Location-aware Web Service by Utilizing Web Contents Including Location Information

YongUk Kim[1], Chulbum Ahn[1], Joonwoo Lee[1] and Yunmook Nah[1]

[1] Department of Computer Science and Engineering, Dankook University,
126 Jukjeon-dong, Suji-gu, Yongin-si, Gyeonggi-do, 448-701, Korea
{yukim, ahn555, jwlee}@dblab.dankook.ac.kr, ymnah@dankook.ac.kr

Abstract. Traditional search engines are usually based on keyword-based retrievals, where location information is simply treated as text data, thus resulting in incorrect search results and low degree of user satisfaction. In this paper, we propose a location-aware Web Service system, which adds location information to web contents, usually consisting of text and multimedia information. For this purpose, we describe the system architecture to enable such services, explain how to extend web browsers, and propose the web container and the web search engine. The proposed methods can be implemented on top of traditional Web Service layers. The web contents which include location information can use their location information as a parameter during search process and therefore they can increase the degree of search correctness by using actual location information instead of simple keywords.

Keywords: Location-Based Service, Web Service, GeoRSS, search engine, GeoWEB

1 Introduction

With the rapid development of related technologies, Web has become an important medium to provide and share information. Various groups from different industries produce and use web contents. Especially, the means of information providing and sharing have been enlarged from large scale Web Service providers into small scale and even individual providers by Web 2.0 technology [1]. The Web now deals with contents such as detail information closely related with casual life of ordinary people as well as traditional contents related with professional research and commercial promotion. The volume of information is much larger than the data volume handled by any other media. We usually produce and consume web contents related with our normal lives while moving continuously in our real life. The production and consumption of information related with specific locations are also ever increasing. It is now very common to search for restaurants by typing 'the most favorite Spaghetti restaurant near Kangnam subway station' on the keyword box of search engines, which then make search engines to check for user blogs and return appropriate results. But, such location information is usually treated as simple text and such retrievals related with positional information are usually processed by only text matching

techniques, resulting in incorrect search results. Search engines usually return huge volume of unnecessary sites because they simply contain the same keyword with the given positional information.

In this paper, we propose how to extend web contents so that their own location information can be built in them. The proposed method prevents location information from being treated as simple text and allows the exact web contents having strong relationship with the given location can only be retrieved. Here, the web contents mean the general extension of HTML/CSS contents [2]. The proposed system can collect location information from extended web contents and search and provide web contents related with the given query location. To support and utilize location-aware web contents, we propose how to extend web browsers and how to build web containers and web search engines.

The remainder of this paper is organized as follows. The common problems of current search systems are described in Section 2. Section 3 shows the overall structure of location-aware Web Service system and describes detail structures and behaviors of the proposed system. Section 4 describes some implementation details and experimental results. Finally, section 5 concludes the paper.

2 Overview

In the location-related web contents retrieval by keyword matching, the location of contents are included in the search keyword or input form and that query is processed by simple keyword matching. Let's consider the query 'Kangnam station BBQ house.' The term 'Kangnam station' is the positional information describing the location of the contents and 'BBQ house' is the normal query term. In the current search engines, the term 'Kangnam station' and 'BBQ house' are all handled as text information and documents including both query terms are included in the search result. The term 'Kangnam station' has a meaning related with position, but it is treated as a simple keyword without any special consideration on its meaning during the search processing. In the current web contents service, there is no difference between the location-related web contents retrieval and the keyword-based retrieval. The problems become more severe when multiple location-related keywords are randomly contained in keyword-based retrievals. In such case, it is very difficult to eliminate unrelated documents. For example, documents that contain information on the Subway Line 2 will contain the names of 44 subway stations and such documents can be treated as documents having 44 location-related keywords, even though these documents are not directly related with 44 specific locations. Therefore, the search engines will return all the documents containing information related with the Subway Line 2 for the query asking about one location among 44 subway stations of the Subway Line 2, resulting in incorrect search results and low degree of user satisfaction.

The *local search services* are services provided by content providers, which show the contents related with specific location on the map. The portal sites, such as Naver and Daum, provide such services, by showing the map on the one side of the screen, while listing the neighboring store list on the other side of the screen. The listed stores

are the ones that have direct contracts with the portal sites. When users select a link, the summary information containing the store name, the address and the telephone number of the selected store appears on the pop-up layer [3, 4]. The local search service of Naver shows the map of 'Kangnam station' on the left side of the screen and displays the store list, with telephone number and grade, in alphabetical order with alphabet balloon symbol on the right side of the screen. When users select a link, the review information is displayed. The search results consist of information provided by contents providers and the short summary information is only provided instead of full web contents. The contents of the search results depend on the contents providers and, therefore, the information provided by local search services is not enough to users in terms of volume and quality. To relieve this problem, some providers also provide links to the review information posted by users in blog services. But, the main problem of this approach is that it only provides the information intentionally prepared by the contents providers and it shows only quick summary and review instead of full web pages.

The research on the *Geospatial Web* (or *Geoweb*), which means the combination of the location-based information and the Web, was started in 1994 by U.S. Army Construction Engineering Research Laboratory [5]. This research can be divided into subareas, such as geo-coding, geo-parsing, GCMS(Geospatial Contents Management Systems) and retrievals. The geo-coding is a technique to verify the location of a document by other tools, by recording the location information within the document as shown in Figure 1. For geo-coding, EXIF information of image files, meta information of web sites and meta tag information on text or picture data can be used [6, 7].

```
GPS Latitude : 57 deg 38' 56.83" N
GPS Longitude : 10 deg 24' 26.79" W
GPS Position : 57 deg 38' 56.83" N, 10 deg 24' 26.79" W
```

Fig. 1. Example of EXIF information of JPEG picture

The geo-parsing is a technique which translates location information in text into real spatial coordinates, thus enabling spatial query processing [8, 9]. For example, if users input 'Hannam-dong Richensia North 50m', that location information is parsed into the real coordinate. The Geospatial Contents Management Systems supports location information by extending traditional CMS. The Geospatial Web technologies support coordinate-based and region-based retrieval and they provide location information of individual documents or domains. However, these technologies depend on specific documents or specific domains and they put focus on region-based retrieval.

3 Structure of Location-Aware Web Service Systems

The overall structure of the system to easily change and service general web contents according to the location information is shown in Figure 2. This system consists of the Client, the Web Container and the Search Engine.

The Web Container module manages and transfers web contents with location information to provide location-aware web contents to users. It supports both static pages and dynamic web applications. It is able to notify update status of contents made by web applications actively to the Search Engine by using pingback methods. The correctness of information returned by the Search Engine can be improved by this mechanism.

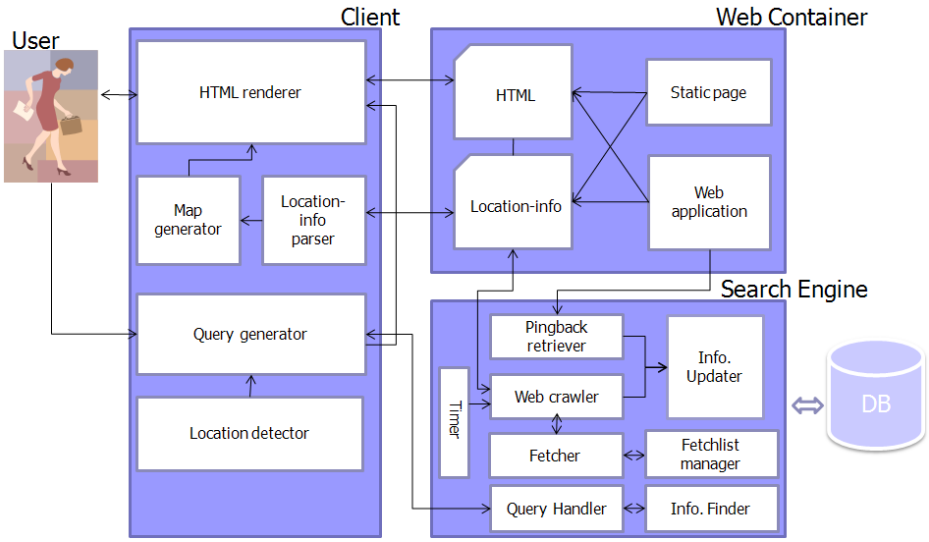

Fig. 2. The location-aware Web Service system structure

The Search Engine module collects information from the Web Container and allows users to retrieve location-aware web contents. It recognizes changes in dynamic web contents by using the Pingback Retriever and periodically collects static documents by invoking Web Crawlers using the Timer. The information is updated by the Information Updater. Location-based queries are delivered through the Query Handler and the Information Finder finally processes these queries.

The Client is the module for users to use location-aware web contents and it provides tools to support easy retrieval. It is an extension of common web browsers and includes modules, such as Map Generator, Location Information Parser, Query Generator, Location Detector, etc. It is able to show the location of contents intuitively to users by using Location Information Parser and Map Generator. The Query Generator is used to allow users more convenient location-aware retrieval.

The Location Detector is a module which captures location information by utilizing external hardware or Web Services. This module can directly get location information by using GPS or devices supporting Global Navigation Satellite System, like Beidou [10]. Also, it can indirectly get location information by using Web Services supporting Geolocation API Spec [13], such as Mozilla Firefox Geode [11] and Yahoo FireEagle [12]. Currently, the utilization of the Location Detector is not high because general PCs do not provide location information. However, it will be possible to provide more exact location information when the IP and GPS information are more commonly utilized.

3.1 Web Contents Extension to Support Location-aware Information

In the specification of traditional Web contents standards, such as HTML and XHTML[14], the markups to specify location information are not included and the namespaces for such extension are not also predetermined. Therefore, the format of web contents needs to be extended to deal with location-aware information. One method is to add new markups to the HTML/XHTML format and another method is to include links to external documents. In the previous cases, the formats of web contents were usually extended by linking external documents using <link> tag [15]. However, the standard committees have recently turned their attitude in a conservative way and they are eliminating the indiscriminately added markup tags, such as <marquee> and <embed>, to manage namespaces clearly [16]. The method adding new markup tags will face some difficulties in maintaining future web contents. Therefore, in this paper, we extended the <link> markup tag to include location information in GeoRSS [17] format, as shown in Figure 3.

```
<georss:where>
    <gml:Point> <gml:pos>45.256-71.92</gml:pos> </gml:Point>
</georss:where>

<link href="http://your.domain/foo.geo" rel="self" type="application/geoweb" />
```

Fig. 3. Location information format

The <where> markup tag, having the 'georss' namespace, is used to represent the object containing the location information. The markup tags located within the <where> tag are used to describe location using geospatial languages, such as GML (Geography Markup Language) [18]. Documents holding location information are referenced within HTML/XHTML documents using <link> markup tag and the file type of such documents is 'application/geoweb.' The documents having this file type can interpret data in GeoRSS format. We will represent coordinates using WGS84 format [19], which is one of coordinate formats supported by GeoRSS. The WGS84 format is consistent with the GPS format and, thus, it can be effectively used to support interoperability with mobile devices. Also, that format can be easily transformed into the KATECH TM128 coordinate developed by the National Geographic Institute.

3.2 Web Client

The Web Client provides facilities for web contents retrieval and visually shows documents stored in the Web Container with location information. Figure 4 shows the interaction between the Web Client and the Web Container.

The Web Client module which is an extension of web browsers consists of HTML Renderer, Location Information Parser and Map Generator as shown in Figure 4. When users provide URI of contents (1 of Figure 4), the HTML Renderer sends request to and receive the required contents from the Web Container (2 and 3). The Web Client then receives the location information (5 and 6) and generates the

appropriate map (7, 8, 9). The steps from 5 to 9 are repeated if there are more external links in the web contents.

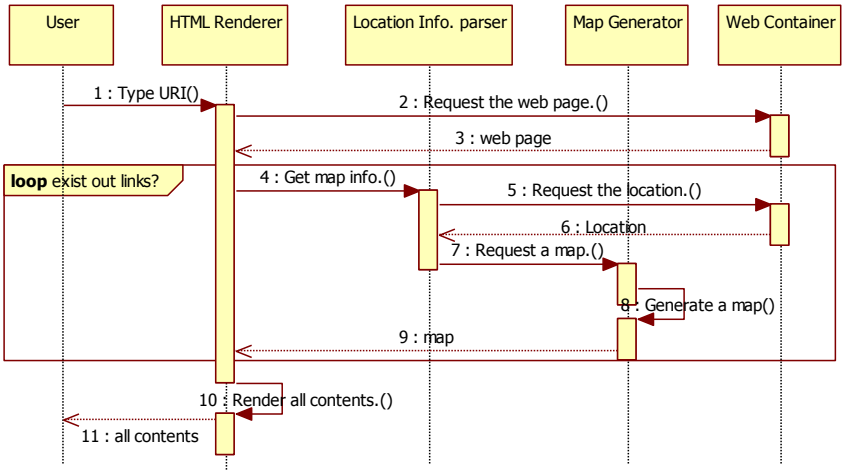

Fig. 4. Basic operations of the Web Client

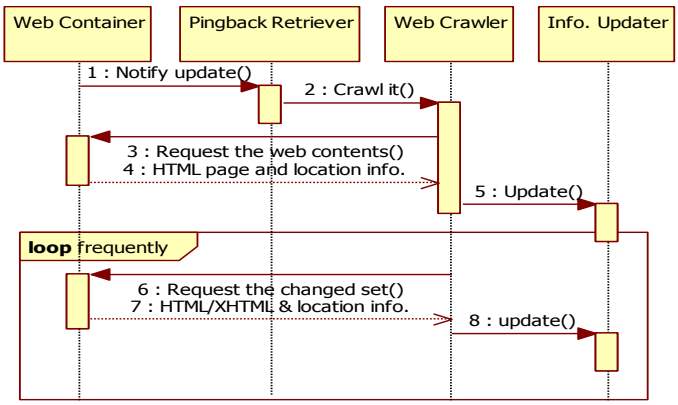

Fig. 5. Information collection by the Search Engine

3.3 Search Engine

When some information is updated by Web applications, the Web Container notifies that fact to the Search Engine (1 of Figure 5). The Pingback Retriever makes request to the Web Crawler to check the updated web contents. The Web Crawler then visits the Web Container to see the updated information (2, 3, 4) and updates the

information in the database (5). The Web Crawler continuously visits the Web Container, checks the newly updated information and downloads such updates, according to the timer event (6, 7, 8).

The Search Engine can handle keyword-based queries, coordinate and keyword-based queries, location and keyword-based queries and keyword plus keyword-based queries.

4 Implementation and Experiments

Both the Web Container and the Search Engine are implemented on the same computer, using AMD Athlon 64 Dual Core 4000+, 1GB memory, Ubuntu 4.1.2-16 Linux 2.6.22-14-server version, Apache2 and MySQL 5.0.45. The Web Client is developed on a computer with Pentium 4, 1GB memory, Windows XP, Firefox 3.0.8. The proposed functions are implemented using NPAPI [20] and extended features [21]. We compared the search results of location and keyword-based queries by using general search engines and the proposed Search Engine. The query is finding information related with 'Subway Line 2.' We first got results using general search engines and then got improved results using location-aware web contents search engine.

Table 1. Filtering rate and error rate

	N search engine	D search engine	Y search Engine
Filtering rate	66.67%	47.22%	53.33%
Error rate	3.33%	10.56%	15.56%

As shown in Table 1, we can eliminate 66.67% unnecessary search results by N search engine. However, there still exist wrong answers (3.33%), which can't be eliminated by location information and which contain the given keywords meaningless in the search results. The most common meaningless results were caused by the keywords contained in the titles of the bulletin boards included in the web pages.

5 Conclusion

In this paper, we proposed a location-aware Web Service system, which adds location information to traditional web contents. We explained the overall structure of location-aware Web Service system, which consists of the Web Client, the Web Container and the Search Engine. We also described detail structures and behaviors of the proposed systems. The Web Container module manages and transfers web contents with location information to provide location-aware web contents to users. The Search Engine module collects information from the Web Container and allows users to retrieve location-aware web contents. The Web Client provides facilities for web contents retrieval and visually provides documents stored in the Web Container with location information. The proposed methods can be implemented on top of

traditional Web Service layers. The web contents which include location information can use their location information as a parameter during search process and therefore they can increase the degree of search correctness by using actual location information instead of simple keywords. To show the usefulness our schemes, some experimental results were shortly provided.

Acknowledgments. This research was supported by the Ministry of Knowledge Economy, Korea, under the Information Technology Research Center support program supervised by the Institute of Information Technology Advancement (grant number IITA-2008-C1090-0801-0031). This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant number R01-2007-000-20958-0 funded by the Korea government (MOST). This research was also supported by Korea SW Industry Promotion Agency (KIPA) under the program of Software Engineering Technologies Development and Experts Education.

References

1. Millard, D. and Ross, M.: Web 2.0: Hypertext by Any Other Name? Proc. ACM Conference on Hypertext and Hypermedia. ACM Press. (2006) 22-25
2. HTML Spec., http://www.w3.org/TR/REC-html40/
3. Daum local information, http://local.daum.net/
4. Naver local information, http://local.naver.com/
5. An Architecture for Cyberspace: Spatialization of the Internet, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.37.4604
6. geo-extension-nonWGS84, http://microformats.org/wiki/geo-extension-strawman
7. Geographic registration of HTML documents, http://tools.ietf.org/id/draft-daviel-html-geo-tag-08.txt
8. NGA GEOnet Names Server, http://earth-info.nga.mil/gns/html/
9. U.S. Board on Geographic Names, http://geonames.usgs.gov/domestic/index.html
10. Beidou, http://www.globalsecurity.org/space/world/china/beidou.htm
11. Mozilla Firefox Geode, http://labs.mozilla.com/2008/10/introducing-geode/
12. Yahoo FireEagle, http://fireeagle.yahoo.net
13. Geolocation API Spec., http://dev.w3.org/geo/api/spec-source.html
14. XHTML Spec., http://www.w3.org/TR/xhtml11/
15. Important change to the LINK tag, http://diveintomark.org/archives/2002/06/02/important_change_to_the_link_tag
16. How to upgrade markup code in specific cases: <embed>, <applet>, <marquee>, <bgsound>, https://developer.mozilla.org/en/Using_Web_Standards_in_your_Web_Pages/
17. GeoRSS(Geographically encoded objects for RSS), http://www.georss.org/
18. GML Specification, http://portal.opengeospatial.org/modules/admin/
19. WGS 84 Implementation Manual, EUROCONTROL and ifEN, 1998.
20. NPAPI, https://developer.mozilla.org/en/Gecko_Plugin_API_Reference
21. Firefox Extensions, https://developer.mozilla.org/En/Extensions