$

$\langle \mathit{red}, \mathit{rdf}{:}\mathit{type}, \mathit{situational\_face\_color} \rangle$

$\langle \mathit{blue}, \mathit{rdf}{:}\mathit{type}, \mathit{situational\_face\_color} \rangle$

$\langle \mathit{blue}, \mathit{rdf}{:}\mathit{type}, \mathit{rigid\_eye\_color} \rangle$

In the ontology, the compositional concept *blue_apple* is illegal but the compositional concepts $blue\_face$ and *blue_eye* are not. This is because the class *blue* is an instance of the attributes $situational\_face\_color$ and $rigid\_eye\_color$ but not an instance of any subclass of the attribute *apple_color*.

As shown in Fig 6, we use a set of RDF triples to ontologically define temporal attributes such as time units and time references. Let the attribute classes *uo*:*temporal_attribute*, *uo*:*time_unit*, *uo*:*time_reference*, *uo*:*within* be instances of *rdfs*:*Class*. The following RDF triples contain many instances of the attributes *uo*:*time_unit* and *uo*:*time_reference*.

$\langle \mathit{uo}{:}\mathit{temporal\_attribute}, \mathit{rdfs}{:}\mathit{subClassOf}, \mathit{uo}{:}\mathit{attribute} \rangle$

$\langle \mathit{uo}{:}\mathit{time\_reference}, \mathit{rdfs}{:}\mathit{subClassOf}, \mathit{uo}{:}\mathit{temporal\_attribute} \rangle$

$\langle \mathit{second}, \mathit{rdf}{:}\mathit{type}, \mathit{uo}{:}\mathit{time\_unit} \rangle$

$\langle \mathit{minute}, \mathit{rdf}{:}\mathit{type}, \mathit{uo}{:}\mathit{time\_unit} \rangle$

$\langle \mathit{hour}, \mathit{rdf}{:}\mathit{type}, \mathit{uo}{:}\mathit{time\_unit} \rangle$

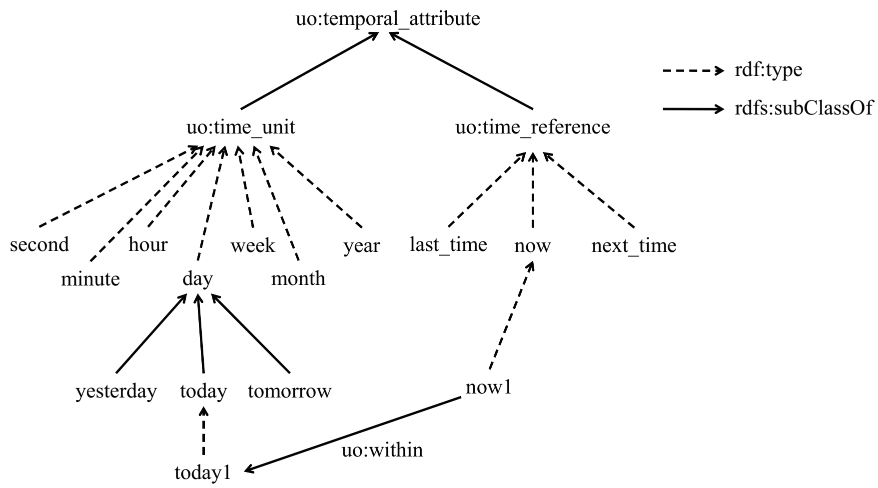

Fig. 6 Temporal Attributes in RDF(S)

$\langle day, rdf{:}type, uo{:}time\_unit\rangle$
$\langle week, rdf{:}type, uo{:}time\_unit\rangle$
$\langle month, rdf{:}type, uo{:}time\_unit\rangle$
$\langle year, rdf{:}type, uo{:}time\_unit\rangle$
$\langle now, rdf{:}type, uo{:}time\_reference\rangle$
$\langle last\_time, rdf{:}type, uo{:}time\_reference\rangle$
$\langle next\_time, rdf{:}type, uo{:}time\_reference\rangle$

The class *day* is an instance of meta-class *uo*:*time_unit*, and the class *now* is an instance of meta-class *uo*:*time_reference*.

Moreover, the classes *yesterday*, *today*, and *tomorrow* are defined by the classes *day* and *now*. Let $today1$ be an instance of *rdfs*:*Resource* and let *yesterday*, *today*, *tomorrow* be instances of *rdfs*:*Class*.

$\langle today1, rdf{:}type, today\rangle$
$\langle yesterday, rdfs{:}subClassOf, day\rangle$
$\langle today, rdfs{:}subClassOf, day\rangle$
$\langle tomorrow, rdfs{:}subClassOf, day\rangle$
$\langle now1, rdf{:}type, now\rangle$
$\langle now1, uo{:}within, today1\rangle$

In the RDF triples, the compositional concept *today* is well defined by that it is a subclass of *day* and its instance *today1* is related to an instance of the class *now*, i.e., *now1* is time within *today1*.

§7 Discussion

7.1 Related Work

Many upper ontologies have been proposed in the area of ontology in information systems. Guarino et al. have formalized the upper ontology DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering),[17] in which event is a subclass of perdurant, disjoint to endurant, quality, and abstract. DOLCE can be partially translated into OWL (Web Ontology language) axioms because it is defined in first-order logic. Furthermore, BFO (Basic Formal Ontology)[7] is a formal ontological framework that consists of two categories of sub-ontologies SNAP and SPAN. SNAP (for snapshot) consists of ontologies for endurant entities (also called continuants, such as 3D objects at a given time), while SPAN (for spanning-time) encompasses ontologies for perdurant entities (also called occurrents, such as processes that span time, thus enabling formalization of 4D entities).

The above theories and other existing upper ontological frameworks have been well formalized by ontology researchers over a long time. However, they do not address compositional concepts using predicate variables in second-order logic, as in our logical and ontological framework. Our formalization also provides a way to verify legal and illegal compositions, which cannot be performed with first-order logic.

Modeling meta-concepts (i.e., classes of concepts) to express the meanings of upper-level data and vocabularies (e.g., species, colors, and product types) in OWL ontologies is supported by OWL-Full (the most expressive language of OWL). The semantics of modeling meta-concepts and the undecidability of meta-modeling in OWL-Full have been discussed in [19]. HILOG[5] uses second-order expressions but it is an undecidable higher-order language for logic programming. However, *monadic* second-order logic is decidable in many formalizations, because the quantification of predicate variables is limited to subsets of the universe (i.e., monadic).

In addition to the undecidability, the related logical approaches do not embody any method to define and verify the compositional concepts based on the upper ontologies. The important point is that our approach is a new combination of the upper ontology and the logical language for meta concepts. To the best of our knowledge, there are no other approaches to treat the compositional concepts in second-order logic.

7.2 Applications of the Formalization of the Compositions

One of the applications of ontology is to make the semantics of many concepts machine-readable in order to treat conceptual data across multiple knowledge resources. The formalization of the compositions helps to precisely model the complex concepts corresponding to the defined compositions. The compositional concepts 'red face,' 'yesterday,' 'today,' etc. often appear in texts but it is difficult to handle their semantics in information systems, even by using ISA and PART-OF relations. Thus, the axioms of the characterizing, temporal, and spatial compositions have to be added to the conceptual modeling in the information systems.

The standard relational database language SQL has data constraints in tables, e.g., by datatypes, UNIQUE, and REFERENCES in CREATE TABLE. Unlike SQL, we can apply the upper ontology of the RDF triples in Section 6 to a RDF data store system. An RDF fact data set can be well modeled and maintained by the semantic data constraints under the ontology. Consider the following instance data in the RDF form that indicates that Tom's face is essentially red.

$\langle st1, \mathit{rdf}{:}\mathit{type}, \mathit{uo}{:}\mathit{rigid}\rangle$
$\langle st1, \mathit{rdf}{:}\mathit{type}, \mathit{rdf}{:}\mathit{statement}\rangle$
$\langle st1, \mathit{rdf}{:}\mathit{subject}, \mathit{face1}\rangle$
$\langle st1, \mathit{rdf}{:}\mathit{predicate}, \mathit{rdf}{:}\mathit{type}\rangle$
$\langle st1, \mathit{rdf}{:}\mathit{object}, \mathit{red}\rangle$
$\langle \mathit{Tom}, \mathit{hasPart}, \mathit{face1}\rangle$

The rigidity of Tom's face color is expressed by that the statement $st1$ is an instance of $\mathit{uo}{:}\mathit{rigid}$. This is a constraint in the data store system such that we cannot update his face color data without deleting the rigid statement. By replacing the first triple with the following, the update of his face color data is available in the data store system.

$\langle st1, \mathit{rdf}{:}\mathit{type}, \mathit{uo}{:}\mathit{temporal\_dependent}\rangle$

Instead of the update, we may add several face colors with timestamps to the data store system. In addition, the following instance data cannot be inputed in the data store system because the blue color is illegal as a rigid face color.

$\langle st2, \mathit{rdf}{:}\mathit{type}, \mathit{uo}{:}\mathit{rigid}\rangle$
$\langle st2, \mathit{rdf}{:}\mathit{type}, \mathit{rdf}{:}\mathit{statement}\rangle$
$\langle st2, \mathit{rdf}{:}\mathit{subject}, \mathit{face2}\rangle$

$\langle st2, rdf{:}predicate, rdf{:}type \rangle$
$\langle st2, rdf{:}object, blue \rangle$
$\langle John, hasPart, face2 \rangle$

It is useful for us to ask the query "What is essentially (or temporally) red?" for finding some Web pages including the word 'red' with the RDF annotation data.

Furthermore, we can represent a causal relation in knowledge bases when the cause event occurred yesterday and the effect event occurs today, e.g., $occurs(e_1, yesterday)$ causes $occurs(e_2, today)$, where e_1 and e_2 are events. The cause and effect of a causal relation are temporally constrained by the fact that the cause event occurs before the effect event. However, computers can read time stamps but not these abstract concepts ('yesterday' and 'today') implying time points and day intervals. If the temporal compositions are formally defined by axioms, then they are well embedded in knowledge representation and reasoning on causal relations.

In order to use the abstract time concepts such as 'yesterday' and 'today' in a knowledge base, *current_time_stamp* in the following triple has to be automatically updated in the RDF data store system.

$\langle current\_time\_stamp, rdf{:}type, uo{:}now \rangle$

As a result of this, the RDF triples in Section 6 provide real instance data of the abstract time concepts such as 'yesterday' and 'today.' For example, we can make the following SPARQL (SPARQL Protocol and RDF Query Language) query:

```
SELECT ?date
WHERE
{
   ?today1 rdf:type uo:today .
   ?now1 rdf:type uo:now .
   ?today1 uo:within ?now1 .
   BIND (day(?now1) AS ?date)
}
```

in order to ask the question "What is today's date?" to the knowledge base.

§8 Conclusion and Future Work

We presented an upper ontology for compositions of objects and properties. To formalize the ontology, we axiomatized the nature of the compositions in monadic second-order logic. Unlike logical conjunctions, the ontological compositions have to be characterized by attributes as higher-order concepts. When combining an object and a property, a restricted attribute (e.g., 'face color') is used to exclude illegal concepts in the compositions (e.g., 'gold' in 'gold face' is illegal). Moreover, the compositions of temporal properties and objects (e.g., *today* is the composition of *now* and *day_interval*) are well defined by using the temporal attributes *time_unit* and *time_reference*. The ontology-based formalization provides a classification of (non-temporal and temporal) attributes with respect to restrictedness and uniqueness.

There remains work to formalize the spatial compositions classified in the composition ontology, in which mereotopological axioms (e.g., concerning PART-OF relations) should be reasonably considered. The formalization of such compositions would enable us to develop a logical reasoning system for the fused concepts involving spatial information.

References

1) http://www.w3.org/tr/owl-time.

2) J. Baget, M. Croitoru, A. Gutierrez, M. Leclère, and M.-L. Mugnier. Translations between rdf(s) and conceptual graphs. In *Proceedings of the 18th International Conference on Conceptual Structures (ICCS 2010)*, LNCS 6208, pages 28–41. Springer, 2010.

3) C. Bettini, S. G. Jajodia, and S. X. Wang. *Time Granularities in Databases, Data Mining and Temporal Reasoning.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2000.

4) D. Brickley and R. V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation, http://www.w3.org/tr/2004/rec-rdf-schema-20040210/. Technical report, 2004.

5) W. Chen and M. Kifer. Sorted HiLog: Sorts in higher-order logic data languages. In *Proc. of the 5th International Conference on Database Theory (ICDT'95)*, LNCS 893, pages 252–265. Springer, 1995.

6) N. Dershowitz and E. M. Reingold. *Calendrical Calculations.* Cambridge University Press, 1997.

7) P. Grenon and B. Smith. SNAP and SPAN: Towards dynamic spatial ontology. *Spatial Cognition & Computation*, 4(1):69 – 104, 2004.

8) N. Guarino and C. Welty. A formal ontology of properties. In *Proceedings of EKAW-2000: The 12th International Conference on Knowledge Engineering and Knowledge Management*, pages 97–112, 2000.

9) K. Kaneiwa. Order-sorted logic programming with predicate hierarchy. *Artificial Intelligence*, 158(2):155–188, 2004.

10) K. Kaneiwa and R. Mizoguchi. Ontological knowledge base reasoning with sort-hierarchy and rigidity. In *Proceedings of the 9th International Conference on the Principles of Knowledge Representation and Reasoning (KR2004)*, pages 278–288, 2004.

11) K. Kaneiwa and R. Mizoguchi. An order-sorted quantified modal logic for meta-ontology. In *Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX2005)*, pages 169–184. LNCS 3702, Springer–Verlag, 2005.

12) K. Kaneiwa and R. Mizoguchi. Distributed reasoning with ontologies and rules in order-sorted logic programming. *Journal of Web Semantics*, 7(3):252–270, 2009.

13) O. Lassila and R. Swick. Resource Description Framework (RDF) model and syntax specification, W3C Recommendation, http://www.w3.org/TR/PR-rdf-syntax, 1999.

14) M. J. Loux and D. Zimmerman, editors. *Oxford Handbook of Metaphysics*. Oxford University Press, 2003.

15) E. J. Lowe. *The Four-Category Ontology: A Metaphysical Foundation for Natural Science*. Oxford University Press, 2005.

16) M. Manzano. Introduction to many-sorted logic. In *Many-sorted Logic and its Applications*, pages 3–86. John Wiley and Sons, 1993.

17) C. Masolo, S. Borgo, A. Gangemi, N. Guarino, A. Oltramari, and L. Schneider. Wonderweb deliverable d17. the wonderweb library of foundational ontologies and the dolce ontology, 2002.

18) P. Materna. Concepts and objects. *Acta Philosophica Fennica*, 63, 1997.

19) B. Motik. On the Properties of Metamodeling in OWL. *Journal of Logic and Computation*, 17(4):617–637, 2007.

20) H. J. Ohlbach and D. M. Gabbay. Calendar logic. *Journal of Applied Non-Classical Logics*, 8(4), 1998.

21) G. Rosenkrantz and J. Hoffman. The independence criterion of substance. *Philosophy and Phenomenological Research*, 51(4):835–853, 1991.

22) B. Smith. Basic concepts of formal ontology. In *Formal Ontology in Information Systems*. 1998.

23) L. H. Tharp. The characterization of monadic logic. *Journal of Symbolic Logic*, 38(3):481–488, 1973.

24) C. Welty and W. Andersen. Towards ontoclean 2.0: A framework for rigidity. *Applied Ontology*, 1(1):107–116, 2005.

25) C. Welty and N. Guarino. Supporting ontological analysis of taxonomic relationships. *Data and Knowledge Engineering*, 39(1):51–74, 2001.