An Upper Ontology for Event Classifications and Relations

Ken Kaneiwa[1], Michiaki Iwazume[1], and Ken Fukuda[2]

[1] National Institute of Information and Communications Technology
3-5 Hikaridai, Seika, Soraku, Kyoto 619-0289, Japan
{kaneiwa,iwazume}@nict.go.jp
[2] National Institute of Advanced Industrial Science and Technology
2-42 Aomi, Koutou-ku, Tokyo 135-0064, Japan
fukuda-cbrc@aist.go.jp

Abstract. For knowledge representation and reasoning, there is a need to consider the nature of events because event data describe various features and behaviors of the occurrences of actions and changes in the real world. In this paper, we propose to establish an upper event-ontology in order-sorted logic as an infrastructure for event knowledge bases. Our event ontology contains a classification of event entities (e.g., natural events and artificial events) and event relationships (e.g., causal relations and next-event relations). These ontological characterizations are needed for a theoretical basis of applications such as implementation of event databases, detection of event relationships, and annotation of event data.

1 Introduction

Events are entities classified in a formal ontology that are difficult to treat in knowledge base systems. To describe the occurrences of actions and changes in the real world, knowledge base designers have to consider that objects and properties are static, but events are dynamic.

In the fields of logic, linguistics, ontology, artificial intelligence, and deductive databases, the nature of events has been investigated as follows. Allen et al. [1] explained that events were methods used to classify useful and relevant patterns of change rather than entities in the real world. Sowa [12] categorized events as changes that occur in the discrete steps of a process. Active object-oriented databases [3, 2] have event specification facilities to describe complex events. Galton and Augusto [2] attempted to combine the two kinds of event definitions of knowledge representation and databases. As an object-oriented approach, Worboys and Hornsby [13] proposed the foundations of modeling objects and events for dynamic geospatial domains.

Upper ontologies have defined the nature of events for information systems. In fact, several standard upper ontologies are distributed on the Web. Guarino's group built the DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering)[8], in which events are a subclass of perdurant occurrences that are disjoint to the entities of endurant, quality, and abstract. The SUMO (Suggested

Upper Merged Ontology)[9], designed by the IEEE Standard Upper Ontology Working Group consists of a set of concepts, relations, and axioms where abstract and physical entities are divided, in which the physical entities are classified into objects and processes. The OpenCyc [10] contains an upper ontology, in which event entities are defined as temporal and intangible.

In this paper, we propose to extend an event ontology, which is designed in the structural representation of order-sorted logic [5], to an infrastructure for event knowledge bases. First, we classify the types of events (e.g., natural events and artificial events). Our approach attempts to construct event classifications based on our two ontological views: component structures and semantic functions of events. The component structures lead to the knowledge representation of events (e.g., the argument structures of predicate formulas) and the semantic functions imply the logical and ontological semantics of events for reasoning. Second, we introduce event relations (e.g., causal relations and next-event relations) capturing the differences between instances and classes of events. In order that these relations represent a sequence of events and constraints in event knowledge bases, we modify Smith's relationship ontology for bioinformatics [11].

2 Ontology Description in Order-Sorted Logic

First, we briefly explain the basic usage of order-sorted logic for formalizing ontologies and concept definitions. Ontologies are usually represented by concept hierarchies. We build an event ontology as a sort-hierarchy in order-sorted logic where many sort symbols denote concepts as sets of individuals and their ordered relation corresponds to a concept hierarchy. The following form represents an ontology consisting of four sorts where a sort symbol s_1 has two subsort symbols s_2 and s_3, and the sort symbol s_3 has a subsort symbol s_4.

Sort s_1
    Subsort s_2
    Subsort s_3
        Subsubsort s_4

Such a sort-hierarchy is interpreted according to the semantics of order-sorted logic in such a way that every instance of sort s_2 is also an instance of its supersort s_1. Let S be the set of sort symbols. In the syntax of order-sorted logic, a subsort relation $\sqsubseteq$ $(\subseteq S \times S)$ of the above ontology is declared as follows.

$$s_4 \sqsubseteq s_3,\ s_3 \sqsubseteq s_1,\ s_2 \sqsubseteq s_1$$

In the usual manner, each sort symbol expresses a set of physical objects (e.g., humans) or a set of abstract objects (e.g., rational numbers) where the same type of object belongs to a sort. In an extension to the sort representation, we regard event instances as the occurrences of an event that belong to an event sort. This notion is based on the fact that if the same event happens many times, their respective occurrences can be recognized as instances of the same event sort.

Every event sort E has to be distinguished from every object sort. Each physical or abstract object of a sort has an identifier, but each instance of an event sort has a different time and location. Then, two special sorts $Time$ and $Location$ are introduced as the set of times and the set of locations, respectively. In addition, every instance of event sorts can play the role of an n-ary predicate representing a relationship among components of the event (e.g., actor and object). The following sort declaration of an n-ary predicate fixes the component structure of event sort E in a sorted signature.

$$E \colon \langle s_1, \ldots , s_n \rangle$$

where sorts $s_1, \ldots , s_n$ are disjoint to event sort E. This declaration implies that every instance of event sort E is an n-ary predicate whose argument structure is of sorts $s_1, \ldots , s_n$. In the next section, using the sort representation, we define classifications and relations of events in an event ontology.

3 Event Classifications in an Ontology

Components of Events: We show the following ontology that classifies natural events, artificial events, dynamic states, and static states together with their respective component structures.

- Event
  - NaturalEvent
    - $Occurence1 \colon \langle Time, Location \rangle$
    - $Occurence2 \colon \langle Object, Time, Location \rangle$
  - ArtificialEvent
    - $Action1 \colon \langle Agent, Object, Time, Location \rangle$
    - $Action2 \colon \langle Agent, Time, Location \rangle$
    - $Action3 \colon \langle AgentGroup, Time, Location \rangle$
  - DynamicState
    - $ObjectChange \colon \langle Object, Time, Location \rangle$
    - $EnvironmentChange \colon \langle Time, Location \rangle$
  - StaticState
    - $ObjectState \colon \langle Object, Time, Location \rangle$
    - $EnvironmentState \colon \langle Time, Location \rangle$

In the event ontology, natural events and artificial events are defined as having different component structures. If an event happens naturally and there is no actor as a component of the event, the event is called a natural event. A natural event does not contain any actor, but there may be a main component of the event. For example, a volcanic eruption is caused by a volcano and a flood is caused by a sea or river. In contrast, earthquakes and typhoons are described with locations, without any component. For example, an earthquake happens in a location and a typhoon is moving to a location. In our opinion, this difference in component structures helps in representing events in predicate formulas. More precisely, natural events without any component are represented by a binary predicate formula of time and location. Natural events with a main component are represented by a trinary predicate formula of an object, time, and location.

If there are some actors as components that generate an event, then it is called an artificial event. The three types of artificial events are defined with respect to the actors and other components of the events as follows:

(1) Actions from agents to objects or agents (e.g., murder and environmental disruption)
(2) Intransitive actions (e.g., breathing and moving)
(3) Actions with many agents (e.g., conference and discussion)

There are relationships among agents and objects in the three types of artificial events. The first type of artificial event is an action that is generated by an agent against objects (or agents) in the world. This event is described by a quaternary predicate formula of an actor, an object (or agent), time, and location. The second type of artificial event is an action describing the motion of an agent. This event consists of an actor and does not contain other objects. Hence, the event is described by a trinary predicate formula where the arguments are an actor, time, and location. Unlike the first two event types, the third type of artificial event is the whole entity that is generated and in which many agents participate. This event type does not focus on the action of each agent, although the internal action of the event may contain the first two event types.

In addition to these events, we deal with states that are divided into states of objects and states of environments. As a result of this, the object state is expressed by a trinary predicate formula because the state has the three components of an object, time, and location. The environment state is described by a binary predicate formula that consists of the two components of time and location. We regard states of objects or environments as events because states are events representing the changes of objects or environments in dynamic time and location. In philosophical research, there is a distinction between events and states, but states are categorized as events in the above ontology. In practice, the unified category of events and states is useful for us to formalize their relationships, such as causal relations discussed in Section 4. Related to this, Hobbs et al.[4] defined event concepts including state concepts for the objective of tagging in natural languages.

Definition 1 (Activity of States) *A state is dynamic if the state implies the activity and dynamic change of an object or environment in time. A state is static if the state implies the static property in time.*

For instance, “rolling continues”, “rising/dropping”, “slightly active”, and “become strong,” which represent the changes of states, are dynamic states, but “hot”, “cold”, and “fine” are static states. The dynamic state “become higher” indicates that the temperature of the static state “low temperature” will become higher because of a change of the static state.

Semantic Functions of Events: We next characterize semantic functions of events in addition to the event classification given by event components. Semantic functions formally and semantically indicate that each event implies the functional change and behavior of objects in the real world. Let us introduce six semantic functions of events in the event ontology as follows.

EventSemanticFunctions
  StateChange
  TemporalExistenceChange
  SpatialExistenceChange
  CardinarityChange
  Comparison
  ObjectIdentificationChange

These semantic functions are strongly related to the meaning of verb expressions in natural languages because each verb is a word to express an action or state.

Related to this end, Kaneiwa and Mizoguchi[6, 7] have proposed a property classification in an upper ontology where properties are divided by their rigidity in modal order-sorted logic. Using their formalization of the ontology, the semantic functions of events can be defined by modalities, cardinalities, quantifications, and logical connectives. In the semantic functions, each event affects an object or an environment and therefore changes their property or state in the next time. Therefore, we define the semantic function of an event in logic by the truth value changes of properties and states over dynamic times and locations. To embed such complex events in semantics, we have to introduce equations and generalized quantifiers as well as modal and temporal operators.

Definition 2 (State Change) *An event is a state-change-event if the occurrence yields the change of states from now to the next time as follows:*

$$F_1 \rightarrow \bigcirc F_2$$

The modal operator $\bigcirc F_2$ implies that F_2 is true at the next time. Therefore, the semantic function of event e is defined by the statement $F_1 \rightarrow \bigcirc F_2$, which implies that if a state or property F_1 is true, then the event e yields another state or property F_2 at the next time.

Definition 3 (Existential Change over Time) *An event is a temporal-existence-change-event if its occurrence changes the existence of an object according to a change in time as follows:*

$$\Box_P(\neg E(x)) \wedge E(x)$$

The temporal operator $\Box_P F$ implies that the formula F was always true in the past. The semantic function of event e contains the existential formula $\Box_P(\neg E(x)) \wedge E(x)$ implying that an object x did not exist in the past but it exists now.

Definition 4 (Existential Change over Space) *An event is a spatial-existence-change-event if its occurrence changes the existence of an object depending on movement through space as follows:*

$$\neg E(x) \land \blacklozenge E(x)$$

The spatial operator $\blacklozenge F$ implies that the formula F is true in a place accessible from here. Hence, the semantic function of $\neg E(x) \land \blacklozenge E(x)$ indicates that an object x does not exist here, but it exists in a place accessible from here.

Definition 5 (Cardinality Change) *An event is a cardinality-change-event if it changes the cardinality of objects as follows:*

$$\exists_i x F(x) \rightarrow \bigcirc \exists_{>i} x F(x)$$

Let n be a natural number. The generalized quantifier $\exists_n x F(x)$ (called counting quantifier) expresses the existence of n objects x such that the formula $F(x)$ is true. In the cardinality change, we have to introduce a variable i of natural numbers for the generalized quantifier $\exists_i$ that is more expressive than the the generalized quantifier $\exists_n$. Hence, the semantic function of $\exists_i x F(x) \rightarrow \bigcirc \exists_{>i} x F(x)$ indicates that if there exist i objects x such that $F(x)$ is true, then there exist more than i objects x such that $F(x)$ is true the next time. This function means that the number of objects increases by the next time because of the event occurrence.

Definition 6 (Comparison) *An event is a comparison event if the attribute value of an object is found to change when comparing that value with the attribute value at the next time as follows:*

$$\exists y(\mathit{Value}(x) = y \rightarrow \bigcirc(\mathit{Value}(x) > y))$$

In the semantic function, $\mathit{Value}(x)$ denotes the attribute value of an object x and the inequality symbol $>$ is used to compare the value of the attribute now to that of the next time. To compare the value of x now to that of the next time, a fixed value y and the changed value $\mathit{Value}(x)$ appear in the formula $\exists y(\mathit{Value}(x) = y \rightarrow \bigcirc(\mathit{Value}(x) > y))$.

Definition 7 (Object Identification Change) *An event is an object-identification-change-event if the essential property of an object is changed and therefore the object cannot be recognized as the former object at the next time as follows:*

$$\exists y(x \equiv y \rightarrow \bigcirc(x \not\equiv y))$$

The object identification is lost at the next time after the occurrence of an object-identification-change-event. Hence, the function of $\exists y(x \equiv y \rightarrow \bigcirc(x \not\equiv y))$ implies that if there exists an object y such that y is identical to object x, then the event makes them different at the next time. This means that an object x is changed into another object y.

There may be several semantic functions for an event. In other words, many logical operations can be used to functionally define an event. Using the above definitions, a correspondence list of event predicates and quantified modal formulas denoting events and their semantic functions is shown in Table 1.

Table 1. Semantic Functions of Events

Event predicates	Quantified modal formulas
$Cure(x, y)$	$\neg Healthy(y) \wedge (Act(x, y) \vee Affect(x, y)) \rightarrow \bigcirc Healthy(y)$
$Drink(x, y)$	$(InMouth(y) \wedge Swallow(x, y) \wedge \neg Bite(x, y)) \rightarrow \bigcirc InBody(y)$
$Stop(x, y)$	$Active(y) \wedge (Action(x, y) \vee Affect(x, y)) \rightarrow \bigcirc \neg Active(y)$
$Die(x)$	$\Diamond_P(\Box_P \neg E(x) \wedge E(x)) \wedge (\neg E(x) \mathcal{S} E(x)) \wedge \Box_F(\neg E(x))$
$Print(x, y)$	$\exists v(Object(v) \wedge Act(x, v) \rightarrow \bigcirc \exists z(Above(z, v) \wedge Press(Mark(y), z)))$
$BeBorn(x)$	$\Box_P(\neg E(x)) \wedge E(x)$
$Go(x)$	$Act(x) \rightarrow \bigcirc(\neg E(x) \wedge \blacklozenge E(x))$
$Separate(x, y)$	$Adjoint(x, y) \vee Overlap(x, y) \rightarrow \bigcirc(\neg Adjoint(x, y) \wedge Overlap(x, y))$
$Understand(x, y)$	$\exists v((v = Fact(x) \vee v = Content(x) \vee v = Meaning(x)) \wedge Get(x, v))$
$Increase(x)$	$\exists i(Nat(i) \wedge \exists_i x\ Countable(x) \rightarrow \bigcirc \exists_{>i} x\ Countable(x))$ $\exists y(Rel(y) \wedge Quantity(x) = y \rightarrow \bigcirc(Quantity(x) > y))$
$Decrease(x)$	$\exists i(Nat(i) \wedge (i > 0) \wedge \exists_i x\ Countable(x) \rightarrow \bigcirc \exists_{i-1} x\ Countable(x))$ $\exists y(Rel(y) \wedge Quantity(x) = y \rightarrow \bigcirc(Quantity(x) < y))$
$Raise(x)$	$\exists y(Location(x) = y \rightarrow \bigcirc(\neg E(x) \wedge \blacklozenge(E(x) \wedge Location(x) > y))$ $\exists y(Value(x) = y \rightarrow \bigcirc(Value(x) > y))$
$High(x, y)$	$\exists r(Rel(r) \wedge Value(x) = r \wedge (Value(y) < r))$
$Change(x)$	$\exists y(x \equiv y \rightarrow \bigcirc(x \not\equiv y))$
$Make(x, y)$	$\exists z(Act(x, z) \rightarrow \bigcirc(BeBorn(y) \wedge y \not\equiv z \wedge Valuable(y)))$ $Act(x) \rightarrow \bigcirc BeBorn(y)$

4 Event Relations: Cause-Effect and Others

We modify Smith's ontology [11] to define relationships between events in knowledge bases, which are important for describing a sequence of many events, e.g., a sequence of causes and effects. For such event relations, two events may be connected by causal, temporal, and spatial constraints, and the distinction between event instances and event classes should be carefully considered.

Before defining the event relations, we generally construct a binary relationship ontology for objects and events as follows:

BinaryRelation
    $ObjectRelation \colon \langle Object, Object \rangle$
    $EventRelation \colon \langle Event, Event \rangle$
    $CausalRelation \colon \langle Object \sqcup Event, Object \sqcup Event \rangle$

A causal relation is a complex relation over various entities such as objects and events (including states). For example, if John sets fire to a house, the cause of the fire is John. Then, we can write the causal relation $John \rightarrow_{cause} fire$. As a part of the above binary relations, we give an event-relationship ontology that contains various event relations as follows:

- EventRelation
  - EventInstanceRelation
    - EventTemporalRelation
    - EventSpatialRelation
    - NextEventRelation
    - PartOfRelation
    - EventInstanceCausalRelation
      - DisjointCausalRelation
      - ContinuousCausalRelation
      - OverlappingCausalRelation
      - PartialCausalRelation
  - EventClassRelation
    - DisjointRelation
    - SubclassRelation
    - PartOfRelation
    - EventClassCausalRelation

To formalize event relations in the above ontology, event instances and event classes have to be distinguished for identifying elements of the relations. An event instance is the actual occurrence of an event (e.g., the occurrence of an earthquake), but an event class is a set of event instances corresponding to a type or feature of events (e.g., the set of event instances of earthquakes). We should note that each event instance involves time and location information in which the event actually occurs, but each event class does not need to have the information. Following the distinction between instances and classes of events, their event relations are respectively defined as follows:

Definition 8 (Event-Instance Relations) *Let* e_1, e_2 *be two event instances. Then, a binary relation* $r(e_1, e_2)$ *between* e_1 *and* e_2 *is called an event-instance relation, which is defined by the following:*

(1) If an event instance e_1 *causes an event instance* e_2*, then the causal relation* $e_1 \rightarrow_{cause} e_2$ *holds.*
(2) If an event instance e_2 *occurs after event instance* e_1*, then the next-event relation* $e_1 \rightarrow_{next} e_2$ *holds.*
(3) If an event instance e_1 *temporally includes an event instance* e_2*, and* e_1 *occurs in a spatial part of* e_2*, then the event-part-of relation* $e_1 <_{po} e_2$ *holds.*

The temporal relations uniquely determined by event instances further divide the event-instance causal relations into disjoint, continuous, overlapping, and partial causal relations (as shown in Fig.1).

Definition 9 (Event-Class Relations) *Let* E_1, E_2 *be two event classes. Then, a binary relation* $R(E_1, E_2)$ *between* E_1 *and* E_2 *is called an event-class relation, which is defined by the following:*

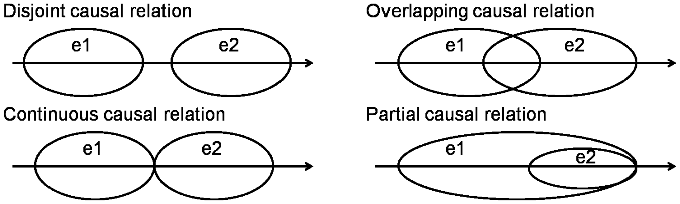

Fig. 1. Causal Relations over Time

(1) *If each event instances of E_1 and E_2 cannot occur simultaneously, then the event disjoint relation $E_1||E_2$ holds.*
(2) *If every instance of E_1 belongs to E_2, then the event-subclass relation $E_1 \sqsubseteq E_2$ holds.*
(3) *If for every event instance e of E_1, there is an event instance e' of E_2 such that $e' <_{po} e$, then the event-class part-of relation $E_1 <_{po} E_2$ holds.*
(4) *If for every event instance e of E_1, there is an event instance e' of E_2 such that $e \rightarrow_{cause} e'$, then the event-class causal relation $E_1 \rightarrow_{cause} E_2$ holds.*

For example, the event disjoint relation $heavy\_rain||fine\_weather$ holds because they cannot happen simultaneously. Event classes are structured by an event-subclass relation. The event class *earthquake* is used to describe an event concept as an occurrence set of earthquakes. However, a unique event instance (e.g., Sumatra earthquake as an event instance of *earthquake*) can be regarded as the only instance of the event class, which is a singleton. Then, the event-subclass relation $\sqsubseteq$ is declared as follows:

$$\{Sumatra\_earthquake\} \sqsubseteq earthquake \sqsubseteq natural\_disaster \sqsubseteq disaster$$

Each event-instance causal relation is transitive. That is, if $e_1 \rightarrow_{cause} e_2$ and $e_2 \rightarrow_{cause} e_3$, then $e_1 \rightarrow_{cause} e_3$. However, the event-class causal relation is not transitive as follows:

Proposition 1 (Intransitivity of Causal Relations). *Let E_1, E_2, E_3 be three event classes. Then, there are event-class causal relations $E_1 \rightarrow_{cause} E_2$ and $E_2 \rightarrow_{cause} E_3$ such that $E_1 \rightarrow_{cause} E_3$ does not hold.*

We discuss a part-of relation between events. Each event contains two types of partial events. The first partial event is an essential event that always occurs in the whole event. If we consider an earthquake event, a P-wave and S-wave are essential events of every earthquake. The second partial event is a non-essential event in the whole event. For example, a rescue operation is a non-essential event of an earthquake. The rescue operation is an artificial event, which is an action performed by rescuers on victims. This partial event consists of rescuers, victims, suffering things, and stricken areas that are people, objects, and locations. From the view of instances and classes of events, the following property is derived. The essential event parts are often defined by a part-of relation between event classes. If an event instance has a non-essential event part, then the event instance is defined by a part-of relation between event instances.

5 Conclusion

The result of this research was the characterization of event entities in an upper ontology and logical formalization. For natural/artificial events and dynamic/static states in the ontology, the event component structures were classified so that they correspond to the argument structures of predicate formulas. As an alternative view, the semantic functions of events were analyzed in expressive logical formulas that enabled us to infer logical conclusions from event occurrences. Furthermore, we explained event relations where four types of causal relations and other relations were introduced by distinguishing event instances and classes. These relations would be necessary for relational descriptions of events in knowledge bases. In future work, we plan to develop a reasoning system for the event relations based on our proposed event ontology.

Acknowledgment

This work was supported by KAKENHI (Grant-in-Aid for Scientific Research) on Priority Areas (No.18016032, "Systems Genomics") from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

1. James F. Allen and George Ferguson. Actions and events in interval temporal logic. *Journal of Logic and Computation*, 4(5):531–579, October 1994.
2. A. Galton and J. C. Augusto. Two approaches to event definition. In *Proceedings of DEXA 2002, LNCS 2453*, pages 547–556, 2002.
3. S. Gatziu and K. R. Dittrich. Events in an Active Object-Oriented Database System. In *Proc. of 1st Inter. Work. on Rules in Database Systems*, 1994.
4. J. Hobbs and J. Pustejovsky. Annotating and reasoning about time and events. In *The Language of Time.* Oxford University Press, 2005.
5. K. Kaneiwa. Order-sorted logic programming with predicate hierarchy. *Artificial Intelligence*, 158(2):155–188, 2004.
6. K. Kaneiwa and R. Mizoguchi. Ontological knowledge base reasoning with sort-hierarchy and rigidity. In *Proc. of the 9th International Conference on the Principles of Knowledge Representation and Reasoning (KR2004)*, pages 278–288, 2004.
7. K. Kaneiwa and R. Mizoguchi. An order-sorted quantified modal logic for meta-ontology. In *Proc. of TABLEAUX2005*, pages 169–184. LNCS 3702, 2005.
8. 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.
9. I. Niles and A. Pease. Towards a standard upper ontology. In *Proc. of the 2nd Inter. Conf. on Formal Ontology in Information Systems (FOIS-2001)*, 2001.
10. OpenCyc. http://www.opencyc.org.
11. B. Smith, W. Ceusters, B. Klagges, J. Kohler, A. Kumar, J. Lomax, C. Mungall, F. Neuhaus, A. Rector, and C. Rosse. Relations in biomedical ontologies. *Genome Biol*, 6(5):R46, 2005.
12. J. F. Sowa. *Knowledge Representation.* Brooks/Cole, 2000.
13. M. F. Worboys and K. Hornsby. From objects to events: Gem, the geospatial event model. In *Proc. of GIScience 2004*, pages 327–344, LNCS 3234, 2004.