Sri Lanka Association for Artificial Intelligence (SLAAI)
Proceedings of the fourth Annual Sessions
31st October 2007 - Colombo

Ontology Driven Approach to Disaster Management

A. Thushari Silva & Ajith P. Madurapperuma

Faculty of Information Technology,
University of Moratuwa, Sri Lanka.

E-mail: {thushari,ajith}@itfac.mrt.ac.lk

Abstract

Disasters have become a common threat to humans. Although disasters cannot be evaded, human can manage the disastrous situations to minimize the damage. A successful disaster management system requires information from heterogeneous data sources that belongs to various services. As such, data integration is a crucial issue for disaster management systems. Recent research has recognized ontology as a potential approach to develop solutions for systems involving diversified sources of data. This paper presents an ontology driven approach to data integration in disaster management systems. The proposed ontology enables editing, sharing, converting, searching and querying on data available in various formats in databases, web servers, text files and image files. Protégé ontology development environment has been used to implement the proposed disaster management ontology.

Keywords: Disaster Management System Ontology, Data Integration

1. Introduction

Environmental conditions in the modern world are changing in an irregular manner world wide. This has resulted in the occurrence of natural disasters as well as for the disasters for which humans are responsible. For example, the Sumatran-Andaman earthquake that triggered a Tsunami in 2004, the earthquakes which occurred in Kashmir/Pakistan in 2005, the hurricanes Katrina and Rita in 2005, and the outbreak of SARS – a condition that emerged in southern China in 2003 that spread rapidly throughout the world, are a few that shocked the world unexpectedly.

As such, prevention of disasters and management of disastrous situations have become a theme of common interest. One of the biggest issues in disaster management can be identified as the need for integrating and coordinating many services to handle disastrous situations. For example, disaster management requires considering various services including hospitals, community centres, police, ambulance and funding agencies. Obviously, modern

information and commutation technology can provide potential solutions for integrating these services. However, effective integration of services scattered over heterogeneous hardware and software environments require technologies that can bring all services together. According to modern researchers [3, 4, 11, 9, 20, 10, 18] ontology has been identified as a technology which forms categories of objects (services) which are related to each other. The applications of ontologies have become so challenging, as data related with different services are presented in different formats. As such, ontologies become an artefact which facilitates the communication among heterogeneous sources of information. Therefore, it is appropriate to devise an ontology driven approach to design and implement disaster management systems that can link up associated services together.

This paper proposes an ontology, for the design of disaster management systems that has access to data sources available in different formats including XML, Text, graphics, and databases entries. The disaster management ontology enables editing, sharing, converting, searching and querying on data available in various formats in databases, web servers, text files and image files. Open source software, Protégée has been used to develop the disaster management ontology.

The rest of the paper is organized as follows. Section 2 is a review of the current state of disaster management systems and identifies the need for service integration as a major issue of such systems. Section 3 points out the relevance and the appropriateness of the use of the concept of ontology to integrate services in disaster management systems. Section 4 describes the architecture and functionality of disaster management ontological systems. In the fifth section we discuss disaster management ontological systems can be further improved in order to provide better access and functionality.

2. Overview: Disaster management system

Disasters can occur in forms such as Tsunamis, Hurricanes earth quakes and terrorist's attacks. Unexpected and unavoidable disasters are generally and may create scenarios that are novel. This leads to generate, or gather unstructured or semi structured data at different locations in an unplanned manner resulting in heterogeneous and autonomous data sources. Furthermore, disaster management cannot be an isolated application. It must quickly be aware of information from many sources which are autonomous and heterogeneous. The data and information that we keep in a disaster information management system will have to be timely, reliable, consistent and secure. One of the main observations during the post tsunami relief and rehabilitation stage was that different organizations collect and read data in using a variety of structures, formats and management tools. Thus, integrating such data, which allows isolated data sharing by giving meaningful access to such data, has become crucial due to the heterogeneity of those data sources.

The world looks at a disaster, often willing to help, but needing an accurate picture about the current situation. Possible funders, including the general public, need to see what is necessary. Relief organizations need an accurate on the ground picture so that resources are sent to the right place and are not wasted. Families of those involved need survivor information. For chemical or biological events, scientists across the globe with expertise in a particular field may need to apply themselves to the problem at short notice.

It is evident that, once a disaster occurs, the people who are dealing with disaster management seek accurate real time data. In order to provide such information there should be unique access to those autonomous and heterogeneous data sources. This exploits the concept called data Integration. Semantec heterogeneity makes integration a challenging task and dynamic changing of such sources makes the process further complicated. Once we resolve the semantic heterogeneity, multiple, autonomous data sources will be able to be amalgamated and to provide a unique interface to users of disaster data. As described in the next section, the ontology by its definition has the power of resolving semantic issues.

A few Disaster Management Systems have been developed recently. 'Sahana' [17] is one such disaster management system which is already developed and deployed in different countries as a stand alone application. Sahana Disaster Management System is free and open source software developed using LAMP (Linux-Apache-Mysql-PHP/Postgres) environment. Providing a wide-ranging solution for recovery and rehabilitation is one of the main objectives of the Sahana. This has several modules including missing persons registry, inventory control system, camp management. Idea of integration of data with other sources has not well developed Sahana as well as other tools.

The current disaster management systems provide users with the access to pool of data/information without any structural guidance for navigation through a huge collection of resources. In other words, such systems cannot be used by ordinary persons who are not educated about related services. More importantly, since people are excited and strenuous in disastrous situations, it is rather impractical for them to explore various services consciously. Therefore, a technology that can integrate various services and guide the users to explore services in a distributed environment is a crucial for implementing effective solutions for disaster management. The recent research shows that the concept of Ontology is the best technology for implementing integrated solutions for problem solving in heterogeneous environments [10]. Section 3 gives an overview of ontology.

3. What is Ontology?

People have evolved from procedural thinking to object oriented concepts as a means of modelling the real world. However, object-oriented thinking encourages looking at the world through objects, which are specifically defined and separated from each other. In the recent past, ontology has emerged as an approach to look at the world though categories of objects which are related to each other in some manner. However, ontology is still a relatively new area in computing. In fact, there is no exact definition for ontology, and many views are expressed to define ontologies. There are several working definitions for this valuable term ontology starting from "an explicit specification of a conceptualization" [3, 4]. In the time of writing this paper there are more than seven definitions some time contradictory [15] for ontology [2].

The spirit of ontology is portraying of concept in the world which we live. Most of the scientists in the computer world use the term *class* instead of *concept* when defining ontology. In computing terminology ontology behaves as a controlled vocabulary which consists of classes of objects, classes of relationships among these objects and consists of grammar which has enough capability to specify rules for using vocabulary terms in meaningful manner in the domain of interest.

In spite of some puzzlement about the definition, numerous practical developments in different areas have taken place during the past few years. For example, a system for specifying ontology in the form which is compatible with multiple representation of languages called Ontolingua [4,5] and a framework for finding data in distributed geographic information services [10]. OIL [8] has been developed as the first Ontology representation language to meet W3C standard for Semantic Web and, very recently, and Web Ontology Language (OWL) emerged as the key ontological language for Semantic Web applications [13].

From the database system point of view, ontologies also behave like conceptual database schema. The main functionality of a conceptual database scheme is to provide a logical description of shared data at the logical level, allowing application programs and databases to interoperate without having common sharing structure. While a conceptual schema defines a relation on *data*, ontology defines terms which with to represent *knowledge*. It is evident that there is an astonishing relationship among relational data bases and ontologies. In addition, as highlighted in the previous section disaster data which are in different formats such as relational Databases, XML, and text can be integrated to provide a unique interface by solving problematic level heterogeneity by using ontology. Such ontology is capable to extract the data from those autonomous & heterogeneous data sources and provide intelligent answers based on the requested query.

There are many tools for construction of ontolgies. Some tools are specific to development of ontologies, while others are used for editing, merging, annotating of ontologies. OntoWeb has identified 11 environments, namely, Apollo, LinkFactory, OILEd, OntoEdit, Ontolingua, OntoSaurus, OpenKnoME, Protégé 2000, SymOntoX, WebODE and WebOnto for building or development of ontologies. All these are graphical tools, based on object- oriented concepts and various extensions to XML including RDF, RDFS, OIL and DAML+OIL. Most tools are developed using Java, while some have chosen C++. Further almost all tools provide a library of ontologies. Most tools have some means of working with other Ontology environments, which support interoperability among tools.

Protégé is the latest tool in the ontology development environment [16]. Since Protégé-2000 provides interactive knowledge-base-development environment and ontology-design frame work, the experts can perform knowledge management-task efficiently. It has the capability to handle knowledge bases in multiple languages. Due to this feature, ontology developers can access relevant information quickly whenever they need it, and can use direct manipulation to pilot and manage ontology. Protégé has three types of plug-in libraries; (1) **backends –** provide user with store and import knowledge bases in various formats. Ontlogies in various formats including RDF Schema, XML files with a DTD, and XML Schema files can be imported ; (2) **slot widgets-** which are used to display and edit slot values or their combination in a domain-specific and task-specific ways, and (3) **tab plugins**- which are knowledge-based applications usually tightly linked with Protégé knowledge bases.

Some other systems translate a common ontology into several systems. One such tool, Ontolingua [4,5] was developed with a set of portable ontologies as a means for share and reuse formally represented knowledge. It deals with ontologies represented in predicate logic, frames and relational languages. The ontology of Ontolingua defines forms with classes, relations, functions, objects and theories. Ontolingua ontologies can be translated into different languages including KIF [6], LOOM [14], Epikit [7] and EXPRESS [19]. Using Ontolingua the same ontology can be used for different purposes with the help of KIF, LOOM, Epikit and EXPRESS. For example, LOOM can be used for conceptual design of the ontology and managing the knowledge base of facts and objects. Epikit provides explanatory reasoning. EXPRESS is the standard language for logical database design for sharable data. KIF provides ontologies for sharing data.

Although Ontolingua works with four other systems, it is a relatively bulky product [8] with an essentially narrow range of applications. Until a framework for the development of a generic top-level ontology is created and translation of Ontolingua ontologies into a wide range of ontological systems is enabled, its usage will be very limited. In its present state Ontolingua cannot be used as a general system to support ontology translation and sharing for the Semantic Web application. Nevertheless, the

Ontolingua approach can be considered as a good initiative and the same philosophy can be extended towards the development of a more general approach to translating ontologies across various applications.

4. Ontology for disaster Management

The design of a disaster management ontology is based on the identification of data sources that should be considered and the ontology development environment, which will be used to construct the ontology. Our research has shown that a disaster management system concerns with services such as security service, health service, emergency service and charity service. The study also revealed XML, databases or text files as the main formats of data sources. Therefore, we have decided to develop the disaster management ontology that can work with XML and Databases technologies.

There are number of design goals that should be fulfilled by the proposed ontology, in order to perform operations including editing, sharing, converting, searching and querying on data available in various formats in databases, web servers, text files and image files, and guide users in disaster environment while linking available services in the disaster management. Most important goal is, integrating textual data and multimedia data. It is obvious that, unexpected nature of the disaster creates a lot of data which are unstructured and stored in text format. Integrating textual data is difficult, as objects and relationship among the objects is hard to extract from natural language. To circumvent such difficulties, a component called 'intelligent information extractor' which resolves textual heterogeneity while dealing with reference ontology, is to be developed. The next goal is to sustain flexibility of the disaster management ontology. In other words, once the system was developed it should be extended by integrating other newer data sources in the disaster domain. Then only data integration can be achieved successfully. This challenge has been addressed by providing separate wrappers for each data source. Thus only a new wrapper is needs to be developed in case of integrating with new source. Third goal is providing automatic integration. Since extracted data from the underlying data sources should be available as fast as possible, ontology is developed to support automatic integration.

A survey on tools and environments for ontology development has shown that Protégé is the most widely used open source software for development of ontolgies. Therefore we have decided to use protégé for the design of the disaster management of ontology. As required by the Protégé system, different data formats will be converted to what is called Resource Description Framework (RDF). RDF [1, 12] provides a means for expressing semantics of documents without making any assumptions on the structure of the document.

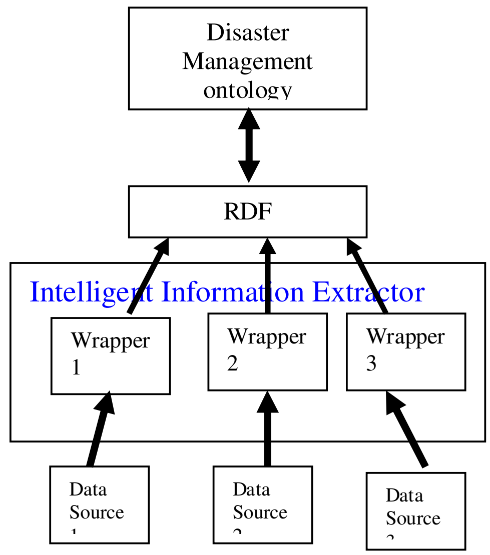

Figure 1: Disaster management ontological system

Figure 1 shows the top level design of the disaster management ontological system. The overall approach can be seen as four-layer architecture with ontology layer, RDF layer, wrapper layer and the data source layer.

A. Ontology layer

Ontology Layer which contains disaster management ontology is the nucleus constituent of the system. Its functionality includes editing, sharing, converting, searching and querying on data available in various formats in databases, web servers, text files and image files. In other words ontology is knowledgeable about the disaster domain including all services and it deals with external data sources and gets real time instances of such services.

The RDF layer construct the RDF data for each and every external source which contains different data formats including XML, relational and text and resolves the semantic heterogeneity among them. Developed ontology has been provided with enough strength to integrate underlying data sources to provide unique access over them and provide an intelligent answer based on such data sources.

Most importantly, Disaster Management ontology has been developed in such a way that a user of this system is guided by the ontology itself for navigating through related services. This, automatic navigation through related services including security service, health service and charity service fulfils the need of assistance for decision making, for any individual in such disastrous situation. For an example consider a child who missed his/her parents wants to find them. It has been noticed that any information about them has not been recorded among the disaster data. In that case child may not know what to do next? In this scenario our ontology can guide him to go for the hospital services before going to the charity service.

Protégé ontology development environment has been used to implement the proposed disaster management ontology. The protégé back end plug-in allows knowledge bases to be in different formats. Using this feature heterogeneous data sources can be plugged with developed ontology after resolving heterogeneity among those sources.

B. RDF Layer

This layer consists of RDF form for each data source which has been generated by the wrappers. The main functionality of this layer is to resolve Semantic heterogeneity in the extracted RDF formats. The translated ontology may differ from each other and may not share the same vocabulary even for referring to the same entities. For example, in case of searching for a person may use the term person in its ontology, while dealing with the donations, it may use the term person to refer to individual who has donated. This simple example shows that there is a need for translation or merging between ontologies, before making any attempt at machine interpretation of meaning. Consequently ontology merging operation is to be performed in this layer.

C. Wrappers

Wrappers are fundamentally parsers which extract the data from different sources and resolvs syntactic differences. In this layer, source-specific wrappers have been developed to transform each and every data source which contain data with different formats, into common RDF. Source-specific wrappers are developed to provide the flexibility for integration of other data sources as well. Integration approach has been developed by establishing a joint data format all over the underline data sources. As for the semi-structured data format, RDF has been chosen. The key point of our approach for integration is we convert all other different data formats into RDF data using reference ontology, and only performs queries locally to the resulting data stored in the central repository without going again to the data source.

D. Data source layer

Data sources are coming from heterogeneous systems of hardware and software. Mainly we are considering data sources available in different formats including XML, Text, graphics, and databases entries. Thus disaster data in such sources may be structured or semi structured. In case of unstructured or semi structured refinement over underlying data sources are to be performed.

5. Discussion

We have been involved in the development of disaster management ontology. Once the wrappers are constructed they will be integrated with disaster management ontology. Then this ontology can be used to query about disaster management information coming from various data sources for which wrappers are developed. It should be noted that this project is more than writing wrappers and converting data sources to RDF. More importantly, we have constructed the ontology in such a way that a user of this system is guided by the ontology itself for navigating through related services. This feature address the issue that disaster management ontology developed for a country cannot be used by another country due the differences in the services available.

This ontology has many advantages it can provide intelligent answers for any query. For instance suppose a user want to find a person who went missing during a disaster and Information shows that there is no particular information about that missed person. Then the developed ontology provides guidance about what he should do next while automatic connection to corresponding services. This ontology can be used by any user who does not have very good knowledge in IT. We deliver an ontology running on a web server and can be accessed by PC.

Further work of this project has many directions. Firstly, this ontology can be extended to enable drawing from new data source just by developing a wrapper for the particular data source. However, the above data formats are very common in use and they are adequate for extracting data from many sources.

Secondly, from maintenance view point, the overall system will be improved to edit, merge, share, and search through expert system. From this end, if we merge with an ontology which describes Geographical Information Systems we will be able provide an answer for queries including geographical information.

Thirdly, we will enhance the power of developed ontology while enable accessing of the system by using mobile devices. This exploits any person from any place at any time can access this developed ontology to get any information.

5. References

[1] Frank, M. and Eric, M. 2004. "RDF Primer", W3C Recommendation, 10th February 2004, http://www.w3.org/TR/2004/REC-rdf-primer-20040210/

[2] Guarino, N, "Understanding, building and using ontologies", International Journal of Human -Computer Studies, vol. 46, pp. 293-310, 1997.

[3] Gruber T.R, "Towards principles for the design of ontologies used for knowledge sharing", International Journal of Human-Computer Studies, vol. 43, pp. 199-220, 1995.

[4] Gruber T. R, "A Translation Approach to portable Ontology Specification", Knowledge Acquisition, pp 9-220, Apr 1993.

[5] Gruber T.R, "Ontolingua: A mechanism to support portable ontologies", Technical Report KSL pp 91-96, 1992

[6] Genesereth, M. R. and Fikes, R. E.. "Knowledge Interchange Format, Version 3.0 Reference Manual", Technical Report, Computer Science Department, Stanford University Logic-92-11, 1992.

[7] Genesereth, M. R. "The Epikit Manual", Epistemic, Inc. Palo Alto, CA,1990.

[8] I, Horrocks, D. Fensel, J. Broekatra, S. Decker, M. Erdmann, C. Goble, F. Van Hermelen, M.Klein, S. Staab, R. Studer, and Motta, "The Ontology Inference layer OIL",Technical Report,2000.

[9] Jin Y., Decker S., Widerhold Gio, "OntoWebber: Model-Driven Ontology-Based Web Site Management", Semantic Web and Database, 529-54, 2000.

[10] Klien E., Mmichael L, Werner K, "Ontology-Based Discovery of Geographic Information Services", unpublished.

[11] Lenat, D., Miller, G. and Yokoi, T.. Cyc, "WordNet, and EDR", Communications of the ACM, Vol. 38, pp 45-49, 1995.

[12] Lassila, O. and Swick, R.. "Resource Description Framework (RDF)", W3C Recommendation, 1999 http://www.w3.org/TR/WD-rdf-syntax/

[13] McGuinness, D. L. ,van Harmelen, F. "*OWL Web Ontology Language Overview"*, W3C Recommendation 10th February 2004.

[14] MacGregor R. "The Evolving Technology of Classification-based Knowledge Representation Systems, in: Principles of Semantic Networks: Explorations in the Representation of Knowledge", edited by Sowa, J.F., San Mateo, CA: Morgan Kaufmann,1991

[15] Noy N. F. and Deborah L. M, "Ontology Development 101: A Guide to Creating Your First Ontology", Technical Report,2001, http://www.bibsonomy.org/bibtex/208759c07a29e371220cddb39a26a6200/fparreiras.htm

[16] Noy, N F, Fergoson, R. W. and Musen, M. A., 'The Knowledge model of Protégé 2000: combining interoperability and flexibility", Stanford medical Informatics Technical Report, Stanford university, 2000.

[17] Rachid L., Careem M., Silva C., Silva D,, Weerawarna S., "Sahana: Overview of a Disaster Management System", International Conference on Information and Automation, pp. 15-17, 2006.

[18] Shamsfard, M. and Barforoush, A. A. "Learning ontologies from natural language texts", International Journal of Human-Computer Studies, Vol 60, pp 17-63, 2004.

[19] Spiby, P., "EXPRESS Language Reference Manual", International Standard ISO 10303-11, Project Three ("Express Language")

Working Group Five ("STEP Development Methods") of Subcommittee Four ("Manufacturing languages and Data") of ISO Technical Committee 184 ("Industrial Automated Systems and Integration"); International Standards Organization. 1991.

[20] Warren, D. H. D. and Pereira, F. C. N., "An efficient easily adaptable system for interpreting natural language queries", Computational Linguistics, Vol. 8, pp 110-122, 1982.