2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress

An Ontology Pattern for Emergency Event Modeling

Wei Liu[1], Yue Tan[1], Ning Ding[1], Yujia Zhang[1], Zongtian Liu[1]
School of Computer Engineering and Science, Shanghai University
Shanghai, China
liuw@shu.edu.cn

***Abstract*—Ontologies play an increasingly important role in emergency domain. They have become essential components of emergency response system. Development of emergency ontology is a laborious and difficult task because of the increasingly scale and complexity of the ontology. This paper proposes the concept of Event Ontology Pattern (EOP) and the development method of EOP, which is to capture formally reoccurring models and realize the reuse of existing emergency ontology vocabulary, and then facilitate emergency ontology development, help users avoid some of the most frequent modeling mistakes. An example of pollution emergencies ontology pattern is provided to demonstrate the creation of EOP and how to reuse EOP to develop new ontologies.**

Keywords—Ontology Pattern, Event Ontology, Pollution Emergency

I. INTRODUCTION

In the last decade, ontologies play an increasingly important role in the big data processing of emergency domain. By analyzing the objects in emergencies, concepts and the relations between these concepts can be identified, and then emergency ontologies could be constructed. The emergency ontologies can be used to capture and reasoning the evoluation of emergency. They have become essential components of emergency alerting and response system. Whereas, traditional ontologies are concept-centered, there exist deficiencies while modeling knowledge of emergency domain: (1) separateness of concepts (the person, objects, places and action of event are not organized as a whole knowledge unit); (2) insufficient capacities of capturing dynamic aspects of emergency domain, especially the status-changing of event over time; (3) relationship between domain concepts couldn't describe semantic relation between events.

Event-centered modeling could capture the dynamic aspects of a domain. Events provide a natural way to describe complicated relations between people, places, actions and objects. Event relationships provide more sophisticated description and reasoning of event-centered concepts. This paper firstly proposed an Event Ontology Model (EOM) structure for representation of generic event information on Web. The event model is proposed to model events in different domains, without making assumptions about the domain-specific vocabularies used. In addition, the model describes event classes and different types of event elements by using extended OWL 2 and integrating some terms from external ontology languages or vocabularies that based on W3C's RDF technology, fox example, reusing Agent class from FOAF[2] to describe actors of event, using terms from Time Ontology[3] to describe time of event, thus to improve the reusability of event element classes.

Generally, it is difficult to develop an event ontology for specific domain because of the increasingly scale and complexity of the ontology. Many developers of ontologies are domain experts rather than logic experts, and focus on the concepts in the domain rather than their ontological representation. While creating specific event ontologies, we found there exist plenty of similar event classes, event elements and event relations structure in these different event ontologies these common event classes, event elements and relation structure can be abstracted as ontology pattern. Event ontology pattern is a proposed solution to capture formally reoccurring models and realize the reuse of existing event ontology vocabulary, and then facilitate event ontology development, help users avoid some of the most frequent modeling mistakes. In this paper, we present the process for construction of event ontology pattern and an algorithm for extraction of event classes and elements. And then, a general environmental pollution ontology pattern is proposed based on shared vocabulary abstracting from air pollution ontology and water pollution ontology[1].

The remainder of this paper is structured as follows. First, we introduce some concepts related to event and event ontology. Section 3 introduces the conceptual foundation for the pattern, and demonstrates how to create a pollution emergency ontology pattern. In section 4, we conclude by summarizing our results and pointing out directions for future work.

II. EVENTS AND EVENT ONTOLOGY MODEL STRUCTURE

In recent years, researchers have proposed a variety of event-based model for knowledge representation [4-7]. These models are different from each other because they focus on specific domain and have different level of formalization. But most of them align to the 5W model (Who, What, When, Where, How). 5W models can be used to describe a static event knowledge unit structure, but they lack the ability to describe the dynamic characteristics of event, especially the changing of internal status during event happening. In additional, most event ontologies currently are created manually by developer, and the language expressions of event in the text are always ignored, in fact, which play important role in the text-based automatic detection and identification of

978-1-5090-4065-0/16 $31.00 © 2016 IEEE
DOI 10.1109/DASC-PICom-DataCom-CyberSciTec.2016.44

events. This section, we will introduce an event model structure for representation of generic event information on Web.

A. Event Specification

Definition 1. Event We define an event as a thing happens in a certain time period and place, which some actors participate in and show some action features, along with the changing of internal status. Event e can be defined as a 6-tuple formally:

$$Event ::=< A, O, T, P, S, L >$$

We call elements in 6-tuple as event elements. Where A means an action or a set of actions happen in an event. In the text, an action is usually regarded as a trigger word to identify an event; O means objects involved in the event, including all participants and entities involved in the event. We name participant as actor and define two types of actors, action initiators and action receiver. T means the period of time that event lasting, the time period can be absolute time and relative time, T can be converted to ordered pairs as [t_1, t_2], when the start time equals end time means the event happens in instant time. P means the location of an event happens; S describes object statuses during an event happens, including *pre-condition* set and *post-condition* set. *Pre-condition* describes a set of object statuses that have to be satisfied for triggering the event. *Post-condition* is a result set of object statuses after event happens; L indicates language expressions of text-based event, it includes a *Core Words Expressions* (CWE) set and a *Core Words Collocations* (CWC) set.

Example 1. *At 7 am today, Hangzhou environmental protection bureau received an emergency report that a chemical tanker truck suffered rollover accident in Huhang highway, part of tetrachloroethane in tank leaked into the Fuchun River.*

Event Name: Huhang highway chemical tanker truck leakage accident

Actions (trigger words): received report, rollover, leak

Objects: Hangzhou environmental protection bureau, tetrachloroethane, chemical tanker truck, Funchun River

Time: At 7 am today

Location: Huhang highway

Status: Pre-status={ $\neg$ Exposed(pollutants), $\neg$ Polluted (Fuchun River), $\neg$ Unhappy(residents around accident area)},
Post- status = {Exposed(pollutants), Polluted(Fuchun River), Unhappy(residents around accident area) }

Language Expression: CWE(event) = {rollover, accident, leakage}, CWC(event) = {"suffer" + CWE}

Definition 2. Event Class is an abstract event that represents a set of events with some common characteristics, denoted as EC:

$$EC = (E, C_A, C_O, C_T, C_P, C_S, C_L)$$

$$C_i = \{c_{i1}, c_{i2}, \cdots, c_{im}, \cdots\} \quad (i \in \{A, O, T, P, S, L\}, m \geq 0)$$

Where E means an event set, is called extension of the event class. C_i is the set of event elements, called intension of the event class. It denotes the common characteristics set of certain event element (element i). C_{im} denotes one of the common characteristics of event factor i. C_{im} is also called event elements class.

B. Event (Class) Relationships

The relationships between the events are divided into two categories: taxonomic relation and non-taxonomic relations. The taxonomic relation describes the hierarchical structure of event classes, typically means subsumption relation or parent-child relation between event classes. Non-taxonomic relations describe the internal semantic relationships between events or event classes, including composition relation, follow relation, causality relation and concurrency relation.

Subsumption relation *(is_a)*: An event class can subsume or be subsumed by other event classes; an event class subsumed by another is called a sub event class (or sub event type) of the subsuming event class (or super event type).

Composition relation (*Composition*):If an event instance e_1 of class EC_1 can be decomposed to several sub-events e_i (i>0, instance of class EC_i) with smaller granularity, and while all the smaller events e_i happend means e happened, there exists composition relationship between e and e_i (or EC_1 and EC_i).

Causality relation (*Causality*): If an event e_1 (instance of EC_1) happened, then another event e_2 (instance of class EC_2) happens at above a specified probability threshold, there is a causality relationship between e_1 and e_2 (or EC_1 and EC_2). EC_1 is cause and EC_2 is effect.

Follow relation (*follow*): Follow means events coming after in time order, as a consequence or result, or by the operation of logic. If in a certain length of time, the event e_2 (instance of class EC_2) follow event e_1 (instance of class EC_1) at above a specified probability threshold, there is a follow relationship between e_1 and e_2 (or EC_1 and EC_2).

Concurrency relation *(concurrency)*: If there are event e_1 (instance of class EC_1) and event e_2 (instance of class EC_2) occur simultaneously or successively in a certain period of time (the two event are coincident events), there is a concurrency relationship between e_1 and e_2 (or EC_1 and EC_2).

In the above five kinds of relationships, subsumption relationship is usually used to describe the relations between the event classes, the rest of the four kinds of relationships can be used to describe both the relations between the event classes and the relations between the event instances.

C. Event Model Structure and Event Ontology

Based on event specifications, an event model structure is presented, as shown in Fig.1. The event model supports the representation of event, event classes and event (classes) relationships.The event model distinguishes the representation of event instance and event class strictly. It references the category representation method from most terms vocabularies in area of knowledge representation, as well, is corresponding with the concepts and individuals in traditional ontology. In the event model, event elements class can be created and maintained independently, also reuse some existing ontology (concept terminology vocabularies). In additional, we established the equivalence between these elements and existing external ontology terms to encourage reusability of the event elements as far as possible. As shown in Fig.1, the actor element of event reuses *Agent* class from the FOAF ontology[2]. Location element reuses SpatialThing class from Geospatial ontologism[8]. The event elements do not intersect with the associated event classes, thus the event element classes can be reused. For example, an event place element class could be *GeographicalRegion* (class of city or region), *LandAddress* or *WaterAddress*, which can be reused by different event classes. For another example, the event time can reuse class TemporalEntity from Time Ontology[3], which can express the instant time or periods of time, as well, it can express absolute time or relative time. The event model structure is designed with a minimum of semantic commitment to guarantee maximal interoperability among event and event classes.

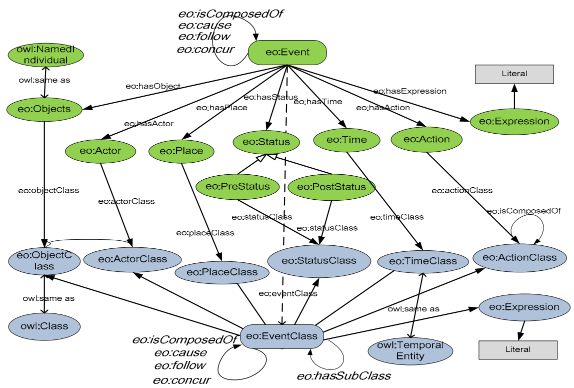

Fig.1. The event model structure

It is important to note that although six elements are defined in the event model structure, only an event trigger word (usually recognized as event action elements) is needed to identify an event or event class while processing text, some other elements can be omitted sometimes. Thus, we can create abstract event classes from text easily, and ensure the simplicity and reusability of the event classes.

Definition 3. Event Ontology: An event ontology is a shared, formal and explicit specification of an event class system model that exists objectively. Event ontology *EO* can be defined as a 4-tuple formally:

$$EO = <UECs, ECs, R, Rules>, \text{ where:}$$

UECs is a set of upper event classes in the event ontology, each *UEC* represents a category, all of the *UEC* constitute a tree category structure of an event ontology; *ECs* is a set of event classes; *R* is a relation set; *Rules* is set of rules be expressed in logic languages, including rules for inference of upper level event categories and rules for relations inference among events.

III. A Pollution Emergency Event Ontology Pattern

A. What is Event Ontology Pattern?

An ontology design pattern (ODP) is "a reusable modeling solution to solve a recurrent ontology design problem"[9]. The basic ideas behind ontology design patterns are very similar to the ideas behind software design patterns. Software design patterns are standard solutions to common situations in software design. The proponents of ontology design patterns claim that they help to improve ontology quality, improve ontology reuse, provide documentation of design rationale, make ontologies more maintainable, and make ontologies easier to understand [10]. Because event ontology is a formal, explicit specification of system model that compose of event classes[11]. ODP is also can be applied to the development of event ontologies. We call these ontology patterns EOP. EOP improve the reusability and scalability of event ontology model, also simplify and standardize the integration of heterogeneous event-based semantic data in various application domains.

The granularity of event ontology patterns differ according to their abstraction degree. In general, while ontology pattern has higher degree of abstraction, it has greater reusability scope, but less reusable elements. Conversely, while the ontology pattern has lower degree of abstraction, it has smaller reusability scope, but more reusable elements. Therefore, the construction of event ontology pattern needs to find a tradeoff between abstraction degree and reusability of event elements. In our view, different event ontology patterns should be established according to different application domain, for example, environmental emergency, traffic accident, terrorist attacks, etc. The event ontology pattern can be created by abstracting specific event ontologies within an application domain. That is, we can create an event ontology pattern by extracting common event classes, event elements and event relationship structures from different specific event ontologies. As a result, while creating a new specific event ontology, the event classes (including event elements, event class relation) in pattern can be inherited, extended or overridden according to the scenarios of modeling.

In this paper, we construct event ontology pattern as following steps:

(1) Create specific event ontology for different sub-domain, and then extract event classes, event class elements that have common characteristics and similar relation structure (maximal similar sub-graphs) from these specific ontologies, next, review and check these common parts.

(2) Create an abstract event ontology as expected EOP by combining and refining the similar classes and elements from step 1.

(3) Create a new specific event ontology by using the EOP from step 2, check the consistence of the event classes and event classes elements between the EOP and the new specific event ontology, revise the EOP.

In environmental domain, environmental pollution issues are diverse, including air pollution, water pollution, soil pollution, nuclear/electric radiation, etc. So, it is difficult and time-consuming to develop ontologies according to different pollution issues. By analyzing the characteristics of different types of pollution incidents and related social events caused by pollution, we found that the event ontology models for these pollution emergencies are always have some similar event classes, event class elements and event relations structure. For example, both air pollution ontology and water pollution ontology have an event class like "*pollution*", and "*pollution*" event class always contains a participant "*polluters*" and a property (object) "*pollutant*". We will illustrate how to create an ontology pattern for pollution emergency in following sections.

B. Creation of Specific Event Ontologies

In this section, we will firstly create event ontology models respectively for particular application scenes of environmental pollution. Due to the limited space, we just discuss two specific ontologies: *air pollution caused by factory gas emissions* and *water pollution caused by vehicle chemical leakage*. And then we will try to found the similar event classes and event class elements among the two ontologies. By analyzing and event-oriented annotating more than 60 articles about air pollution and water pollution from internet, and under the instruction of environmental experts, we create two ontology models as shown in Fig.2 and Fig.3. Fig.2 shows key event classes in air pollution incident and event relations among them (event class elements are hidden). The air pollution event ontology describes *factory gas emission* would result in *air pollution*; and *air pollution* would result in resident's *inhalation of toxic gas* and *death of trees*; resident's *inhalation of toxic gas* would cause *health hazard* and *complaints from residents*; *health hazard* includes *adverse reaction of residents*, such as *dizzy*, *nausea* and *cough*, and *severely health injury*, such as *poisoning* and *death*; resident's *complaints* would result in *investigation* from environmental protect administration (EPA).

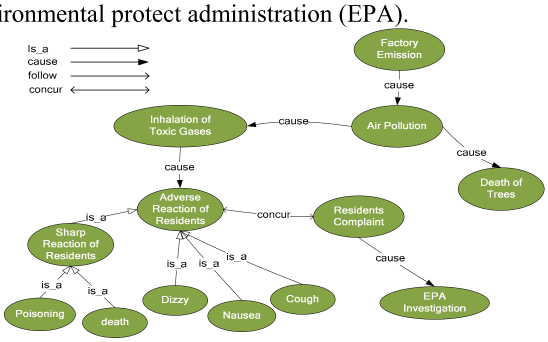

Fig.2. Event Ontology of Air Pollution Caused by Factory Gas Emissions

Table.1 listed key event classes in air pollution event ontology, with their unidirection related event classes and relation type.

Table 1. Event Classes and Relations in Event Ontology of Air Pollution

Event Class Name	*Related Event Class*	*Parent Event Class*
Factory Emission	*<<causal>>_Air Pollution*	*Environment Emergency*
Air Pollution	*<<causal>>_Inhalation of Toxic Gases; <<causal>>_Death of Trees*	*Pollution*
Inhalation of Toxic Gases	*<<causal>>_Adverse Reaction of Residents*	*Exposure to Toxic Substances*
Death of Trees	*<<null>>*	*Ecological Damage*
Adverse Reaction of Residents	*<<is_a>>_Sharp Reaction of Residents; <<is_a>>_Dizzy; <<is_a>>_Nausea; <<is_a>>_Cough; <<concur>>_Residents Complaint*	*Adverse Reaction of Residents*
Sharp Reaction of Residents	*<<is_a>>_Poisoning; <<is_a>>_Death*	*Sharp Adverse Reaction of Residents*
Residents Complaints	*<<causal>>_EPA Investigation*	*Residents Complaints*
EPA Investigation	*<<null>>*	*Official Investigation*

Fig.3 is an event ontology model of *water pollution caused by vehicle chemical leakage*. This event ontology model describes *chemical leakage of vehicle* would result in *river pollution*; *river pollution* would result in *pollution of resident's drinking water* and *death of animals*; *pollution of resident's drinking water* would cause resident's *health hazard*. While residents suffered heath hazard, they will complaints to government or post their experience in internet, at the same time, some of them will make *panic purchase of drinking water*; *health hazard* classified into *severely heath injury*, such as *poisoning* and *death*, and *adverse reaction*, such as *dizzy*, *emesis*, *nausea* and *diarrhea*; resident's *complaints* often cause *reporter's secretly investigation*, also result in *investigation from environmental protect administration*; After *investigation*, EPA would take actions to *purify drinking water source* .

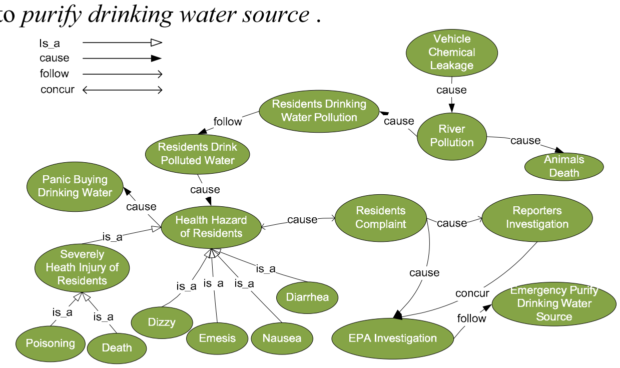

Fig.3. Event ontology model of water pollution incident caused by vehicle chemical leakage

Table.2 listed all event classes in water pollution event ontology, with their unidirection related event classes and relation type.

Table 2. Event Classes and Relations in Event Ontology of Water Pollution

Event Class Name	*Related Event Class*	*Parent Event Class*
Vehicle Chemical Leakage;	*<< causal >>_River Pollution;*	*Environment Emergency;*
River Pollution;	*<<causal>>_Residents Drinking Water Pollution; <<causal>>_Animal Death;*	*Pollution;*
Residents Drinking Water Pollution;	*<<follow>>_Residents drink polluted water;*	*Pollution;*
Residents Drink Polluted Water;	*<<causal>>_Adverse Reaction of Residents;*	*Exposure to Toxic Substances;*

Animal Death;	*<<null>>;*	*Ecological Damage;*
Panic Buying Drinking Water;	*<<null>>;*	*Residents Resist Pollution;*
Adverse Reaction of Residents;	*<<is_a>>_Dizzy; <<is_a>>_Nausea; <<is_a>>_Emesis; <<is_a>>_Diarrhea; <<concur>>_Residents Complaints;*	*Adverse Reaction of Residents;*
Severely Heath Injury of Residents;	*<<is_a>>_Poisoning; <<is_a>>_Death;*	*Sharp Adverse Reaction of Residents;*
Residents Complaints;	*<<causal>>_EPA Investigation; <<follow>>_Reporter Secretly Investigation;*	*Complaints;*
Reporter Secretly Investigation;	*<<causal>>_EPA Investigation;*	*Private Investigation;*
EPA Investigation;	*<<causal>>_Emergency Purify Water Source;*	*Official Investigation;*
Emergency Purify Water Source;	*<<null>> ;*	*Official Response Action;*

C. Creation of Event Ontology Pattern

In order to get an abstract ontology pattern, firstly, we reviewed all event classes from specific ontologies, and extracted a parent event class for each of them (see column "Parent Event Class" in table.1 and table.2), for example, we created a parent class *Ecological Damage* for *Death of Animal*. If there exist common abstract event classes in different specific ontologies, these common abstract event classes, elements and event relation would be selected to create the pattern. We presented an algorithm to create event ontology pattern. Algorithm in table.3 describes how to extract abstract event classes, event class elements (objects and actors) and event relations from specific ontologies. In an event class described by ontology language, the *time* and *location* always defined as a non-formal expression or refer to a external vocabulary, for example, the *location* of *river pollution* class can be defined as a static concept "*WaterAddress*" from an external vocabulary, also can be defined as with natural language "*a certain range of river, lake, sea*". The event *status* and *action* always defined as predicates with constraints, for example, the *post-status* of *environmental emergency* defined as *discharged(pollutants, place)*. As a result, the *time*, *location* and *status* of event class are difficult to be identified and compared accurately by machine. So, while creating an event ontology pattern, we need to refine the time, location and status elements for event classes in pattern artificially.

Table 3. Algorithm of Event Classes and Elements Extraction for Event Ontology Pattern

Algorithm: Event Classes and Elements Extraction for Event Ontology Pattern

Input: *eventOntologySet* //specific event ontologies

Step 1: Traverse event ontologies in *eventOntologySet*, and then find the abstract event classes with the same class name and contained in all event ontologies, store all such abstract event classes into a set, *abstractEventClassSet*.

Step 2: Select a class from *abstractEventClassSet*, named as *abstractEventClassA*, and then check out all child classes of *abstractEventClassA* in different event ontologies respectively, put these child classes into a set named *eventClassASet* respectively.

Step 3: Try to find an event class *eventClassA* in all different *eventClassASet*[*i*] respectively. All these *eventClassA* related to another event class *eventClassB* with the same type of relation named *relationA*, and all the *eventClassB* have a similar parent event class, *abstractEventClassB*, *eventClassA* is the start associated class of *relationA*. If *eventClassAs* are found, then create an abstract relation named *abstractRelation*, it has the same type with *relationA*. *abstractEventClassA* and *abstractEventClassB* are two associated event class of *abstractRelation*. Add *abstractRelation* into the relations of *abstractEventClassA*.

Step 4: check all the event classes *eventClassA* in *eventClassASet*[*i*], and try to find objects or actors with same name, store into *objectASet* and *actorASet*, and then append *objectASet* to *abstractEventClassA*'s object set, append actorASet to *abstractEventA*'s actor set.

Step 5: Repeat step 2 to step 4 until all the event classes in *abstractEventClassSet* are checked.

Output: *abstractEventClassSet*.

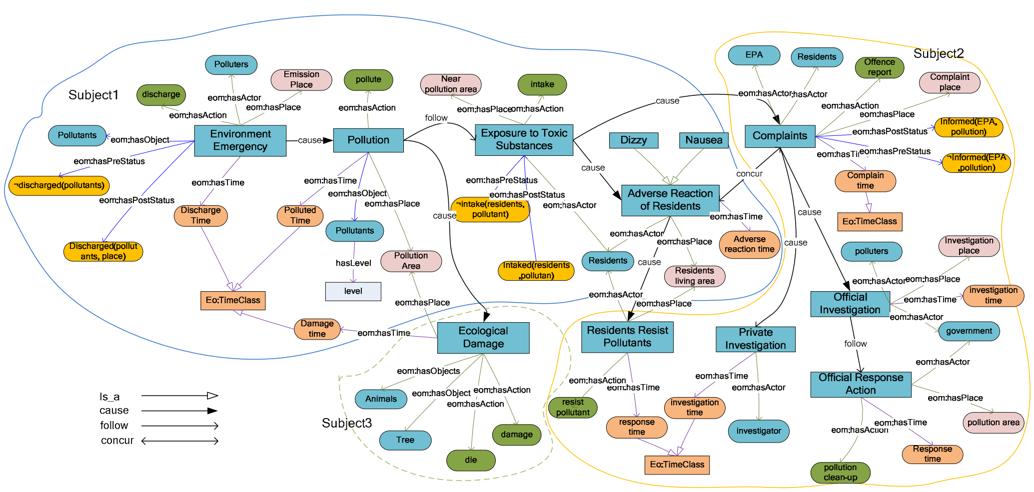

Fig.4. Event Ontology Pattern for Environmental Pollution Emergency

By using the algorithm above, there were 10 abstract event classes extracted from air pollution ontology and water pollution ontology. Also, some common elements were extracted from the two ontologies, such as polluters, pollutants, residents and investigator, etc. But, not all event elements can be extracted automatically. We complement some elements for these abstract event classes artificially, such as time, place and status. As a result, we constructed the event ontology pattern for environmental pollution emergency, see Fig.4. By reusing the ontology pattern, we can create the event ontology for the fog and haze pollution of city, the nuclear pollution, and so on.

IV. Conclusion and Future Work

As a reusable modeling solution to solve a recurrent ontology design problem, Ontology design pattern (ODP)[15] attracted more and more attention and high regards from the semantic web community. Currently, ontology design patterns are widely applied to medical informatics[10,16], geographical ontology[17-19] and bioinformatics[20], etc. The research on event ontology has just started. Event-based ontology modeling is a difficult task because of diversity of people understanding the event. This paper introduced a generic event model structure and an event ontology model for the modeling and reasoning of event-based web knowledge. Furthermore, an event ontology pattern for environmental pollution emergency is proposed, which provides a reusable ontology design solution just like object-oriented design pattern and facilitate the development of event ontology. However, our work remains some limitations, such as lacking of standard event ontology language, further tractable reasoning for event non-taxonomic relations, are need to be solved in the future. In addition, how to construct an event ontology pattern at a reasonable level also need a tradeoff between the reusability and effectiveness.

Acknowledgment

This paper is supported by the Natural Science Foundation of China, No.61305053 and No.61273328.

References

[1] Ministry of Envrionmental Protection of People's Republic of China, 2013 China Environmental Report, http://jcs.mep.gov.cn/hjzl/zkgb/ , (2013)

[2] FOAF Vocabulary Specification, http://xmlns.com/foaf/spec/

[3] Time Ontology in OWL, http://www.w3.org/TR/owl-time

[4] Hage, Willem Robert, et al. Abstracting and reasoning over ship trajectories and web data with the Simple Event Model (SEM). Multimedia Tools & Applications 57(1), 175-197 (2012)

[5] Raimond Y, Abdallah S, Sandler M, et al. The music ontology. Proceedings of the 8th International Conference on Music Information Retrieval. vol. [S.l.], 417-422. ACM, New York (2007)

[6] Scherp, Ansgar, et al. F-a model of events based on the foundational ontology dolce+ DnS ultralight. Proceedings of the fifth international conference on Knowledge capture, pp. 137-144. ACM, New York (2009)

[7] Shaw, Ryan, R. Troncy, and L. Hardman. LODE: linking open descriptions of events. Lecture Notes in Computer Science, vol. 5926, pp. 153-167. Springer, Heidelberg (2009)

[8] Di L, Zhao P. Geospatial semantic web, interoperability. Encyclopedia of GIS, pp. 398-403. (2008)

[9] Daga, E., Blomqvist, E., Gangemi, A., et al. NeOn D2.5.2 Pattern based ontology design: methodology and software support. NeOn project. http://www.neon-project.org. (2010)

[10] Mortensen J M, Horridge M, Musen M A, et al. Applications of ontology design patterns in biomedical ontologies. AMIA Annual Symposium Proceedings, pp. 643–652, American Medical Informatics Association (2012)

[11] W. Liu, et al., Extending OWL for Modeling Event-oriented Ontology, Proc. 4th International Conference on Complex, Intelligent and Software Intensive Systems(CISIS -2010), pp 581-586

[12] Straccia U. A fuzzy description logic for the semantic web. Capturing Intelligence, vol. 1, pp. 73-90. (2006)

[13] Gasse F, Sattler U, Haarslev V. Rewriting rules into SROIQ axioms. Description Logics, pp. 353. (2008)

[14] Krotzsch M, Rudolph S, Hitzler P. ELP: Tractable rules for OWL 2. Proceedings of the 7th International Semantic Web Conference (ISWC-08), Lecture Notes in Computer Science, vol. 5318, pp. 649-664. Springer, Heidelberg (2008)

[15] Gangemi A. Ontology Design Patterns for Semantic Web Content. Proceedings of the 4th International Semantic Web Conference, vol. 3729, pp. 262-276. Springer, Heidelberg (2005)

[16] Martínez-Costa, C., Karlsson, D., Schulz, S.: Ontology Patterns for Clinical Information Modelling. ;In WOP(2014)61-72

[17] Yingjie Hu, Krzysztof Janowicz, David Carral, et al. A Geo-Ontology design pattern for semantic trajectories. Proceedings of 11th International Conference, COSIT 2013, Scarborough, UK, September 2-6, 2013. Lecture Notes in Computer Science, vol. 8116, pp. 438-456. Springer (2013)

[18] Carral, D., Scheider, S., Janowicz, K., et al. An ontology design pattern for cartographic map scaling. Proceedings 10th ESWC2013, Montpellier, France, May 2013, pp. 76-93. (2013)

[19] Carral, D., Janowicz, K., Hitzler, P.: A logical geo-ontology design pattern for quantifying over types. In: Cruz, I.F., Knoblock, C., Kr¨oger, P., Tanin, E., Widmayer, P. (eds.) SIGSPATIAL 2012 International Conference on Advances in Geographic Information Systems (formerly known as GIS), SIGSPATIAL'12, Redondo Beach, CA, USA, November 7-9, 2012. pp. 239–248. ACM (2012)

[20] Aranguren M E, Antezana E, Kuiper M, et al. Ontology design patterns for bio-ontologies: a case study on the cell cycle ontology. BMC Bioinformatics, 9(1):279-282. (2008)