KIDGS: A Geographical Knowledge-informed Digital Gazetteer Service

Yu Liu, Runqiang Li, Kaichen Chen, Yihong Yuan, Lingli Huang, Hao Yu
Institute of Remote Sensing and Geographical Information Systems, Peking University
Beijing, China
liuyu@urban.pku.edu.cn

Abstract—**This paper describes the design and implementation of a geographical knowledge-informed digital gazetteer service, KIDGS. It is a standard web service that provides unified XML-based access interfaces for various applications. At present, many digital gazetteer systems are implemented directly based on relational databases. In other words, place name entries are managed by data tables. However, several components in geographical knowledge, such as category and relation information, should be explicitly represented to support various queries. In KIDGS, we adopt OWL and Protégé to model the conceptual level of knowledge, and use PostGIS to store place names. This article also introduces an application developed based on KIDGS.**

Keywords- Geographical knowledge; Digital gazetteer; Web Services; Ontology

I. INTRODUCTION

With the development of World Wide Web (WWW), unstructured or semi-structured documents that contain place names have been recognized as one important component of modern geographical information [1]. In order to deal with textually represented geographical knowledge, we should establish and maintain the relations between place names and their properties, especially geographical coordinates. Such an objective can be achieved by the development of digital gazetteers. A digital gazetteer manages a collection of place names [2,3,4]. It provides a foundation of applications that deal with textual geographical knowledge, such as geographical information retrieval (GIR) [4,5]. Generally, a record in digital gazetteers manages the name, footprint, type, and alias, of a geographical feature. At present, a number of digital gazetteers have been developed, such as the Alexandria Digital Library (ADL) gazetteer[1] and the Getty Thesaurus of Geographical Names[2] (TGN). The gazetteer content standard (GCS) and feature type thesaurus[3] (FTT) developed by ADL project contribute significantly to the development of digital gazetteers, while TGN focuses on place names related to art and architecture.

A digital gazetteer can be viewed as the foundation of a geographical knowledge base. Additionally, a geographical knowledge base should consider semantic characteristics of a feature type and the relations among places. Unfortunately, a relational structured digital gazetteer fails to manage well some aspects of knowledge, such as constraints and relations. Since the web ontology language (OWL) provides a convenient approach to formally managing knowledge, in this research, we proposed a solution to this challenge. It adopts relational database management systems (RDBMS) to store the information about places and uses OWL to manage concept level knowledge. Following this solution, a service that supports queries about place names and the associated knowledge is developed.

II. CONCEPTUAL DESIGN

A. Requirements

Generally, a digital gazetteer can deal with "where is …?" queries, which resulting the location or footprint of a geographical feature, and "what is there?" queries, which return all the geographical features satisfying the given constraints. As stated in [3], the "minimum required elements" of entries in digital gazetteers include names, footprints, and types. However, spatial relations should also be taken into account to establish structures in a geospatial knowledge base. For a knowledge-informed digital gazetteer service (KIDGS), we can identify the following four aspects of requirements.

First, in addition to traditional place names (e.g. Beijing), many textual strings are associated with particular locations in geographical space. For example, aliases, postal codes, phone numbers, and IP addresses can serve as descriptions of locations. Some phrases, such as "north of Beijing" and "the airport in Beijing" [7], are also used to represent places. Ideally, a KIDGS should deal with all these situations. Second, the information about types of places is an important component of geographical knowledge [8,9]. In many query statements about a digital gazetteer, types are specified. Types of places can be managed by a hierarchical structure, or a tree, where we can generally obtain one super-type and at least one sub-type for a given type. In KIDGS, such a structure can be used to manipulate the following two query strategies when a type is specified, namely, retrieving sub-types and similar types of the provided type. Third, in the "what is there?", the term "there" stands for not only quantitatively expressed locations, but also spatial assertions used to describe localities, such as the phrase "inside Beijing". Such a spatial assertion consists of at least one spatial relation and reference object. Finally,

[1] http://www.alexandria.ucsb.edu
[2] http://www.getty.edu/research/conducting_research/vocabularies/tgn
[3] http://www.alexandria.ucsb.edu/gazetteer /FeatureTypes/FTT_metadata.htm

uncertainty is an inherent feature of place names. It may be due to ambiguity, vagueness, imperfection, etc. In terms of the spatial extent of a place, the boundary is often indeterminate [10]. Hence, the footprint of a place in digital gazetteers is often managed using a generalized representation, since we generally consider the approximate location of a place rather than its precise location in communicating geographical knowledge.

B. System architecture

From the perspectives of software development, the KIDGS should provide a unified programming interface. Hence, in this research, it is implemented to be a Web Service, and XML-based access interfaces are provided for various applications. The conceptual system architecture is depicted in Fig. 1.

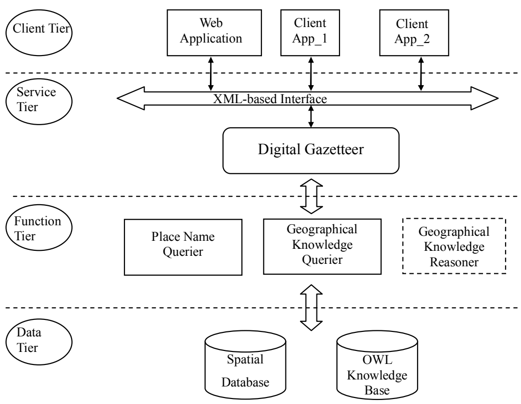

Figure 1. System architecture of KIDGS

With the objective of extensibility, the structure of KIDGS is composed of four tiers and one XML-based interface for accessing the service. From bottom to top, the four tiers are data tier, function tier, service tier, and client tier.

1) Data tier: In KIDGS, the ontological model represents commonsense geographical knowledge in KIDGS, such as the relations among places and types, dimensionality and vagueness of various types [11,12]. For efficiency, we use OWL to manage the concept based knowledge, such as the fact that city is a sub-type of settlement, while employ RDBMS to store the instance based knowledge, such as properties of places and relations among places. The footprints of places are also stored in relational tables.

2) Function tier: The function tier establishes a bridge between the data and the exposed Web Service interfaces. It includes a set of different function modules, namely, *Place Name Querier* (PNQ), *Geographical Knowledge Querier* (GKQ), and *Geographical Knowledge Reasoner* (GKR). PNQ access DBMS directly and obtain the properties of a geographical feature according to its place name. GKQ can execute the knowledge incorporated queries, which must take into account the information from both the conceptual level knowledge and individual level knowledge. Finally, GKR can manipulate existing knowledge and deduce new knowledge following particular rules. Note, in Fig. 2, that its dashed border of implies that this component is open can be extended to accomplish user-defined reasoning functions.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="Query">
<xs:complexType>
<xs:sequence><xs:group ref="Condition"/></xs:sequence>
</xs:complexType>
</xs:element>
<xs:group name="Condition">
<xs:choice>
<xs:element ref="Name"/>
<xs:element ref="Predicate"/>
<xs:element ref="Type"/>
<xs:element ref="AND"/>
<xs:element ref="OR"/>
<xs:element ref="NOT"/>
</xs:choice></xs:group>
<xs:element name="Name" type="xs:string"/>
<xs:element name="Type" type="xs:string"/>
<xs:element name="Predicate">
<xs:complexType>
<xs:sequence>
<xs:element name="Reference" type="xs:string"/>
<xs:element name="Relation" type="relationtype"/>
</xs:sequence></xs:complexType></xs:element>
<xs:element name="AND">
<xs:complexType>
<xs:group ref="Condition" minOccurs="2"/>
</xs:complexType></xs:element>
<xs:element name="OR">
<xs:complexType>
<xs:group ref="Condition" minOccurs="2"/>
</xs:complexType></xs:element>
<xs:element name="NOT">
<xs:complexType>
<xs:group ref="Condition"/>
</xs:complexType></xs:element>
<xs:simpleType name="relationtype">
<xs:restriction base="xs:string">
<xs:enumeration value="within"/>
<xs:enumeration value="contain"/>
<xs:enumeration value="owerlap"/>
            ......
</xs:restriction></xs:simpleType></xs:schema>
```

Figure 2. XML Schema of queries provided to KIDGS

3) Service tier: The KIDGS is released as a web service in at this tier. It is in charge of receiving client's requests, invoking relevant function modules and returning request results. One important point is that the service tier encapsulates implementation details and complicated function interface inside the KIDGS, and provides clients with a unified and friendly invoking interface.

4) Client tier: KIDGS can be accessed through the XML-based interface from various applications in different platforms. Such applications are depicted in the client tier. In order to demonstrate the functions provided by KIDGS, we developed a Web application that allows end users to send various query requests to KIDGS.

C. XML-based Access interface

All applications submit queries to KIDGS and obtain results. These two directions of interaction are achieved through an XML-based interface. The interface is designed to be flexible so that interactions between various applications and the gazetteer are in a unified XML form. An XML-based interface consists of two parts: the request represented by a *QueryXML* document, and the service response described by a *ResultsetXML* or *ErrorXML* document.

The requests sent to KIDGS can be defined as a triple <name, type, predicate>. For example, in the query "obtain the cities named London inside Canada", the three elements, i.e., name, type, and predicate, correspond to "London", "City", and "inside Canada", respectively. Hence, by specifying the contents of the triple, users could express various queries. Each *QueryXML* document expresses one query. Fig. 2 indicates the schema of *QueryXML* documents.

Element <Name> represents the name of the target geographical feature. Besides conventional places names, we can use an IP address like "162.105.0.1" as a name directly. Additionally, regular expression and wild-card characters are available when query condition is not clear. Element <Type> restricts the type of the target places. Since taxonomy is an important component in commonsense knowledge, users can obtain more precise results by providing type category information. In KIDGS, we direction use the FTT of ADL as the whole set of available types. Element <Predicate> constrains the query results by defining the relation between the target geographical feature and the reference geographical feature. It is composed by an element <Reference>, which stands for the reference geographical feature a geometrical entity with geographical coordinates, and an element <Relation>, which stands for the relation between them.

The condition elements <Name>, <Type>, and <Predicate> describe the content of the query. However, we may encounter more complex queries like "obtain all fast-food restaurants named McDonald's inside Beijing or Shanghai". To express such complex requests using *QueryXML*, the schema provides three logical operators: <AND>, <OR>, and <NOT>, which describe the logic relations among the query condition elements so that the *QueryXML* data are flexible in various applications. Following the schema of *QueryXML*, an example query of "what are the fast-food restaurants named McDonald's inside Beijing or Shanghai?" is listed in Fig. 3.

```
<?xml version="1.0" encoding="UTF-8"?>
<Query><AND><Name>McDonald's</Name>
<Type>fast-food restaurants</Type>
<OR>
<Predicate>
<Reference>Beijing</Reference>
<Relation>within</Relation>
</Predicate>
<Predicate>
<Reference>Shanghai</Reference>
<Relation>within</Relation>
</Predicate></OR></AND></Query>
```

Figure 3. XML document for query "what are the fast-food restaurants named McDonald's inside Beijing or Shanghai"

When the query was successfully performed, the KIDGS would return a *ResultsetXML* document, the schema of which is depicted in Fig. 4.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="Resultset">
<xs:complexType><xs:sequence>
<xs:element ref="Result" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence></xs:complexType></xs:element>
<xs:element name="Result">
<xs:complexType><xs:sequence>
<xs:element name="Placename" type="xs:string"/>
<xs:element name="Location" type="xs:string"/>
<xs:element name="Footprint" type="xs:string"/>
<xs:element name="MatchingDegree" type="degreetype"/>
</xs:sequence></xs:complexType></xs:element>
<xs:simpleType name="degreetype">
<xs:restriction base="xs:decimal">
<xs:minExclusive value="0"/>
<xs:maxExclusive value="1"/>
<xs:fractionDigits value="1"/>
</xs:restriction></xs:simpleType></xs:schema>
```

Figure 4. XML Schema of query results created by KIDGS

The root element <Resultset> represents the whole results set, which is composed by an arbitrary number of element <Result>. The number of <Result> element depends on the query and the quantity of the gazetteer data. Each <Result> has the same structure, representing one place name returned from KIDGS.

Element <Result> contains three mandatory elements, namely <Placename>, <Description>, and <Footprint>. <Placename> is the place name of the result, and it also can be the GPN. <Description> provides a textual phrase describing the locality of the result. It may be more helpful when the place name is a phone number or an IP address. <Footprint> adopts the well-known text (WKT) representation [13] to represent the footprint of a general place. Two optional element, <Type> and <MatchingDegree>, indicate the type and goodness of a resulting place according to the query. For instance, the XML document in Fig.5 answers "Where is Beijing?".

```
<?xml version="1.0" encoding="UTF-8"?>
<Resultset> <Result>
    <Placename>Beijing</Placename>
    <Location>Beijing</Location>
    <Footprint>POINT(116.3916 39.9059)</Footprint>
    <Type>City</Type>
    <MatchingDegree>1</MatchingDegree>
</Result></Resultset>
```

Figure 5. Resulting XML document of the query "Where is Beijing?"

When a query requirement is not executed successfully, an *ErrorXML* document, which contains some information about the error, will send back to the end users.

III. KEY TECHNIQUES AND IMPLEMENTATION

A. *Representing geospatial knowledge*

In the implementation of KIDGS, geographical knowledge is represented by OWL. The data described by OWL ontology are interpreted as a set of "individuals" and a set of "property assertions" which relate these individuals to each other.

1) Categories of places: Following [3], there are both similarities and differences between place and geographical feature. In KIDGS, however, since most entries directly represent geographical features, we use features types for the categorization system of places, with additionally specifying a number of types, such as IP address and telephone number, which are outside the range of traditional geographical features.

For many places names, such as "Beijing City", the type information (i.e., city) is explicitly represented by the corresponding type postfix (i.e., "City"). Depending on various contexts, the type postfix of a place name may be in presence or in absence. For example, in the sentence "Let us meet at Beijing", the type postfix is omitted. However, almost all audiences of such a sentence know the exact type of the place. We take into account type postfixes in implementing a digital gazetteer for the following two reasons. 1) The type of a queried place name can be determined according to the given type postfix, even the type information is not provided in the *QueryXML* document. 2) It is equally probable for users to provide queries with or without postfixes, whereas the place names stored in DBMS might be deficient to satisfy all the inquiry circumstances. In order to deal with the above two cases, the postfix is stored in Protégé as a value property for each class, while a field indicating whether place names stored with postfix or not for a certain table is appended to the metadata table.

2) Relations: Relations among places are divided into spatial relations and aspatial relations. In digital gazetteers, spatial relations between two places are implicitly represented, that is, they are computed according to footprints when necessary. However, there are several disadantages of this strategy. 1) Since most spatial relation calculations are time-consuming, and will decrease the efficiency in executing queries. 2) some special kinds of spatial relations, such as "sibling", could hardly be obtained using ordinary geometrical computations. Hence, KIDGS provides a mechanism to explicitly represent spatial relations, especially part-whole relations. A query like "Is Maine State a sibling of California State?" can thus be supported, since both states have the "inside" relation to the United States. Additionally, the "whole place" can be used to disambiguate identical place names.

a) Spatial relations: The common types of binary spatial relations are topological, directional, and metric. Locality descriptions consisting of place names and spatial relations are widely used to make qualitative or semi-quantitative predicates about geographical features. In KIDGS, we explicitly manage a hierarchy of spatial relations used in representing geographical knowledge (Fig. 6). Some issues should be taken into account when managing spatial relations.

First, in terms of localities with directional relations, the directions might be external or internal to the reference object. External directional relations can be viewed as a refinement of the topological relation "disjoint", while internal directional relations refine the topological relation "within" [14]. Second, a metric relation can also be viewed as a refinement of "disjoint", and it may be expressed quantitatively (e.g., "50 kilometers to Beijing") or qualitatively (e.g., "near Beijing"). In qualitative cases, a series of distinctions may be used, such as "close", "far". Finally, directional and metric relations are generally too rough to make a specific determination of a locality. Thus, they are usually combined to form a more precise description in practice.

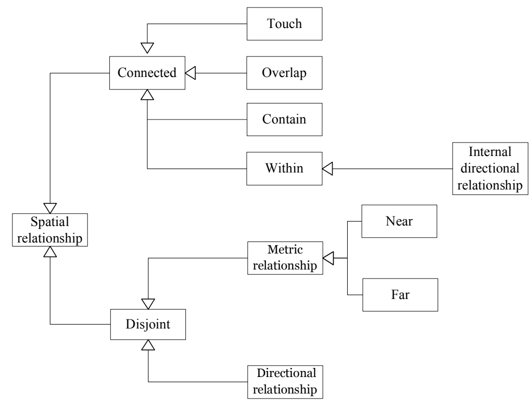

Figure 6. Spatial relations used in KIDGS

b) Aspatial relations: In addition to ordinary spatial relations that can be geometrically measured, a number of aspatial relations should also be considered in a geographical knowledge base. Generally, an aspatial relation can be modeled by a spatial relation with particular geographical semantics. For example, the relation "flows into" is related to the type "channel", while the relation "is capital of" corresponds to the types "city" and "administrative unit". Moreover, several aspatial relations do not take into account spatial attributions. For instance, the relation "greater population than" between two settlements only considers the population properties of involved objects.

c) Relations among features, place names, and footprints: For a digital gazetteer, it is essential to definitely define relations among features, place names, and footprints. IKIDGS, we find the following four basic relations: M (is name of) and M^{-1} (has name) are a pair of relations between place names and features, while G and G^{-1} exist between place names and footprints. These four relations are all many-to-many relations. Two places may share the same name and a place may have a number of aliases. Meanwhile, two places can have identical footprint and a place may be represented by various geometries due to different generalization levels.

The category and relation information provides an explicate framework for representing geographical knowledge. Based on the ontological model, several functions are provided by KIDGS to access the model. For example, given a feature type, such as river, we can use the function *getDirectParentClass* (String Collection <OWLNamedClass>) to get its direct parent class, i.e., channel.

B. Storage schema of geographical data

The ontological model of common sense geographic world provides a guidance of physical geographical data storage. In KIDGS, most place names are associated with geometric locations as footprints. Hence, in order to increase efficiency, detailed information on places is collected and managed in a PostGIS database, an extension of PostgreSQL to support spatial data storage and operations.

In PostGIS, each data table stores information of one kind of named feature. Two metadata tables containing required knowledge information are created to describe the type properties of each table and related field. It also links the PostGIS database to the OWL ontology in protégé. Metadata table I registers the mapping relations between tables and feature classes. For a table, the following descriptive information is provided in metadata table II: 1) the column that stores general place names, 2) whether the place names have postfixes, 3) the column that manages locations, and 4) the representation approach of footprints. In KIDGS, two approaches, coordinate-based geometries and textual descriptions, are available for representing the footprint of a place. Hence, it is necessary to distinguish these two situations in the metadata table.

C. Query execution strategies

1) Managing addresses and aliases: An address provides an encoding representation a certain place on the earth. In KIDGS, besides ordinary mail addresses, IP addresses, postal codes, and telephone numbers are also managed. We can adopt geo-coding techniques to deal with ordinary addresses. In terms of other types of addresses, their locations of such addresses are often represented by textual strings instead of quantitative coordinates. For instance, when searching "100871" in the table "Chinese postal code", the result will be "Peking University" other than a quantitative representation. An additonal query is necessary to obtain the geographical coordinate.

In practice, we often use several names for the same place. For example, "Peking" is an alias of "Beijing". Hence, aliases should be taken into account in digital gazetteers. In KIDGS, we store the mapping relation between an alias and the associated primary name in a special table. When an alias is mentioned in a query statement, KIDGS first obtain the authoritative name according to the mapping relations, and then rewrite the statement to be a normal one so that the query can be performed.

2) Dealing with types: When users query place names with a certain type, KIDGS can easily determine which table should be queried, since metadata table I is designed for providing the type information of each table. Note that we should access all tables if no type information is provided by the users. However, it is also possible that the queried type is not a leaf node in the geographical feature type tree. In this case, KIDGS uses the ontological model of geographic knowledge to get all subtypes of the provided type so that a complet search can be performed to obtain all records with these subtypes.

Moreover, postfixes provide an effective implication of categorize place names. In the ontological model, the postfix for each type is defined. In practice, the postfix word may be absent in both records of gazetteers and user-inputted query statements. For example, "Beijing City" is stored in the database, while "Beijing" may be used in users' query statements. Hence, we cannot adopt complete matching to search the resulting records. If type information is provided, then we can search in particular tables and check whether the differential word is consistent with the given type information. For example, if the queried named is "Beijing" and the required type is city, then the records like "Beijing River" do not satisfy the query condition, since the differential word, i.e., "River", is not consistent with the required type. On the other hand, if the type postfix is specified, then KIDGS can generally determine the type of target places. Unfortunately, this strategy may fail when a place name contains two (or more) type postfixes. For example, there is a city named "Huang Shan (Mountain)". If the users input "Huang Shan (Mount)" without type information, and the system views "Shan (Mount)" as a type postfix, then the record "Huang Shan City" cannot be obtained.

3) Manipulating spatial relations: As mentioned earlier, spatial relations can be used to specify the range of target objects. Most binary spatial relations are directly supported in PostGIS, and can thus be expediently used in SQL query statements.

However, the representation of vague spatial relations requires development of both computational techniques to manage vagueness and empirical research to identify how people perceive spatial relations [15]. In this research, several kinds of vague spatial relations are implemented to extend spatial operation library of PostGIS, such as near and far. Take "near" as an example to illustrate how it works. In different contexts, the relation "near" may correspond to various quantitative distances [16], e.g., the predicate "near a city" implies a greater distance than the predicate "near a hotel". In KIDGS, according to commonsense knowledge and a number of surveys, we determine the numeric distance for each type. For example, the distances are 1km, 50km, and 200km for types of building, city, and Chinese province. Note that the type of a locality also influences nearness, and we will investigate this issue in the future.

For explicitly expressed part-whole spatial relations, relevant operations, e.g., deciding whether Maine State and California State are siblings, can be accomplished. It is indispensable to determine whether their direct parent object is the same place, that is, the United States. Moreover, neighboring relations between two child districts could be used to infer the relation between their parent places. For instance, it is doubtless that Hebei province and Shandong province are touched given that two counties, one inside Hebei and one inside Shandong, are border upon.

4) Calculating matching degrees: As argued in [4,17], several aspects, including names and types, ought to be taken into account when matching two places. In KIDGS, both factors are considered to calculate the matching degree between the resulting records and the query condition. Additionally, when a vague relation, such as near, is used to specify the spatial range of a place, the matching degree computation should also consider the vagueness of the relation. In the implementation, we compute the similarities associated with the three aspects, i.e., names, types, and relations. These three metrics are normalize to [0,1], and their average value is adopted to be the final matching degree of a place name.

IV. CASE STUDY – A WEB APPLICATION OF KIDGS

In this research, we implemented a java web application to illustrate the use of KIDGS. This web application supports submitting the query of place name and viewing the footprint in the text form, both in client browser. It is developed based on the Apache Struts framework[4] adopting model-view-controller (MVC) architecture. KIDGS can be called as a model in the server. Fig. 7 illustrates a calling process between the web application and KIDGIS. From the perspective of the web server, six steps are involved in the process as follows: 1) listening and accepting the request from the client browser (Fig. 8a); 2) translating the contents of page to a *QueryXML* document; 3) sending the *QueryXML* document to the KIDGS; 4) receiving the query results in a *ResultsXML* document; 5) extracting the contents of the document in need; and 6) creating a new JSP page and updating the client browser (Fig. 8b). The web application indicates that the XML based interfaces of KIDGS can be easily invoked to build various client applications.

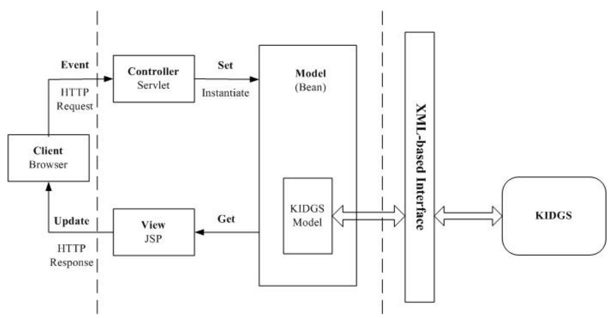

Figure 7. Architecture of the web application of KIDGS

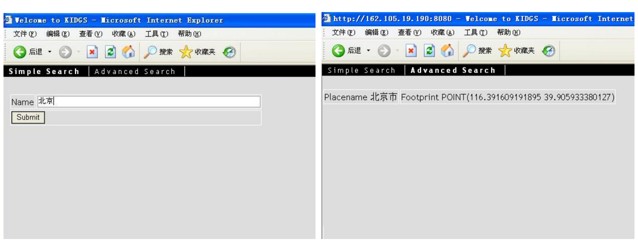

Figure 8. User interface of the web application of KIDGS, a) interface for inputting query, b) interface for displaying resulting data

V. CONCLUDING REMARKS

In order to deal with textual geographical knowledge, a digital gazetteer service should be involved. In this paper, we introduce the conceptual design and implementation of a digital gazetteer service, where the representation of geographical knowledge is emphasized on. Several aspects, including type, relation, and vagueness, are considered in KIDGS. The service is designed to be a standard web service that provides XML-based access interface. It contains four tires, i.e., data tier, function tier, service tier, and client tier, from bottom to top. In the data tier, we use OWL to model the conceptual level of knowledge and employ ORDBMS to manage the individual level of knowledge. The function tier can be easily extended to involve more features, such as spatial reasoning. The service is written in Java, and a number of open source libraries, such as Lucene, Jena[5], and JTS[6,] are reused.

[4] http://struts.apache.org/

[5] http://jena.sourceforge.net/.

[6] http://www.vividsolutions.com/jts/jtshome.htm.

REFERENCES

[1] Y. Liu, Q. Guo, J. Wieczorek, M.F. Goodchild, "Positioning localities based on spatial assertions," International Journal of Geographical Information Science, in press.

[2] L.L. Hill, J. Frew and Q. Zheng, "Geographic names: the implementation of a gazetteer in a georeferenced digital library," D-Lib Magazine. vol. 5, 1999. available online at: http://www.dlib.org/dlib/january99/hill/01hill.html.

[3] M.F. Goodchild and L.L. Hill, "Introduction to digital gazetteer research," International Journal of Geographical Information Science. vol. 22, pp. 1039–1044, 2008.

[4] J. T. Hastings, "Automated conflation of digital gazetteer data," International Journal of Geographical Information Science. vol. 22, pp. 1109–1127, 2008.

[5] C.B. Jones, and R.S. Purves, "Geographical information retrieval," International Journal of Geographical Information Science. vol. 22(3), pp. 219–228, 2008.

[6] Y. Zhang, Y. Gao, L. Xue, S. Shen and K. Chen, "A common sense geographic knowledge base for GIR," Science in China Series E: Technological Sciences. vol. 51(S), pp. 26–37, 2008.

[7] B. Bennett and P. Agarwal, "Semantic categories underlying the meaning of 'place'," in COSIT 2007, S. Winter, M. Duckham, L. Kulik and B. Kuipers, eds. Berlin: Springer, 2007, pp. 78–95.

[8] J.L. Mennis, D.J. Peuquet and L. Qian, "A conceptual framework for incorporating cognitive principles into geographical database representation," International Journal of Geographical Information Science. vol. 14(6),pp. 501–520, 2000.

[9] A. Skupin, "The world of geography: Visualizing a knowledge domain with cartographic means," PNAS. vol. 101(S1): 5274–5278, 2004.

[10] S. Hwang, J.-C. Thill, "Modeling localities with fuzzy sets and GIS," in Fuzzy Modeling with Spatial Information for Geographic Problem, M. Cobb, F. Petry and V. Robinson, Eds. Berlin: Springer, 2005, pp. 71–104.

[11] Y. Liu, Y. Zhang, Y. Tian and L. Xue, "On generalized place names and the associated ontology," Geography and Geo-information science. vol. 23(6), pp. 1–7, 2007.

[12] C. B. Jones, H. Alani, and D. Tudhope, "Geographical information retrieval with ontologies of place," in COSIT 2001, D.R. Montello, ed, 2001, pp. 322–335.

[13] OGC, OpenGIS Simple Features Specification For SQL Revision 1.1, Open GIS Consortium, 1999.

[14] Y. Liu, X. Wang, X. Jin, L. Wu, "On internal cardinal direction relations," in COSIT 2005, A.G. Cohn and D.M. Mark, Eds. Berlin: Springer, 2005, pp. 283–299.

[15] M.F. Worboys, "Nearness relations in environmental space," International Journal of Geographical Information Science. Vol. 15(7), pp. 633–651, 2001.

[16] E. Clementini, P. Di Felice and D. Hernández, "Qualitative representation of positional information," Artificial Intelligence. vol. 95, pp. 317–356, 1997.

[17] G. Fu, C.B. Jones and A.I. Abdelmoty, "Building a geographical ontology for intelligent spatial search on the Web," in: The IASTED International Conference on Databases and Applications, DBA 2005. 2005, pp. 167–172.