ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN ONTOLOGY BASED APPROACH FOR DISASTER PREDICTION

[1,2]YINGLONG MA, [1]XIAO YU

[1] School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China

[2] State Key Laboratory of Computer, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China

E-mail: [1]yinglongma@gmail.com, [2]yuxiao512@163.com

ABSTRACT

Natural disasters are those anomalies created by the nature in the process of the Earth. The harms they have caused to human society tend to be horrifying. Almost every country will be more or less affected by natural disasters annually. In consequence, giving a fairly accurate forecast of the disaster that may happen turns into a vital part of disaster prevention. Traditional research on disasters is mainly based on classical statistical methods or expertise judgment, which is lack of semantic-driven and intelligent reasoning. In this paper, an ontology based approach for disaster prediction was put forward. As the explicit specification of a conceptualization, ontologies can achieve a certain degree of knowledge sharing and reuse, and improve the system ability of communications, interoperability and reliability. On the basis of analyzing the influencing factors and historical evolution of disasters, a domain ontology model of natural disaster in OWL ontology language was created. The hidden influencing factors or disaster chain information can be excavated by the reasoner with domain-specific rules. As a result, we built an ontology-driven disaster prediction and information system. By performing the task of typhoon prediction based on our ontology-based system, our approach can improve the intelligence level of the disaster in its management and forecast.

Keywords: *Ontology Modeling, Semantic Web, Reasoner Mechanism, Disaster Prediction, Information System*

1. INTRODUCTION

Natural disasters such as floods, volcanic eruptions, earthquakes, tornado and windstorm affect thousands of people every year. A natural disaster may cause a great loss of life or property damage; typically leave some economic damage in its wake. As a major adverse event resulting from natural processes of the Earth, a natural disaster poses a serious threat to the industrial and agricultural production. Therefore, the primary task of disaster prevention and mitigation is to improve its forecast level and to take protective and preventive measures in time by analyzing the influencing factors or disaster chain information.

Along with the wide application of ontology in information system and Semantic Web [1], ontology modeling and reasoning has become one of the active areas in the field of ontology-based systems. At present, a great deal of research has been done for the approach of disaster prediction and analysis, and has made progress in the field of disaster forecast. However, most of the existing approaches are mainly based on classical statistical methods or expertise judgment, which are lack of semantic-driven and intelligent reasoning. With the development of ontology theory and technology, a new approach was put forward in this paper, which is aiming at better analysis of natural disasters and making specific research on natural disasters. After analyzing the causing factors and pregnant environment, as well as their relationship in the field of natural disasters, a domain ontology model of some disaster can be created, which is used to mine the hidden influencing factors or disaster chain information by using inference machine with domain-specific rules. It is of great significance to build an ontology driven disaster prediction and information system to prevent or mitigate disaster in reality. This paper will be towards this goal.

This paper is organized as follows. Section 2 is to introduce the ontology representation and reasoning. In Section 3, we focus on the typhoon disasters and further construct the domain ontology about typhoon disasters from the perspectives of disaster-

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

pregnant environment, disaster-causing factors and disaster-bearing bodies. In Section 4, we combine ontology knowledge with domain specific rules to perform semantic reasoning for typhoon disaster prediction. Section 5 is the system architecture we developed. Section 6 is the conclusion.

2. ONTOLOGY REPRESENTATION AND REASONING

Ontologies can be represented in some ontology description languages such as Resource Description Framework (RDF), Web Ontology Language (OWL) and Description logic (DL) [2]. OWL is an extension of RDF and a machine-readable language for sharing and reasoning information on the Web. By reasoning we can derive facts that are not expressed in ontology or in knowledge base explicitly. So inference engine was born. An inference engine, which provides a richer set of mechanisms to work with, is able to infer logical consequences from a set of asserted facts or axioms.

2.1 Web Ontology Language

Web Ontology Language (OWL) [3] is part of the growing stack of W3C recommendations related to the Semantic Web. It is designed for use by applications that need to process the content of information instead of just presenting information to humans. The OWL Web Ontology Language is intended to provide a language that can be used to describe the classes and relations between them that are inherent in Web documents and applications. An OWL ontology in the abstract syntax contains a sequence of annotations, axioms, and facts [4]. OWL ontologies can have a name. Annotations on OWL ontologies can be used to record authorship and other information associated with an ontology, including imports references to other ontologies. The main content of an OWL ontology lies in its axioms and facts, which provide information about classes, properties and individuals in the ontology.

OWL provides three increasingly expressive sublanguages that are designed for use by specific communities of implementers and users: OWL Lite, OWL DL and OWL Full. OWL Lite supports a classification hierarchy and simple constraints. OWL DL supports the maximum expressiveness while retaining computational completeness and decidability, which means that all conclusions are guaranteed to be computable and all computations will finish in finite time. OWL DL contains all OWL language constructs, but they can be only used under certain restrictions. OWL Full is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees.

2.2 Reasoner Introduction

A semantic reasoner is a piece of software able to infer logical consequences from a set of asserted facts or axioms. The notion of a semantic reasoner generalizes that of an inference engine, by providing a richer set of mechanisms to work with. In the existing semantic reasoner and related software [5, 6], Jess, RacerPro, Pellet, FaCT and Jena are widely used. In this paper, Jena was taken for use.

Jena [7] is an open source of Java framework for building Semantic Web applications, which was originally developed by researchers in HP Labs, starting in Bristol, UK, in 2000. It provides an API to extract data and write to RDF graphs. Furthermore, it provides extensive Java libraries for helping developers develop codes that handle RDF, RDFS, RDFa, OWL and SPARQL in line with published W3C recommendations. Jena includes a rule-based inference engine to perform reasoning based on OWL and RDFS ontologies. A variety of storage strategies to store RDF triples in memory or on disk are also included.

How the reasoner operates can be summarized as the following steps. At first, create original ontology model and read the information that the OWL file describes. Then the reasoner registers the created ontology model into the Model Factory and creates a domain-specific rule based inference engine, which will be bound with the ontology that needs to be queried and reasoned. With Ontology API and Model API, we can reason the created model by customized rules [8, 10].

3. CONSTRUCTION OF DOMAIN ONTOLOGY

Due to the different research areas and specific projects, the current ontology modeling methods and standards are quite different. Nevertheless, majority approaches of domain ontology modeling are following the basic principle that Gruber proposed in 1995, which is Clarity, Coherence, Extendibility, Minimal encoding bias and Minimal ontological commitment [9]. The general process of building domain ontology model is as follows. Firstly, determine the scope and object of the domain ontology. Then, choose an ontology

Journal of Theoretical and Applied Information Technology
31st May 2013. Vol. 51 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

language to describe and represent the domain ontology. After that, select an efficient tool for ontology development. Next, make a content analysis and detailed design of the domain ontology. The last step, formalized representation and storage of domain ontology should be taken into account.

In this paper, we took the domain ontology about typhoon disasters into consideration and built the ontology model with Protégé 4.0, which is a free, open source ontology editor and knowledge-based framework, and reasoned the model with Jena. At the same time, on account of its ability of knowledge representation and better reasoning process support; OWL DL is taken as our domain ontology description language.

3.1 Basic Classes of Domain Ontology

According to the experts' suggestions of typhoon disasters and the historical evolution of typhoon, there are three basic classes in the domain ontology of typhoon disasters: disaster-pregnant environment, disaster-causing factors and disaster-bearing bodies.

3.1.1 Disaster-pregnant environment

Generally, a variety of environment can easily induce typhoon disaster [11]. For example, subtropical transitional zone (subtropical" for short), irregular coastline or flared estuary ("coastline" for short), the Taiwan Strait and NE-trending mountains ("strait" for short), impact of the north cold air ("cold-air" for short), mountainous rivers ("mount_river" for short), lower eastward terrain ("terrain" for short), area of crushing geological structure and poor soil water retention ("geological" for short).

3.1.2 Disaster-causing factors

There are several kinds of factors that may cause varieties of disasters under different disaster-pregnant environment, which can be regarded as the subclasses of disaster-causing factors [12]. Such as, geological factors, which is made up by landslide, debris flow and soil erosion; meteorological factors, which typhoon, heavy rainfall and strong winds are belonged to; marine factors, including storm surges, huge waves, marine pollution, seawater encroachment and collapse of seawall; floods factors, the factors that covers the following disaster: floods, dam failure and water logging. Besides, other factors, like biological pest, infrastructure damage, power off, mechanical failure or other indirect disasters, may also cause disasters.

3.1.3 Disaster-bearing bodies

A typhoon makes diverse disaster-bearing bodies suffer from different kinds of causing disasters [13]. The area that is easily affected by typhoon can be divided into several parts: coastal or plain zone which was densely populated or the people's livelihood resources concentrated ("coastal_plain" for short), dams or reservoirs that had lower levels of flood protection or moisture proof ("dam_reservoir" for short), the production base of subtropics economic crops and fruits ("pro_base" for short), coastal fisheries or shipping hub ("fisheries_hub" for short), areas that unreasonable use of the land or economically backward ("backward" for short) and others.

The basic classes and their subclasses are illustrated in Figure 1.

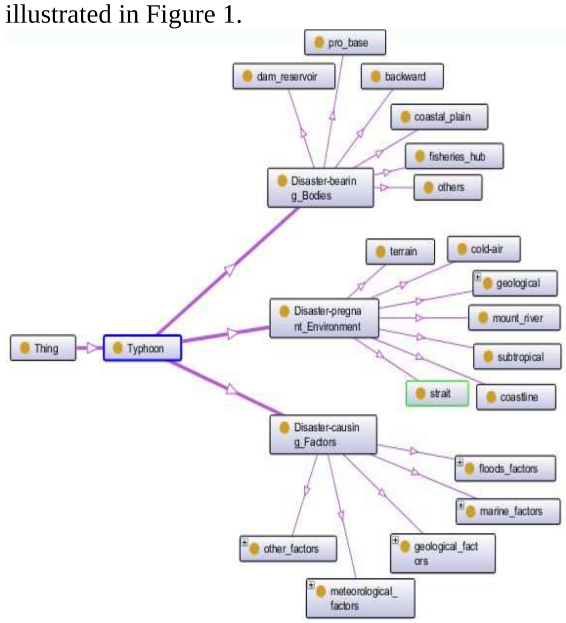

Figure 1: Ontology Model of Main Classes

3.2 Relationship between Classes

Based on the existing related information and law of typhoon activity in recent years, we defined another four kinds of relationships between the basic classes besides the property Subclass: Occur, Bear, Induce and Aggravate. Here comes a detailed description of the relationship.

3.2.1 Relationship Introduction

Property Occur is used to describe the relationship between different disaster-pregnant environment and disaster-causing factors. A certain disaster mainly occurs in one or more disaster-pregnant environment. For example, heavy rainfall mainly takes place in the areas that affected by the north cold air or just the area of NE-trending mountains.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Property Bear is a description of the situation that disaster-bearing bodies suffering from the disasters. A bearing body may suffer from a vast variety of disasters all at once. For instance, the bearing body of dam failure is the area that has lower levels of flood protection or moisture proof.

When some disaster happens, a primary disaster may cause or induce other disasters. In a certain disaster-pregnant environment, the relationship of mutual influence and interdependence between various types of disaster-causing factors is known as the property Induce. Say, floods could induce debris flow under the environment of crushing geological structure and poor soil water retention.

When it comes to Property Aggravate, it is generally used to represent the situation that a variety of disaster-causing factors occurring at the same time or one after another in the same area may aggravate the disaster induced. Take dam failure as an example. It may be aggravated by floods in the area of lower levels of flood protection or moisture proof.

3.2.2 Relationship Representation

As known to all, ObjectProperty in Ontology System has two subclasses: domain and range. While the previous one limits the individuals to the property that can be applied to, the latter has a limit of the individuals in the property they may have as its value. As the relations can be regarded as binary functions or multivariate functions, the classes and their subclasses are treated as domain and range of the function. Take Induce as an example. As a property, Induce is stated to be transitive. It has the domain of a combination of disaster-causing factors and disaster-pregnant environment or their sub-classes, together with the range of disaster-causing factors or its sub-classes. The specific definition of the four relationships is described in Figure 2.

3.3 Domain Ontology Modeling Based on OWL

After the definition of the main classes and relationships between them in the domain ontology of typhoon disaster, we should apply the properties, or relationships, to the defined classes. Say, under the environment of crushing geological structure and poor soil water retention, the meteorological factor heavy rainfall may induce floods, geological factor of debris flow or soil erosion. That's the application of Property Induce.

The description of the application is illustrated in Figure 3.

```
<owl:ObjectProperty rdf:about="#Occur">
  <rdfs:range rdf:resource="#Disaster-pregnant_Environment"/>
  <rdfs:domain rdf:resource="#Disaster-causing_Factors"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#Induce">
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Disaster-causing_Factors"/>
        <owl:Class rdf:about="#Disaster-pregnant_Environment"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
  <rdfs:range rdf:resource="#Disaster-causing_Factors"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#TransitiveProperty"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#Aggravate">
  <rdfs:domain rdf:resource="#Disaster-causing_Factors"/>
  <rdfs:range rdf:resource="#Disaster-causing_Factors"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#Bear">
  <rdfs:domain rdf:resource="#Disaster-bearing_Bodies"/>
  <rdfs:range rdf:resource="#Disaster-causing_Factors"/>
</owl:ObjectProperty>
```

Figure 2: Description of the Four Properties

```
<owl:Class rdf:ID="heavy_rainfall">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#Induce"/>
      </owl:onProperty>
      <owl:someValuesFrom>
        <owl:Class>
          <owl:unionOf rdf:parseType="Collection">
            <owl:Class rdf:about="#soil_erosion"/>
            <owl:Class rdf:about="#landslide"/>
            <owl:Class rdf:ID="floods"/>
          </owl:unionOf>
        </owl:Class>
      </owl:someValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
```

Figure 3: Application of Property Induce

Then set the name, value, type and other constraints of properties in the domain ontology of typhoon disaster, corresponding to Datatype Property in Ontology System.

Next, define individuals or instances of the main classes. So does their properties and values. In this paper, individuals are the typhoons recently took place, including Typhoon Longwang that occurred in 2005 in Fujian Province, Typhoon Morakot in 2009 and Typhoon Muifa in 2011, together with varieties of disasters they had caused.

Thus far, preliminary ontology model of typhoon disaster can be built. To check reasonableness of the model and satisfiability of the concepts, as well revise the unreasonable parts in the model, using one of Protégé own reasoning tools, such as FaCT++, is an necessary step for a better ontology model. Store the modified ontology model in a file or database [14]. In this paper, as the ontology model of typhoon disaster is stored in the format of OWL, the corresponding individuals can be stored in a RDF file.

Beyond that, it is essential to define semantic extensional relationship, such as synonymous, near-synonyms for reasoning. In order to realize the semantic extension query of keywords, ontology model must support the query of original meaning, synonyms, near-synonyms and hyponymy. For

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

example, typhoon has the same meaning with hurricane, cyclone or tropical storm, as well as tropical cyclone. Once any one of these keywords entered, the information about the original meaning of the entered keyword can be displayed, so can its synonyms or near-synonyms.

4. REASONING REALISATION

After building the ontology model, reasoner should be used for excavating the hidden influencing factors or disaster chain information. In this paper, Jena is taken into consideration for semantic reasoning.

4.1 Domain-specific Rules

Jena inference engine is rule-based. It has two internal rule engines: forward chaining reasoning RETE engine and a tabled Datalog engine. Forward chaining, tabled backward chaining and hybrid execution strategies are supported. The two reasoning engines need a set of rules to define their behavior. Jena contains a series of default inference rules for checking the satisfiability of concepts and the relationship between different classes [16], which is aimed at the characteristics of ontology. As an example, the following rules are the definition of the transitive property and disjoint property of the class.

[restriction(?x rdfs: subClassOf ?y)(?y rdfs: subClassOf ?z)→(?x rdfs: subClassOf ?z)]

[restriction(?x owl:disjointWith ?y) (?a rdf: type ?x) (?b rdf:type ?y)→(?a owl:different From ?b)]

These default rules are known as generic rules, which can't meet the requirements of specific information reasoning and retrieval in specific areas. For the accurately definition of the relationships between ontology classes, reasoning rules can be customized and specific inference engine can be created by users. A set of customized rules, called domain-specific rules, is a supplement to the generic rules, but also meet the personalized needs in the field of practical application. The domain-specific rules in the domain ontology of typhoon disaster consist of class relationship rules, instance relationship rules and property relationship rules. Setting class relationship rules as an example, some of the domain-specific rules and their function explanation are listed in Figure 3. Please note that, in Table 1, the prefix "td:" stands for "http://www.semanticweb.org/ontologies/2012/11/Typhoon.owl# ".

Table 1: Part of Domain-specific Rules

Domain-specific Rules	Explanation
[rule1(?a rdfs:subClassOf ?b)(?b td:induce ?c) —>(?a td:induce ?c)]	There are three variables: *a*, *b*, *c*. While *a* is the subclass of *b* and *b* induces *c*, it can be reasoned that *a* can induce *c*.
[rule2(?a rdfs:subClassOf ?b)(?b td:aggravate ?c) —>(?a td:aggravate ?c)]	*a*,*b*,*c* all belong to disaster-causing factors. If *a* is the subclass of *b* and *b* aggravates *c*, it can be reasoned that *a* can aggravate *c*.
[rule3(?a td:induce ?b)(?b td:induce ?c) —>(?a td:induce ?c)] [rule4(?a td:aggravate ?b)(?b td:aggravate ?c) —>(?a td:aggravate ?c)]	Once variable *a* induces variable *b* and *b* induces variable *c*, *a* may induce *c*. So does the relationship of aggravate.
[rule5(?a td:induce ?c)(?b td:induce ?c) —>(all(?a,?b) td:aggravate ?c)]	For variables *a*, *b* and *c*, if *a* induces *c* and *b* induces *c*, *a* and *b* happens simultaneously or consecutively, *c* can be aggravated .
[rule6(?a td:aggravate ?b)(?b td:aggravate ?c) —>(?a td:aggravate ?c)]	As factors *a*, *b* and *c*, if *a* aggravates *b* and *b* can aggravate *c*, *c* can be aggravated by *a*.
[rule7(?a td:induce ?b)(?b td:aggravate ?c) —>(?a td:aggravate ?c)]	As factors *a*, *b* and *c*, if *a* induces *b* and *b* aggravates *c*, *c* can be aggravated by *a*.
[rule8(?a td:induce ?b)(?a td:occur ?d)(?b td:occur ?d) —>(occur(?a,?d) td:aggravate ?b)]	If factor *a* can induce *b* and they both occur in the environment *d*, then *a* may aggravate *b* in the environment *d*.
[rule9 (?a td:induce ?b)(?b td:aggravate ?c)(?d td: bear ?b)→ (bear(?d,?a) td: aggravate ?c)]	On the condition that factor *a* induces factor *b* that aggravates *c* and is born by body *d*, *a* can be born by *d* and aggravate *c*.
[rule10 (?a td:induce ?b)(?b td:induce ?c)(?d td: bear ?c)→ (bear(?d,all(?a,?b)) td: aggravate ?c)]	If variable *a* induces factor *b* and *b* induces factor *c* whose bearing body is *d*, the disaster factor *a* or *b* which is born by *d* may aggravate *c*.

4.2 Domain Ontology Modeling Based on OWL

In terms of the domain-specific rules having been defined above, the reasoning procedure of ontology model can be developed with Jena API or Java integrated development tools (such as Netbeans, Eclipse). For the purpose of unified management, the inference rules that has mentioned above are suggested to be written in a rule file (which can be named "typhoon_dis.rules").

For example, take Typhoon Muifa and those disasters had caused as individuals. As we all know, Typhoon Muifa that took place in 2011 had caused heavy rainfall and strong winds in Liaoning Province. Through the domain-specific rules and the main classes mentioned above, it can be known that heavy rainfall may induce some soil erosion or landslide, especially in a mountainous area and strong winds induce huge waves which may induce. Collapse of seawall usually occurs in the area of irregular coastline or flared estuary. Reasoning with rule 3, rule 6 and rule 8, we can draw a conclusion that coastal city may suffer from the seawater encroachment or heavier collapse of seawall. The area that has unreasonable use of the land in Liaoning may also suffer from soil erosion or landslide. A part of the reasoning procedure is illustrated in Figure 4.

As we all know, Dalian suffers from seawater encroachment so heavily that poses a threat to the local residents in 2011. The reasoning result matches up with the actual caused disaster.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Therefore, the above inference process is reasonable.

```
//Loading ontology model file
OntModel ontModel =
        ModelFactory.createOntologyModel();
ontModel.read("file:F:/yxProtege/Typhoon.owl");
String pre = "http://www.semanticweb.org/
                ontologies/2012/11/Typhoon.owl#";
//Reading  instance
Resource rainfall = m.getResource
                (pre+"rainfall_muifa");
Resource encroachment = m.getResource
                (pre+"encroachment_muifa");
//Loading inference rules
List rules = Rule.rulesFromURL("file:
                F:/yxProtege/typhoon_dis.rules");
GenericRuleReasoner reasoner =  new
        GenericRuleReasoner(rules);
OntModel om = ModelFactory.createOntologyModel
        (OntModelSpec.OWL_MEM,ontModel);
//Building Inference Graph
InfModel inf = ModelFactory.createInfModel(reasoner, om);
StmtIterator si = inf.listStatements
        (rainStorm, null, inwelling);
//Reasonging and Outputing Results
while (si.hasNext()) {
        Statement s = si.nextStatement();
        System.out.println(rainStorm.getLocalName() + " "
                + s.getPredicate().getLocalName()
                + " " +inwelling.getLocalName());
}
```

Figure 4: A Part of Reasoning Procedure

5. SYSTEM ARCHITECTURE

According to the information we have achieved above, the ontology-driven typhoon information system can be built, which was a public platform to provide users to query and analyze information of typhoon disaster. The system is made up by graphical user interface (GUI), ontology-supported platform and query parsing. The Architecture in detail is drawn in Figure 5.

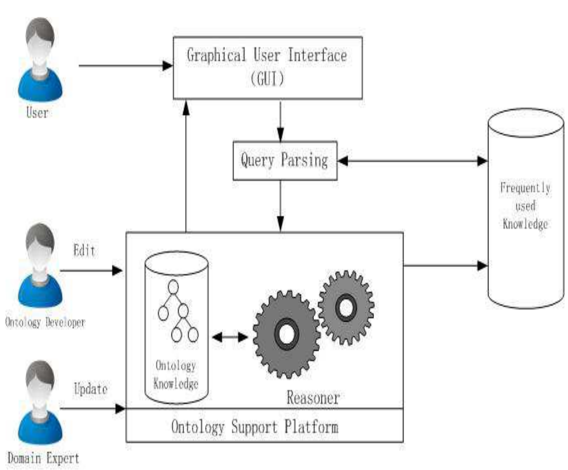

Figure 5: Architecture of Ontology-driven Typhoon Information System

Graphical User Interface is a visual interface for inputting query context and outputting query result. Query parsing analyzes the query context that user have inputted and extracts keywords, in order to look for related information in the ontology-supported platform. Ontology-supported platform has two parts: ontology knowledge and reasoner. Ontology knowledge is used to store the domain ontology of typhoon disaster together with the corresponding instances. Based on default rules and customized rules, reasoner, including Jena and other inference machines, is used for keywords parsing and ontology reasoning. Besides, it is convenient for user to query the same theme many a time if using the frequently-used knowledge, which stores keywords and synonymous or near-synonyms, as well as corresponding result of analysis and reasoning. With frequently-used knowledge, the number of reasoning can be reduced so that query efficiency can be improved and error rate of the system can be reduced.

Once the system is put to use, users can enter a query through GUI. The query can be keywords or simply a sentence. If they are keywords, search them in the frequently used knowledge or ontology knowledge directly. Or else, query parsing analyzes the input sentence to extract keywords. Then the extracted keywords are looked for in the frequently used knowledge library. Once it matched successfully, output the result of analysis and reasoning. If not, the reasoner in the ontology-supported platform is used to analyze and reason keywords in ontology knowledge library. Then output the result and record the number of query. When query times reach the given value, the keywords or sentence together with their reasoning results are all stored in the frequently-used knowledge, which could improve the query efficiency. Along with the analysis and suggestions of domain expert, ontology developers update and improve the system every now and then to offer better service.

6. CONCLUSION

This paper mainly focused on the ontology based approach for disaster prediction and analysis, with setting typhoon disaster as an example. Taking advantage of ontology modeling and reasoning mechanism, a domain ontology model of typhoon disaster presented by OWL is created and ontology-driven disaster prediction and information system can be built. Thus, the problems in lack of semantic-driven and intelligent reasoning can be initially solved and the intelligence level of the natural disasters can be improved in its management and forecast.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

However, due to the research on induced mechanism and influence law of typhoon disaster being not mature, integrity and practicality of domain ontology model and its reasoning mechanism should be further improved. Our model will be also continuously revised and perfected.

ACKNOWLEDGMENTS

This work is partially supported by the National Natural Science Foundation of China (61001197), the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences (SYSKF1010), and the Fundamental Research Funds for the Central Universities.

REFRENCES:

[1] Berners-Lee, T., Hendler, J., Lassila, O., "The Semantic Web", *Scientific American*, Vol. 284, No. 5, 2011, pp. 34-43.

[2] Baader, Franz, Calvanese, Diego, McGuinness, Deborah, Nardi, Daniele, Patel-Schneider, and Peter, "The Description Logic Handbook: Theory Implementation and Applications", *Cambridge University Press*, 2003**.**

[3] Michael K. Smith, Chris Welty, and Deborah L. McGuinness, "OWL Web Ontology Language Semantics and Abstract Syntax", *W3C Recommendation*, 2004.

[4] Peter F. Patel-Schneider, Pat Hayes, and Ian Horrocks, "OWL Ontology Web Language Guide", *W3C Recommendation*, 2004.

[5] Meditskos, G., "A Rule-based Object-oriented OWL Reasoner", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 20, 2008, No. 3, pp. 397-410.

[6] Greco G., Guzzo A., Manco G., and Saccia D., "Mining and Reasoning on Workflows", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, 2005, No. 4, pp. 519-534.

[7] McBride B., "Jena: a Semantic Web Toolkit", *Internet Computing*, Vol. 6, 2002, No. 6, pp. 55-59.

[8] Lozano Tello A, Gomez-Perez A, Kiepuszewski, B., Barros, A.P. OntoMetric, "A Method to Choose the Appropriate Ontology", *Journal of Database Management*, Vol. 15, 2004, No. 2, pp. 1-18.

[9] Gruber T., "Towards Principles for the Design of Ontologies Used for Knowledge Sharing", *International Journal of Human-Computer Studies*, Vol.43, 1995, pp. 907-928.

[10] Valiente, M.-C., Garcia-Barriocanal, E., Sicilia, M.-A., "Applying Ontology-Based Models for Supporting Integrated Software Development and IT Service Management Processes", *IEEE Transactions on Systems, Man, and Cybernetics, vol.*, Vol. 42, January 2012, pp. 61-74.

[11] Motik B, "Reasoning in Description Logics Using Resolution and Deductive Databases", *PhD Thesis*, University of Karlsruhe, Karlsruhe, Germany, January 2006.

[12] Motik B, Sattler U, "A Comparison of Reasoning Techniques for Querying Large Description Logic ABoxes", *Proceedings of the 13th International Conference on Logic for Programming Artificial Intelligence and Reasoning(LPAR 2006)*, November 2006.

[13] Schorlemmer M., Robertson D., "Reasoning about Distributed Knowledge-Transforming Peer Interactions", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 23, 2011, pp. 1419-1431.

[14] d'Aquin, M., Motta, E., Sabou, M., Angeletou, S., Gridinoc, L., Lopez, V., Guidi, D., "Toward a New Generation of Semantic Web Applications", *IEEE Intelligent Systems*, Vol. 23, 2008, No. 3, pp. 20-28.