SOFERS: Scenario Ontology for Emergency Response System

Yu Liu
Department of Computer Science, Wuhan University, Wuhan, China
Department of Computer Science, Wuhan University of Science and Technology, Wuhan, China
Email: mail.liuyu@gmail.com

Shihong Chen and Yunhua Wang
Department of Computer Science, Wuhan University, Wuhan, China
Email: Chen_lei0605@sina.com, yhwang@whut.edu.cn

Abstract—With the ability to assist people to make reasonable decisions in an emergency, scenario-response paradigm has been widely regarded as one of the most effective emergency management models. In order to promote the processing capacity of scenario information for Emergency Response System (ERS), a new kind of scenario ontology is proposed in this paper. Following the well-known modeling primitives, the scenario ontology is constructed on the basis of PROTON (Proto Ontology), the reference layer ontology of FactForge that includes some central datasets of Linked Open Data (LOD). The event module, scenario module and mitigation module comprise the core level scenario ontology, defined with some core concepts and relations, which can be reused in the domain level scenario ontology. Considering earthquake is a representative emergency, we analyze the scenario information required by decision makers in some earthquakes, and design the scenario ontology of earthquake as an example in the domain level scenario ontologies. To show the validity of the scenario ontology, we implement the scenario ontology of earthquake with Protégé and develop a prototype system, which can retrieve some parts of scenario information from FactForge and provide a visualization interface for decision makers to browse the scenario instances involved in an earthquake.

Index Terms—Scenario Ontology; Emergency Response System; PROTON

I. INTRODUCTION

To make a reasonable decision, emergency managers or any emergency stakeholders require Emergency Response System to provide enough information. Among the information in emergency, scenario plays a key role and many researchers have paid attention to it. Jiang et al. regarded the scenario as the occurrence and development situation of sudden event which the decision-making staff is facing [1]. Liu thought the scenario is the understanding and convergence about the occurrences and development rules of the previous event [1]. In [2], a broader concept of scenario was defined, including the past, the present, and the future situation, which could be influenced by the decisions and activities. The objective of scenario awareness is to understand the current situation of an emergency and get the critical information that is helpful to predict the future trend [3]. Due to the importance of scenario in ERS, how to efficiently deal with scenario information, such as information retrieval and scenario inference, is a non-trivial problem. In [4], Wirtz et al. pointed out the many challenges associated with the scenario information collection and management in the analyses of natural disasters.

The Semantic Web provides a common framework that allows data to be shared and reused across applications, enterprises, and community boundaries [5]. It is widely accepted in academics that the pervasive and holistic use of semantic technologies leads to a significant improvement of both the development and the usability of disaster management software, described in detail within [6]. As one of the pillars of the semantic web, ontology technology already has been explored to solve the problems that ERS faces. To share and integrate emergency notification messages over distinct emergency response information systems, Malizia et al. developed a ontology for the accessibility of emergency notification system by investigating different sources [7]. On the basis of some experiments with word segment and extraction for emergency plans, combing with meta-words of ontology, an emergency plan ontology model is established and realized with formal language OWL [8]. The success of disaster management largely depends on finding and integrating related information to make decisions during the response phase, Xu utilized ontology technology to reveal the implicit and hide knowledge in geo-information [9]. For the same purpose, Galton indicated how the ontology technology can be used in the context of emergency management [10]. As far as specific emergency concerned, some domain ontologies are proposed, such as ontologies for agriculture disaster [11] and flood risk assessment [12]. From the above introductions, many great results have been achieved by combing ontology technology with emergency management, but few researchers utilize ontology technology to manage scenario information, which is crucial for all decision makers in an emergency.
Since the W3C Linked Open Data project launched in 2007, more and more datasets were published in the web of Data, which had attracted attentions from the area of emergency management. Borges et al. used LOD technology to design the information infrastructure for emergency management system [13]. Silva et al. enhanced the Sahana disaster management system with LOD, leading to a system named Sahana Asia, which can benefit the different agents in the processes of disaster mitigation and preparedness [14]. For the same purpose, taking full advantage of LOD, the scenario ontology we proposed is on the basis of PROTON [15], which is the lightweight upper-level ontology. Because the original intention of PROTON is to cover the conceptual knowledge encoded within the most popular datasets from LOD, the scenario ontology can inherit the feature. Just like PROTON, the scenario ontology can facilitate the access and navigation to a segment of LOD in FactForge, which includes some central datasets of Linked Open Data.

For all we know, the scenario ontology proposed in this paper is the first ontology based on the scenario-response paradigm. By adopting the ontology, Emergency Response System can manage the scenario information efficiently and use reasoning capability of semantic web to analyze the disaster condition. In addition, users can extend the core level scenario ontology to satisfy the requirements of different specific domains. The remainder of this paper is organized as follows. In Section II, we first introduce the architecture of scenario ontology and the upper level ontology. Then, the detail of core level ontology is described in a formal way. Finally, the scenario ontology of earthquake is explained as an example of domain level ontologies. In Section III, we implement the scenario ontology of earthquake with Protégé and develop a prototype system, which can retrieve the scenario information of earthquake from FactForge and provide some analysis functions. At last, we conclude the paper and discuss the future work in Section IV.

II. SCENARIO ONTOLOGY

A. Architecture

The scenario ontology for ERS is divided into three levels. In Figure 1, the levels are layered from top to bottom. The upper level ontology is PROTON, including four modules, which will be introduced later. The core level ontology, the focus of this paper, consists of event module, scenario module and mitigation module. The definitions of class and property are provided in these modules and will be reused in the domain level. In the domain level, some ontologies for more specific emergencies can be defined, such as earthquake and epidemic.

B. Upper Level Ontology

PROTON, the main part of upper level ontology, contains about 500 classes and 150 properties, providing coverage of the general concepts necessary for a wide range of tasks, including semantic annotation, indexing, and retrieval. The System module is an application ontology, which defines several notions and concepts of a technical nature that are substantial for the operation of any ontology-based software, such as semantic annotation and knowledge access tools. The Top ontology module starts with some basic philosophically-reasoned distinctions between entity types, such as “Object”, “Happening” and “Abstract”. The Extent module and Knowledge Management module define much more specific classes [15]. Figure 2 shows some main classes of PROTON. “Entity” is the top class for any kind of objects and things. “Object”, “Abstract” and “Happenings” are the children of “Entity” and further specialized by about 20 general classes. For example, “Role” inherits from “Happening”; “Agent” and “Location” are the subclass of “Object”.

C. Core Level Ontology

In the core level, the classification and object property of emergency are described in the event module, which also includes some basic evolution mechanisms of emergency. The impact degree of emergency depends on disaster-pregnant context, disaster-causing factors and hazard-affected body, which defined in the scenario module. Considering the mitigation plays a key role in the emergency management, we propose the related concept of mitigation in the mitigation module.

To formally define the scenario ontology in core level, we adopt the modeling primitives suggested in [16], which proposed knowledge in the ontology should consist of five kinds of components: concepts, relations, functions, axioms and instances. Following the primitives, the scenario core level ontology (SCL_Ontology) is defined as a quintuple in this paper.
Definition 1. SCL_Ontology = < SCL_Concepts, SCL_Relations, SCL_Functions, SCL_Axioms, SCL_Instance >

SCL_Concepts stands for the set of concept in the core level ontology. SCL_Relations means the set of binary relation between concepts defined in SCL_Concepts. SCL_Functions and SCL_Axioms respectively represent the function set and the axiom set, which are used in the reasoning process. SCL_Instance is the instance set. Figure 3 shows some concepts and relations in SCL_Ontology.

Definition 2. SCL_Concepts := \{ C \}

C signifies the concept related to the core level ontology for ERS. The major concepts are listed below.

Emergency: “Emergency” is a subclass of “Event”, which belongs to the top module of PROTON. Several subclasses of “Event” have been defined in the PROTON, but we do not reuse them. The reason is that the concept of emergency has some different characteristics, such as sudden happening, uncertainty, perniciousness and diffusivity. It determines that the features of emergency people should include not only time and location but also the scenario information.

Natural Disaster: “NaturalDisaster” is a subclass of “Emergency” and contains meteorological disasters, geological disasters, marine disasters, biological disasters, forest and grassland fires, etc.

Accident Disaster: “AccidentDisaster” is a subclass of “Emergency” and contains traffic accidents, production accidents, public infrastructure accidents, nuclear accidents, environmental pollution accident, etc.

Public Health Event: “PublicHealthEvent” is a subclass of “Emergency” and includes infection contagions, food safety hazards, animal epidemic disaster, etc.

Social Security Event: “SocialSecurityEvent” is a subclass of “Emergency” and includes terrorist attacks, economic security events, foreign-related emergencies, mass incidents, etc.

Scenario: “Scenario” is a subclass of “Entity”, which is defined in the top module of PROTON. In general, scenario has two types: natural scenario and social scenario.

Context: “Context” inherits from “Scenario” and describes the context information of emergency. Normally, context in emergency should include weather, topography, social environment, etc.

Hazard Factor: “HazardFactor” is an entity that endangers people, society and natural resources. It is common that some entities in the scenario are the hazard factors. For its importance in the emergency management, “HazardFactor” is defined as the subclass of “Scenario” in the scenario ontology.

Affected Object: “AffectedObject” could be property, people, and environment. The occurrence of affected objects is caused by the hazard factor. Sometimes, the affected objects may become a new hazard factor in certain specific context. Obviously, “AffectedObject” should be a subclass of “Scenario”.

Scenario Condition: “ScenarioCondition” describes the status of affected object, hazard factor and context. If we only consider the affected object, “ScenarioCondition” can be divided into three types: casualty, property loss and environmental damage.

Disposition Method: “DispositionMethod” is subclass of “Abstract”, which is defined in the top module of PROTON. In SCL_Ontology, disposition method is what should be done or have been done to deal with the different scenario conditions. To execute the disposition method, some specific resources should be employed.

Resource: The concept of “Resource” in this paper refers in particular to the mitigation resource, which exists in the scenario of emergency and can be regarded as a part of the scenario, so we define “Resource” by inheriting from “Scenario”. The concept “Resource” can be divided into two subclasses: human resource and material resource.

Definition 3. SCL_Relations := \{ R(c1, c2) \} | c1, c2 \in C

SCL_Relations is the collection of binary relations between the concepts in the core level of scenario ontology and can be reused in the domain level. The major relations are listed below.

Inheritance: “Inheritance” is a basic relation between classes. Just as the above introduced, all the classes defined in the SCL_Ontology inherit from PROTON. For example, “Entity” in PROTON is the superclass of “Scenario”. In the SCL_Ontology, “Emergency” and “NaturalDisaster” have the inheritance relation; “Scenario” and “AffectedObject” have the inheritance relation.

Located in: “Locatedin” is an object property in PROTON. The domain of “Locatedin” is “Entity” and the range of “Locatedin” is “Location”. Hence, we adopt “Locatedin” to express where the emergency happened and the scenario entity exists.

Involved in: “Involvedin” also is an object property in PROTON. The domain of “Involvedin” is “Entity” and “Agent”; the range of “Involvedin” is “Happening”, the superclass of “Event” in PROTON.

Happened at: “Happenedat” identifies when an emergency or scenario occurred. The domain of “Happenedat” is “Entity”; the range of “Happenedat” is

© 2014 ACADEMY PUBLISHER
“TemporalAbstraction”, having several more concrete concepts for temporal, such as “Daytime” and “Season”.

Result in: “Result in” describes the evolution mechanisms of emergency. When the first variable comes from the domain “Emergency”, the relation indicates the mechanism of transformation and spreading between emergencies. To show the derivative mechanism, the domain of the relation should be “DispositionMethod” that means the response for the current emergency causes the occurrence of new emergency.

Affect: The domain of “Affect” is “HazardFactor” and the range of “Affect” is “AffectedObject”. By using this relation, users can formally point out the objects affected by some hazard factors.

Part of: “Part of” is an object property in PROTON and demonstrates inclusion relationship between entities. In some cases, the affected object has multiple parts. This relation can help people to analyze the disaster conditions comprehensively. For example, people and building in a population area are the parts of the population area.

Has: “Has” indicates the properties of scenario objects in the emergency. In PROTON, some relations about possession have been defined (such as “hasMember”, etc.), but none of them is able to exactly reflect the relationship between the scenario conditions and specific entities. That is the reason why we define the relation.

Handled with: “Handled with” is the relationship between “DispositionMethod” and “ScenarioCondition”. To deal with the scenario condition, there always are several methods, which one is the best depends on the concrete scenarios in emergency.

Utilize: “Utilize” describes the relation between “DispositionMethod” and “Resource”. The implementation of disposition method may employ several resources.

\[
\text{Definition 4. } SCL\_Functions := \{ F : s1 \times s2 \times \ldots \times sn \\
\text{hasValue} d dv \\
\text{GreaterThan GreaterThan} \}
\]

SCL\_Functions is the N-ary association set, including some reformulation rules. The formula in the definition 4 illustrates that s will be achieved as soon as the elements in the left part of formula hold.

For example, the extraordinarily serious earthquake is defined as the death toll exceeds 300 and the magnitude of shaking in the earthquake is greater than 7. Here, we formally define the function with the formula (1). Here, “Shaking” is the subclass of “HazardFactor”; “People” is the subclass of “AffectedObject”; “Magnitude” and “Death” are the subclass of “ScenarioCondition”. “Has” is the super property of “Magnitude” and “HasDeath”. “HasValue” is a data property, which describes the values of “ScenarioCondition”. “Greater Than” is used to compare the values of parameters. “ExtraEarthquake” is a subclass of “Earthquake”.

\[
\forall e, s, p, m, d, mv, dv(\text{Earthquake}(e) \circ \\
\text{Shaking}(s) \circ \text{People}(p) \circ \text{Magnitude}(m) \circ \\
\text{Death}(d) \circ \text{InvolvedIn}(s,e) \circ \text{InvolvedIn}(p,e) \circ \\
\text{HasMagnitude}(s,m) \circ \text{HasDeath}(p,d) \circ \\
\text{HasValue}(m,mv) \circ \text{HasValue}(d,dv) \circ \\
\text{GreaterThan}(mv,7) \circ \text{GreaterThan}(dv,300)
\text{ExtraEarthquake}(e))
\]

\[
\text{Definition 5. } SCL\_Axioms := \{ A \}
\]

SCL\_Axioms is the set of axiom in the ontology. In general, SCL\_Axioms are expressed as the definition of concept. We list two types of axioms below.

<table>
<thead>
<tr>
<th>Component</th>
<th>Core Level</th>
<th>Domain Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event</td>
<td>Natural Disaster</td>
<td>Earthquake, Fire, Flood, Landslide, Debris Flow, Tsunami, etc.</td>
</tr>
<tr>
<td></td>
<td>Accident Disaster</td>
<td>Chemical Leaking, Radioactive Material Leaking, etc.</td>
</tr>
<tr>
<td></td>
<td>Public Health Event</td>
<td>Plague, etc.</td>
</tr>
<tr>
<td></td>
<td>Social Security Event</td>
<td>Group Looting, Rumor, etc.</td>
</tr>
<tr>
<td>Context</td>
<td>Hazard Factor</td>
<td>Shaking, Inflamer, Flood Water, Rock Fall, Huge Wave, Chemical Material, Radioactive Material, Illness, Group Panic, etc.</td>
</tr>
<tr>
<td></td>
<td>Affected Object</td>
<td>People, Property, Environment, Densely Populated Area, Significant Building, Transportation Facility, Electricity Facility, Communication Facility, Production Facility, Hillside, River, etc.</td>
</tr>
<tr>
<td></td>
<td>Scenario Condition</td>
<td>Temperature, Precipitation, Wind Scale, Economy, Convention, etc.; Magnitude, Area of Effect, Direction, Volume, Wave Height, Spread Rate, Fear Gauge, etc.; Injury, Death, Blocking, Collapse, Explosion, Broken, Outage, etc.</td>
</tr>
<tr>
<td>Mitigation</td>
<td>Disposition Method</td>
<td>Rescuing, Evacuating, Repairing, Extinguish, Precaution, etc.</td>
</tr>
<tr>
<td>Relation</td>
<td>Result in</td>
<td>Spread, Transform, Derive, etc.</td>
</tr>
<tr>
<td></td>
<td>Happened at</td>
<td>Begin At, End At</td>
</tr>
<tr>
<td></td>
<td>Has</td>
<td>Has Temperature, Has Precipitation, Has Convention, Has Magnitude, Has Area, Has Injure, Has Death, Has Collapse, etc.</td>
</tr>
<tr>
<td></td>
<td>Affect</td>
<td>Destroy, Damage, Infect, etc.</td>
</tr>
<tr>
<td></td>
<td>Utilize</td>
<td>Requisition, Dispatch, etc.</td>
</tr>
</tbody>
</table>

### Table I. Scenario Ontology of Earthquake

<table>
<thead>
<tr>
<th>Component</th>
<th>Core Level</th>
<th>Domain Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concept</td>
<td>Event</td>
<td>Natural Disaster</td>
</tr>
<tr>
<td></td>
<td>Public Health Event</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Social Security Event</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Context</td>
<td>Hazard Factor</td>
</tr>
<tr>
<td></td>
<td>Scenario</td>
<td>Affected Object</td>
</tr>
<tr>
<td></td>
<td>Mitigation</td>
<td>Disposition Method</td>
</tr>
<tr>
<td>Relation</td>
<td>Result in</td>
<td>Spread, Transform, Derive, etc.</td>
</tr>
<tr>
<td></td>
<td>Happened at</td>
<td>Begin At, End At</td>
</tr>
<tr>
<td></td>
<td>Has</td>
<td>Has Temperature, Has Precipitation, Has Convention, Has Magnitude, Has Area, Has Injure, Has Death, Has Collapse, etc.</td>
</tr>
<tr>
<td></td>
<td>Affect</td>
<td>Destroy, Damage, Infect, etc.</td>
</tr>
<tr>
<td></td>
<td>Utilize</td>
<td>Requisition, Dispatch, etc.</td>
</tr>
</tbody>
</table>
Concept Statement: This kind of axiom specifies a unique label for each concept in the ontology. For example, “Scenario” is a label of the concept in the SCL_Ontology.

Equivalence Statement: The equivalent relationship between two classes can be declared by using this kind of axiom, such as Earthquake ≡ Temblor.

Definition 6. SCL_Instances := \{ i \mid i \in SCL_Concepts \cup SCL_Relations \}

SCL_Instances is the instance set of concept and relation in the SCL_Ontology. Because the SCL_Ontology falls outside of the application ontology, none of instance is described here.

D. Domain Level Ontology

Just as the architecture of scenario ontology mentioned above, there are lots of concrete emergency so that we cannot list all of them in this article. Thus, we choose earthquake as an example of domain level ontologies. Earthquake is a typical natural disaster and has some significant characteristics. First, earthquake often causes a large number of casualties and huge economic losses, especially happened in densely populated areas. Second, earthquake maybe produce several secondary disasters. In general, some secondary disasters have more serious damage than the direct disaster. Third, the social impacts of earthquake are more extensive and stronger than other natural disasters. In a word, the scenario of earthquake is representative in the domain ontologies of emergency.

Table 1 lists some concepts and relations in the scenario ontology of earthquake. The correspondences between module in core level and elements in domain level are embodied in the table. The event module includes several secondary or potential events that may break out in an earthquake. During the emergency response of earthquake, the decision makers often want to access a variety of scenario information about the disaster. Just as the table shows, the subclasses of “Context” are “Weather”, “Topography”, etc.; the subclasses of “HazardFactor” are “Shaking”, “Inflamer”, etc.; “AffectedObject” includes “People”, “Property” and “SignificantBuilding”, etc.; “ScenarioCondition” consists of “Temperature”, “Magnitude”, etc. In the Mitigation module, some concrete disposition methods and relief resources are proposed. Except for those concrete concepts, we also defined sub properties by extending the relations in the core level ontology. For example, the categories of “Resultin” include “Spread”, “Transform” and “Derive”, etc. Since “Has” describes the relation between “Scenario” and “ScenarioCondition”, it certainly has lots of sub properties so that we only list part of specific relations here.

Functions and axioms also are the key components in the scenario ontology of earthquake. By reasoning with those rules, lots of potential knowledge can be derived from the apparent scenario information. For example, decision makers always schedule relief resources according the distribution of the people in the initial stages of earthquake response. So, it is crucial question how to locate those densely populated areas. Some places are bound to densely populated areas whenever the earthquake happened, but other places may or may not be the densely populated areas. In the first case, the deterministic relation can be defined as an axiom, such as “MilitaryOrganization ⊆ RescueOrganization”. With regard to the second case, functions can be employed to describe it. In general, schools involved in earthquake should be regarded as the densely population areas and the focus of rescue areas when the earthquake happened at the day time of working days. Hence, we formally express this relationship with formula (2).

\begin{align}
\forall e,s,dt&(\text{Earthquake}(e) \land \text{School}(s) \land \\
&\text{Involved}(i\in e,s) \land \text{DayTimeofWorkingDay}(dt) \land \\
&\text{Happenedat}(e,dt) \rightarrow \text{Densely Populated Area}(s))
\end{align}

Considering the instances in the scenario ontology of earthquake, we adopt the east Japan earthquake in 2011 as an example. The disaster resulted in a great loss to the people and property, and had a profound impact on the security issue of nuclear power station. There were several events happened during the disaster, such as earthquake, tsunami, radioactive material leaking, etc. All of them can be defined as the instances of “Event”, and relations between them can be expressed with sub relation of “Resultin”. Similarly, the concrete hazard factors and affected objects are the instances of classes in the scenario module. For example, Fukushima nuclear power plant is an instance of “ProductionFacility”, which had an “Explosion” condition. Because of the limited space, the details about the instance of east Japan earthquake are not described in this paper.

III. IMPLEMENTATION

We implement the scenario ontology of earthquake with Protégé, an open-source ontology editor, which supports web ontology language from W3C. After importing PROTON into the software, concepts and relations listed in Table 1 are defined. Then, some axioms are added, such as equivalent axiom and disjoint axiom, etc. Functions are stated with SWRL [17], a semantic web rule language based on a combination of OWL with the rule markup language. Although Protégé has provided tools to operate the instance, none of instance is declared by using Protégé. Because the instances always relate with the specific application, we generate some instances in the prototype system introduced below. In total, the scenario ontology of earthquake contains 134 concepts, 56 relations and 15 functions, excluding original concepts and relations in PROTON.

On the basis of the ontology just implemented, we further develop a prototype system, which can retrieve the scenario information of earthquake from FactForge and provide some analysis functions for the decision makers. The prototype system mainly includes three modules: information retrieval module, information processing module and visualization module. The
information retrieval module is responsible for collecting the information stored in FactForge. More concretely, the module only needs to acquire the data about objects in the affected area of earthquake, a fraction of data in FactForge. Except for some basic information (label, Longitude, latitude), there are other background information (such as number of people) about the objects, which also play a key role for people to make decisions. By querying the SPARQL end-point of FactForge, the module retrieves the information according to the prepared query templates. Then, the information processing module asserts the instances of scenario ontology of earthquake by analyzing the retrieved information. Meanwhile, functions of scenario ontology are executed to reason out some potential knowledge. It is obvious that the time required to reason is proportional to the number of instances involved in the earthquake; nevertheless, there are usually plenty of instances in an earthquake. To improve the capability of prototype system, the information that have been reasoned out previously are stored to the Sesame [18] for later reuse.

The visualization module implements a visualization interface for user to browser the scenario information provided by the aforementioned modules. We combine OpenStreetMap [19] with the prototype system so that user can view the scenario information about instances involved in an earthquake on the geographical map.

By utilizing the prototype system, some objects within 200km radius around Tokyo are retrieved from Factforge. The result set contains 8376 facilities, including 5143 settlements, 218 hotels, 32 parks, 2 reservoirs, 1 power station, 1 hospital, etc. In figure 4, parts of result set are shown on the map, and user can access more scenario information (number of students and staffs in a school) by clicking a specific icon. Although the prototype system adopting the scenario ontology can help users to access and analyze part of scenario information about an earthquake, there are two obvious defects. First, the prototype system does not consider the current scenario conditions, which are very important for decision makers. To solve this problem, social and sensor data [20] can be modeled with the scenario ontology and imported into the prototype system. The other defect is the limited number of information in Factforge – only one hospital found within 200km radius around Tokyo. In order to acquire sufficient information, some rich data sources can be employed, such as DBpedia, LinkedGeodata, etc.

IV. CONCLUSION

In this paper, we propose the scenario ontology for emergency response system, which is organized as three levels. Among of them, the core level includes some basic concepts, relations, functions, axioms, which could be extended to implement the scenario ontologies for concrete domains. Further, scenario ontology of earthquake is built on the core level ontology. By using the scenario ontology implemented with Protégé, a prototype system is developed for verifying the validity of scenario ontology. In the future work, the temporal-spatial relations between the scenario instances will be studied to help people to analyze the condition of an emergency. To improve the availability of the prototype system implemented above, we also will consider how to efficiently operate the social and sensor data with the scenario ontology.

ACKNOWLEDGMENT

This work was partly supported by the Science Guidance Project of Education Department of Hubei Province under Grant No. B20140485, Science Foundation for Yong Teachers of Wuhan University of Science and Technology under Grant No. 2012XZ015.

REFERENCES

[1] Zhiying Z., Lei X., Hecheng Z., Shiming L. Study on Unconventional Emergency Scenario Design: Taking Life-


Yu Liu received the master degree in School of Computer Science and Technology of Huazhong University of Science and Technology. He is a PhD candidate in School of Computer Science and Technology of Wuhan University. His research interests are semantic web and knowledge engineering.

Shihong Chen is a professor in School of Computer Science and Technology of Wuhan University. He also is the deputy director of the national engineering research center for multimedia software. His research interests are software engineering and knowledge engineering.

Yunhua Wang is a PhD candidate in School of Computer Science and Technology of Wuhan University. He is also an instructor in School of Computer Science and Technology of Wuhan University of Technology. His research interests are complex system.