INTEGRATION OF GEOGRAPHIC INFORMATION SERVICES FOR DISASTER MANAGEMENT

G. W. Lan [a, *], X. Q. Zhou [b], Q. Y. Huang [c]

[a] School of Geodesy & Geomatics, Wuhan University, Wuhan,430072, P. R. China, gwlan_770728@163.com
[b] Center for Public Safety Research, Tsinghua University, Beijing, 100084, P. R. China, qyhuang@tsinghua.edu.cn
[c] Key Laboratory of Geo-Informatics of State Bureau of Surveying and Mapping, Chinese Academy of Surveying & Mapping, Beijing,100039, P. R. China, smartmouse@21cn.com

Commission VI, WG IV/5

KEY WORDS: Internet GIS, Services, Metadata, Computer Vision, Interoperability, Geographical Services, Visualization

ABSTRACT:

In recent years the OpenGIS technology standards have been developed by the Open Geospatial Consortium (OGC), which provide the essential basis for syntactic interoperability of geographic information services, also gives an opportunity for data interoperability, data integration and data sharing between different emergency management agencies. However finding suitable services and visualization of geospatial information served over the Internet for decision makers is still a crucial task. Therefore this paper proposes a service-integration architecture to facilitate the discovery of geospatial information services (GI services), and provide a friendly user-interface for visualization of OGC-compliant services. Based on the proposed architecture, a prototype system has been developed to support catalogue service for information sources related to disaster management, including GI services. And OpenLayers, an open source javascript library, is used to access to GI services. The paper concludes with a discussion of the problems encountered and a brief view of our future work.

1. INTRODUCTION

A disaster is defined as a serious disruption of the functioning of a community or a society causing widespread human, material, economic or environmental losses that exceed the ability of the affected community or society to cope using its own resources. Disasters interrupt the society by claiming lives, creating victims and destroying infrastructures and houses. When a disaster occurs, funds and budgets that have been assigned for development purposes are diverted to respond to that disaster and returning quality of life to normal. Disasters also have negative impacts on the environment as they affect natural resources. Therefore, considering society, economy and environment as the three main components of sustainable development, disasters have a negative impact on sustainable development which making appropriate management of disaster a necessity.

The experiences of disaster management activities, particularly responding to the attacks on the World Trade Center and the Pentagon on September 11, 2001 in USA have proven that spatial data can considerably facilitate disaster management as most of the required information for disaster management has a spatial component. It is estimated that 70–80% of information is resolvable to geographic location, therefore the nature and characteristics of geographic information (GI), and the way in which it is used, is paramount in managing crises effectively. Therefore spatial data and related technologies have proven to be crucial for effective collaborative decision-making in disaster management. However, current studies show that although spatial data can facilitate disaster management, there are substantial problems with collection, access, dissemination and usage of required spatial data for disaster management. Such problems become more serious in the disaster response phase with its dynamic and time-sensitive nature.

In China, large amounts of geographic data are gathered and distributed in different governmental departments and each one holds the latest information respectively for their special usages. However these datasets are not available on the network and interactions among agencies or emergency corps usually occur on a personal/phone/fax basis. This leads to limited interaction and slowness in response time, contrary to the nature of the need for information access in an emergency situation. It is suggested that agencies involved in disaster management should collaborate, share data and information in time of peace and in situation of emergency.

In this paper we propose a software architecture that supports data integration and data sharing in disaster management domain. OGC-compliant GI services are employed to support sharing and integration of geographical information between those agencies. Integration of geographic information services can be divided into two steps: (1) Finding suitable services to solve problems at hand; (2) Access to the service by a web client.We propose the ontologies for disaster management to facilitate service discovery, then we introduce an architecture for service integration and its components in detail.

2. RELATED WORKS

Due to the ever increasing use of the Internet, there is a trend to store geographic information in physically distributed database systems and disseminate spatial data over the Internet [5]. Information technology (IT) standards, such as eXtensible Markup Language (XML), Simple Object Access Protocol (SOAP), and Web Services Description Language (WSDL), are

* Corresponding author: G. W. Lan, email: gwlan_770728@163.com

utilized by GIS vendors to support the deployment of geographic web services. OGC has released several OpenGIS Implementation Specifications that serve as interface standards of geographic information (GI) services. This lays down the foundation for building interoperable and system independent spatial data infrastructures (SDI). Data integration and data sharing for disaster management also benefit from technical advancement in the information technology and particularly the geographic information science.

OGC is driven by its members, and thus represents the GIS software industry, user groups, and academia. The major GIS vendors, such as ESRI and MapInfo, as well as many open source projects (for instance GeoServer) developed software compliant with its specifications supporting Web mapping or Web cartography on the Internet. The OGC specifications have been successfully integrated into several geospatial data infrastructures and serve as a base for realizing a prototype of a European geoportal as an important step for implementing the INSPIRE initiative.

There are a lot of tools that can be used to access to the OGC-compliant services, such as ESRI ArcMap9.X, MapInfo8, SIS Map Brower and so on. In the open source community, tools such as WorldWind can do the same job. But we should not expect decision makers to be GIS experts. Access to those services by a web browser may be more suitable for them. Software developers can use APIs such as CarbonTools (http://www.TheCarbonProject.com), OpenLayers to develop applications access to OGC-compliant services.

On the other hand, in the open and distributed environment, the services that are to appropriate for solving problems at hands are often not previously known. Therefore discovering the suitable services from among a large number of available services is a central task within the GI web services domain. In order to find an appropriate service the requirements resulting from the requester's question have to be matched against descriptions of the available service implementations. Two types of semantic heterogeneity can lead to problems if performs a simple keyword-based search (E Klien et al., 2004 and Michael Lutz et al.,2003): (1) Naming heterogeneity, the same real world facts are understood in the same way but are named differently; (2) Cognitive heterogeneity: Because of different perspectives on the same real world facts there may not be a common base of definitions of the underlying facts between two disciplines. The examples of such problems had been described in detail by Michael Lutz et al., which can't be overcome by keyword-based search.

E Klien et al. have developed BUSTER system (http://www.semantic-translation.de/) that provides an ontology-based approach with logical reasoning on metadata for retrieving information sources. Following this idea we develop the ontologies for disaster management, and use OpenLayers as a tool to develop our web-based user-interfaces.

3. ONTOLOGY-BASED APPROACH FOR SERVICE DISCOVERY

Within the disaster management domain decision-makers have to make decisions under time pressure, they want the answer to their question, the whole answer and nothing but the answer! Anything else complicates the picture and adsorbs precious time. Therefore information management in this domain should be greatly enhanced to satisfy their requirements. Generally metadata ("data about data") is used to describe unambiguously information resources thus enhance information retrieval, but this improvement depends greatly on the quality of metadata content. One way to enforce the quality of these metadata is the use of a selected terminology for some metadata fields in the form of lexical ontologies, allowing not only to describe the contents but also to reason about them. As service discovery is concerned, these ontologies should facilitate the classification of resources and information retrieval.

3.1 Ontologies for Disaster Management

Ontology has widely employed for overcoming semantic heterogeneity in information search, data sharing and integration. The term ontology can be defined as "a structured, limitative collection of unambiguously defined concepts". This definition contains four elements: (1)An ontology is a collection of concepts; (2) The concepts are to be unambiguously defined; (3) The collection is limitative and then concepts not in the ontology cannot be used; (4) The collection has structure, which means that the ontology contains relationships between the concepts. Similarly Audi defined ontology as 'the study of explaining reality by breaking it down into concepts, relations and rules and share it with others' (Wei Xu & Sisi Zlatanova, 2007). While disaster management just attracts common concern in China in recent years, we still lack such an ontology in this domain.

Wei Xu & Sisi Zlatanova considered that ontologies for disaster management have to consist of data ontology and organizational ontology. As in disaster management community, decision-makers are inclined to search for required information relevant to occurring accidents/disasters, so we propose accident ontology for this purpose.

Data Ontology

Data ontology consists of formal concepts that describe the datasets needed for disaster management, e.g., topological datasets, utility datasets, cadastre datasets, hydrographical datasets, risk sources, protected targets, monitoring stations, emergency medical services, population distribution, road networks, hospitals, fire stations, forest coverage and so on. Some standard datasets are maintained by appropriate federal agencies and made available so that there is no need for other organizations to develop comparable datasets. These datasets are collected and kept up-to-date before the occurrence of a disaster (relatively "static"), including: (1) Geodetic control; (2) Orthoimagery; (3) Elevation; (4) Transportation; (4) Hydrography; (5) Governmental units; (6) Cadastral information (land ownership boundaries).While others may be regularly collected and updated after the occurrence of a disaster in the aftermath of emergency situations (so-called "dynamic" data such as damaged areas, closed roads and burning areas). Generally speaking, decision-makers are more interested in "dynamic" data (E.g., one will make decision on the closed roads, but not a whole road network of a city.), semantic heterogeneity existing here should be overcome by the data ontology. On the other hand, ISO19115 provides an agreed-upon conceptualization of geographic entities, covering location dimension, temporal dimension and content dimension, thus enables access to data and services at metadata level. Therefore data ontology must be developed by referring to this standard.

Accident Ontology

In accident ontology concepts are defined to describe hierarchical structure of "accidents" (e.g., natural disasters including floods, droughts, typhoons, earthquakes, landslides and so on.) — what causes the accident, where is the affected area, the approach to handle it. In China, disasters are generally classified into four categories, i.e., natural disasters, industrial accidents, society security issues, public health issues. So far there are about 170 concepts defined in this ontology. Figure 1 illustrates a brief overview of the structure of the accident ontology. In Figure 1, most of concepts are not presented for simplicity. By accident ontology, decision-makers can find a certain service related to an accident they are interested in.

Organisational Ontology

Organisational ontology consists of formal concepts that describe the structure of organisation (e.g., the fire brigade, the police, the municipality, the Ministry, and etc.)—how the organisation is structured, what are the responsibilities of each user within this organisation, how the users communicate with each other, how the work is carried out within the organisation, which user needs what data and so on. Such as "static" data, are collected by mapping agencies, while "dynamic" data are generally collected by emergency responders, these differences should be reflected in the organizational ontology.

Those ontologies are represented in OWL and can be of help in service discovery. Data ontology is used to describe data served by those services, while organizational ontology can be served to describe the profile of service providers; accident ontology, to describe a related accident that a service is relevant to.

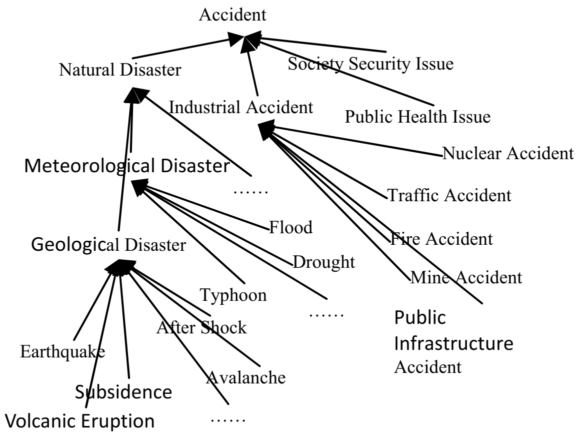

Figure 1. Structure of accident ontology

3.2 Ontology-Based Query

GI service discovery involves the identification of service descriptions that match a service request. The matchmaking process can be done by subsumption test, i.e., determining whether one description subsumes another one. Formally, subsumption can be defined as follows: in a terminology T containing concepts C and D, C is subsumed by D if in every model of T the set denoted by C is a subset of the set denoted by D (Donini, 2003). Donini proposed a structural subsumption algorithm as follows:

Let

$A_1 \sqcap A_2 \sqcap \ldots A_m \sqcap \forall R_1 \bullet C_1 \sqcap \forall R_2 \bullet C_2 \sqcap \ldots \sqcap \forall R_n \bullet C_n$

be the normal form of the FL_0-concept description C, and

$B_1 \sqcap B_2 \sqcap \ldots B_k \sqcap \forall S_1 \bullet D_1 \sqcap \forall S_2 \bullet D_2 \sqcap \ldots \sqcap \forall S_n \bullet D_l$

be the normal form of the FL0-concept description D. Then C subsumes D iff the following two conditions hold:

(1) For all i; $1 \leq i \leq k$, there exists j; $1 \leq j \leq m$ such that $B_i = A_j$.

(2) For all i; $1 \leq i \leq l$, there exists j; $1 \leq j \leq n$ such that $S_i = R_j$ and C_j subsumes D_i.

During the matchmaking process, a requesting service descriptor is matched with each of the advertised service descriptors. Let "R" denote the requesting service descriptors and "A", the advertised service descriptors. The match can basically has five result types:

Exact if R and A are equivalent concepts, formally $R \equiv A$.

PlugIn if R is a subconcept of A, formally $R \sqsubseteq A$, e.g, "nuclear accident" to "industrial accident".

Subsume if R is a superconcept of A, formally $A \sqsubseteq R$.

Intersection if the intersection of R and A is satisfiable, formally $\neg(A \sqcap R \sqsubseteq \bot)$,e.g, a house to an office(if in the ontology, the concepts of house and office are not disjoint).

Disjoint, if the intersection of R and A is not satisfiable.

4. ARCHITECTURE FOR INTEGRATION OF GI SERVICES

In this section we illustrate a three-tier architecture for integration of GI service, based on which a prototype system has been developed. By this architecture, the service/information providers can register their services to catalogue registry, in a situation of emergency, decision-makers can get necessary information from the provided services. As depicted in Figure 2, the proposed architecture consists of a resource manager tier, application tier and presentation tier. We first introduce the components in the architecture, then illustrate how they interact during service registration and the service discovery and present a prototypical implementation.

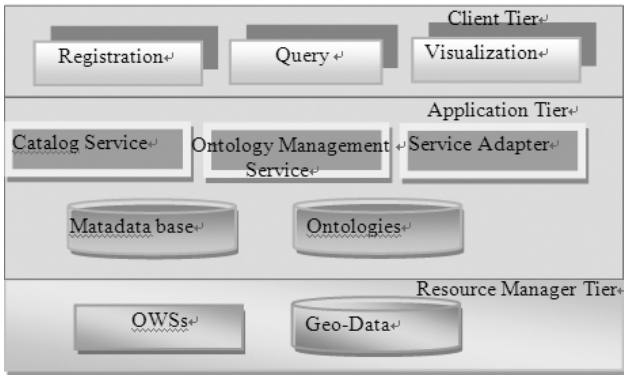

Figure 2. Architecture for integration of GI service

4.1 Service Descriptions

A critical element of the use of services in distributed computing environments is the service description. Whether such a description is available directly from a service, such as with OWS services, or from a 3^{rd} party in the case of most WSDL descriptions, service information is the key both to discovering and consuming useful services in any service-oriented architecture.

As for OGC-compliant GI services, many of the metadata structures are common, based on the ISO 19115 international standard for geographic information metadata, and are provided by the common operation GetCapabilities. The response from a GetCapabilities request is an XML description of the service's information content and supported request parameters, and is therefore both machine- and human-readable. This capabilities document conforms to an XML schema, partly unique for the particular type of service, which allows clients to validate the response. The capabilities document consists of information of the following sections (Arliss Whiteside,2005).

ServiceIdentification: Metadata about this specific server. The schema of this section shall be the same for all OWSs.

ServiceProvider: Metadata about the organization operating this server. The schema of this section shall be the same for all OWSs.
OperationsMetadata: Metadata about the operations specified by this service and implemented by this server, including the URLs for operation requests. The basic contents and organization of this section shall be the same for all OWSs, but individual services can add elements and/or change the optionality of optional elements.
Contents: Metadata about the data served by this server. The schema of this section is specific to each OWS type, as defined by that Implementation Specification. Whenever applicable, this section shall contain a set of dataset descriptions, which should each be based on the MD_DataIdentification class specified in ISO 19115 and used in ISO 19119.
Generally speaking, the *ServiceIdentification* and *OperationsMetadata*, taken together, provide enough information for an client to make use of a service, once found; while the sections of *Contents* and *ServiceProvider* provide the information needed for an client to discover a service. As for emergency response situation, decision makers often present their requirements in the following aspects: (1) Which organization provides this service; (2) What information it serves; (3) What kind of accident or what a certain accident the served information is related to. We enrich the service description by the ontologies mentioned in section 2, to facilitate service discovery for users from disaster management domain. To sum up, we make use of the conventional metadata and additional metadata that is collected by registration process.

4.2 Components

As depicted in Figure 2, the resource manager tier consists of services and data that locate at the providers' side. The application tier contains business logic of the system, accessed by corresponding components in the presentation tier. The presentation tier provides components supporting registration, query and visualization of services. The application tier contains business logic of the system and accessed by corresponding components in the presentation tier, including the following components: (1) Metadata base and (2) Ontologies that are implemented in the Web Ontology Language (OWL), (3) Catalogue Service; (4) Ontology Management Service that based on Jena API; (5) Service Adapter.

Resource Manager Tier
The resource manager tier consists of distributed data repositories with different types of data and storage systems from different disaster management agencies and mapping agencies. Access to those information sources is wrapped by standardized interfaces of the OGC's Web service specifications.

Presentation Tier
The presentation tier consists of interfaces of registration, query and visualization. The registration interface allows users to fill in the metadata about their services and register them to the service catalogue. The query interface includes tools for browsing lists of services, full-text searching, and filtering lists by predefined criteria. Visualization interface supports visualization of geographical content served by a certain selected service. Generally decision-makers are not interested in GI as an end in itself but as a means to an end in making the right decision based on retrieved information. They are seldom GIS experts therefore a user-friendly mapping interface is absolutely necessary here.

Catalogue Service
The catalogue service enables the registration and querying of service descriptions in the metadata base. It organizes GI services into categories according to the proposed ontologies (see in section 2). By the registration process, metadata of a service is formally structured and stored in the metadata base with the help of the ontology management service. On the other hand the system provides query tool to give users an opportunity to present their requests based on the above-mentioned ontologies. When a request for specified services is received, this component searches in the metadata database for matched services and return to the query interface. The catalogue service retrieves the relevant concepts in the ontologies from the ontology management service to infer matching degree of the request and the service description.

Ontology Management Service
Ontology management service is developed based on Jena API to manage and provide access to the ontologies described in section 2. The Jena2 persistent storage subsystem implements an extension of the Model class that provides transparent persistence for models through the use of a database engine, currently supporting MySQL, Oracle, PostgreSQL and Microsoft SQL server, on both Linux and WindowsXP. Therefore we select MySQL to store the ontologies presented in section 2. The functionalities of this component includes (1) queries for the available classes in the ontologies; (2) translates client request into formal concepts; (3) find the sub- or superconcepts of a given concept in the ontologies; (4)interact with the catalogue service to support service registration and discovery.

Service Adapter
As there are several types of GI services and they are accessed in different ways by OpenLayers API, so the service adapter is designed to control access to selected services. The service adapter fetches parameters needed to access the selected service from metadata base or by invoking GetCapabilities operation of this service. For example, access to a WMS service consists of the following steps:
(1) The WMS service allows clients to receive descriptions of layers, coordinate reference systems (CRS), output formats and display styles supported by the service (GetCapabilities).The capabilities are described in an XML format.
(2) Clients can request available layers in supported CRS, display styles and output formats using CGI-style parameter specifications (GetMap). While vector representations of the resulting map are allowed, WMS implementations usually provide image data (e.g., GIF, JPEG, PNG) to the client, which can be displayed on the clients system in combination with other map data.
(3) Conforming WMS implementation may optionally allow the client to request feature information for a specified location (GetFeatureInfo).
It's the tasks of the service adapter to fetches necessary parameters according to the type of a service by parsing the capabilities document from the provider's site, or get them from the metadata base. The last two steps can be done by invoking the corresponding functions of OpenLayers API.

4.3 Service Registration and Discovery

The architecture supports two tasks: registering and discovering services in the presentation tier. The detailed workflow for these tasks is described as following.

Service Registration

As depicted in Figure 3, both requesters and providers use concepts from existing ontologies to formulate their requirements or service descriptions, respectively. This may be impossible over the Internet (perhaps the main reason that the idea of semantic web is unfeasible), while in a specific domain such as disaster management, with related policies by the government, this can be feasible.

The workflow for service registration is depicted as following. Suppose that the provider has a service that provides some information about an occurrent flood. He starts the registration with choosing his domains of interest ("accident" here), and the Registration interface retrieves the available vocabulary for each of the chosen domains (subconcepts of "accident") from the Ontology Management Service. The provider selects the appropriate vocabulary to describe his service, here he selects "flood". Then the Registration interface gives him options to fill in some properties of the flood. He also uses data ontology to describe what information his service provides. Of course the URL identifying the service is necessary. Once he has completed the registration, the semantic description presented and the Capabilities document are stored in the metadata base. The component of catalogue service invokes the GetCapabilities operation to get detailed information about this service. For example, a WMS service identified by a base URL "http://sms.webmap.cn/scripts/openserv.exe?map=/sms_ogc/sms1500.map&", its GetCapabilities operation can be invoked by sending a request "http://sms.webmap.cn/scripts/openserv.exe?map=/sms_ogc/sms1500.map&SERVICE=WMS&REQUEST=GetCapabilities" to get its capabilities document. This document is parsed and then stored.

A registered service may be updated, e.g., the information it serves, thus its metadata is changed. The system automatically collects metadata of registered services periodically. Therefore an updated service needn't to be registered again.

Service Discovery

Once the service about the flood has been registered in the catalogue, it can be found via the query interface. The query interface provides requestors an opportunity to present their requirements in a heuristic and formal way. Similar to the provider, the requestor first chooses her domain of interest and selects the "natural disaster" as a basis for her semantic query. The query interface presents her subconcepts of "natural disaster". She can also further refine the query criteria by using subconcepts of "natural disaster" and/or adding additional constraints. Once she finishes setting of her requirements, her request is formalized and sent to the catalogue service. The catalogue service retrieves the "suitable" service by using matchmaking algorithm based on TBox and ABox reasoning (Li, L., & Horrocks, I., 2003, Gao Shu et.al, 2007).

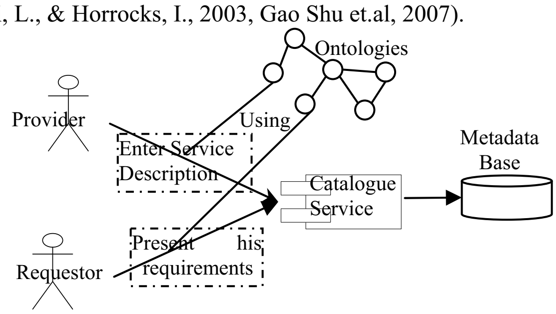

Figure 3. Registration and Discovery

The system also support simple query, here concepts in those ontologies are served as a thesaurus. When the requestor enters her request in the input box, the Ontology Management Service parses the input and tries to find concepts of the requestor, for example if she inputs "Meteorological Disaster", the service about the flood will be a match.

It is assumed that by using formal descriptions of semantics and automatic matchmaking algorithms semantic problems such as those described in section 2 can be avoided to some degree. However, problems listed as follows, can occur.

− No match. Services that fit the requester's requirements are not found at all because the matchmaking algorithm is too rigorous. A threshold value has to be specified by the requester indicating which degree of similarity between advertisements and requirements is still acceptable.

− Unsuitable match. Services that are found do not fit the requester's requirements. This, too, can be caused by the calibration of the matchmaking algorithm. Here, the matchmaking algorithm is too tolerant because the threshold value is too low. Another possible reason is that the query criteria set by the requestor does not correctly reflect her requirements or the semantic annotations do not correctly reflect the providers' conceptualization of the service.

4.4 Access to GI service by OpenLayers

OpenLayers is an open-source client-side JavaScript/AJAX framework for overlaying various mapping services (OpenLayers API can be downloaded http://openlayers.org/). It supports various mapping APIs such as Google, Yahoo, Microsoft Virtual Earth, OGC WMS, OGC WFS, KaMap, Text layers, and Markers to name a few. The nice thing about it being a pure client-side implementation is that you can drive it with any server language such as ASP.NET, PHP, PERL and for simple maps, embed directly into a plain html file. There is minimal requirement from the web server if you are using publicly available or subscription layers. Access to GI services by OpenLayers API can be following steps:

(1) Adding the necessary script references. First for any 3rd party mapping services you will be using, you need to include the libraries in addition to the OpenLayers library file and these should be included before your OpenLayers include.

```
<script src='http://dev.virtualearth.net/mapcontrol/v3/mapcontrol.js'></script>
<script src="http://openlayers.org/api/OpenLayers.js"></script>
```

(2)Creating the OpenLayers map Object. That is to create a blank div with id=map that has the dimensions we want and position that where we want on the page.

```
<div id="map" style="width: 400px; height: 400px"></div>
```

(3)Adding layers. Write the javascript code to create the map and load into our div and add the layers. In this example we access to a WMS server identified by the URL "http://giswebservices.massgis.state.ma.us/geoserver/wms".

```
var map = new OpenLayers.Map(document.getElementById("map"));
wmstaxi = new OpenLayers.Layer.WMS("Boston Taxi Stops",
"http://giswebservices.massgis.state.ma.us/geoserver/wms",
 {layers: "massgis:GISDATA.WATERTAXISTOPS_PT", transparent:
"true", format: "image/gif"},{tileSize: new
OpenLayers.Size(400,400), buffer: 1 });
map.addLayer(wmstaxi);
```

In the above code you may see that we need to specify the parameters such as "layers" directly. While before we select a service, we can't determine the values of such parameters, so does the visualization client. The Service Adapter is designed to set values of those parameters dynamically.

There are two ways to fetch the parameters: (1)Invoke the GetCapabilities operation by sending an AJAX request to the service, then get the capabilities document and parse it; (2) Get those parameters from metadata base. The advantage of the first method is that after the requestor selects a service, the browser can interact directly with the service, so that reduce load of the server which catalogue service hosts. The disadvantage is that a lot of code is necessary to handle the response from GetCapabilities operation to support different browsers. And this XML document may be differently encoded, additional code is also needed. By the second method, we don’t have to write the above code, so it is selected in our prototype implementation.

5. CONCLUSIONS

An architecture for semantic-based integration of geographic information service is proposed in this paper and a prototype system has been developed. We employ ontologies for disaster management domain for service description, which enables the service providers to describe the service semantics more accurately in a standardized form, thus improve accuracy of service discovery. In order to provide users with a friendly interface to access to OGC-compliant services, we use OpenLayers API as a tool to develop the web-based user-interface.

Our experiment has showed that OpenLayers API can be easily used to access distributed GI services. But as service discovery is concerned, there is still a long way to go. The main drawback of the approach could be seen in the matchmaking, which is generally related to high computational complexity. The reasoning task has only been tested with ontologies presented. In an open environment with complex shared vocabularies and also more concepts to reason on, could make the task of match-making during service discovery extremely slow. Li and Horrocks (2003) have shown that while the average time (per registry of a service) for classifying a TBox indeed increases rapidly with the size of the registry, matchmaking in an already classified TBox is extremely fast.

Worth-mentioned is that although we have proposed our thinking of ontologies for disaster management, those domain ontologies should be built with help from experts from this domain.

REFERENCES

Michael Lutz, Catharina Riedemann, 2003. A Classification Framework for Approaches to Achieving Semantic Interoperability between GI Web Services, In: *Lecture Notes in Computer Science, Conference on Spatial Information Theory: Foundations of Geographic Information Science (COSIT 2003)*. Springer, Berlin, Germany, pp. 186－203

Li, L., & Horrocks, I., 2003. A software framework for matchmaking based on semantic web technology.*In Twelfth international World Wide Web conference,* Budapest, Hungary.

Arliss Whiteside, 2005. *OpenGIS® Web Services Common Specification (OGC 05-008)*, Open Geospatial Consortium Inc., pp.10-32.

Xu,W., Zlatanova, S., 2007. Ontologies for disaster management, In: *Lecture notes in geoinfomation and cartography, Geomatics solutions for disaster management*, Duolunduo, Canada, pp. 185-199.

E. Klien, M. Lutz, W. Kuhn, 2004. Ontology-based discovery of geographic information services: An application in disaster management, Computers, environment and urban systems, 30, pp.102-123.

Gao Shu, Omer F. Rana, 2007. Ontology-based semantic matchmaking approach, *Advances in Engineering Software,* 38, pp.59-67.

Massimo Paolucci, Takahiro Kawamura, 2002. Semantic matching of web service capabilities. In: *1st international semantic web conference (ISWC2002)*, Sardinia, Italy.

Donini, F. M., 2003. *The description logic handbook: Theory, implementation and applications*, Cambridge University Press, Cambridge, pp. 96-136.