Creating Location-based Services by utilising a Web of Places

Benedikt Ostermaier, Philipp Bolliger
ETH Zurich
Institute for Pervasive Computing
Clausiusstrasse 59, 8092 Zurich, Switzerland
{ostermaier, bolligph}@inf.ethz.ch

Abstract

We present an open, decentralized, participatory system which realizes location-based services by gently extending the existing World Wide Web. Utilizing a geospatial search engine for such a Web of Places, users are able to consume location-based services generated by arbitrary parties using their mobile phones. Special care has been taken in order to foster the creation of places and services and their mobile usage by users. We further discuss that by using ranking algorithms based on place and service definitions, finding services relevant to the current location by searching only by geographic position might be feasible even in case of large numbers of relevant places.

1 Introduction

With the convergence of several technologies such as GPS-positioning, WiFi-based connectivity and full-featured web browser applications, that become available in modern portable devices and in particular in mobile phones [11], the vision of creating a web of places and locations that are linked with arbitrary services is becoming truly feasible. The idea to associate services with a specific location, thus enabling the user to access information based on his or her position, is well-known and many systems have been built around this idea. Among the best-known are the Cooltown project [7] and the GUIDE tourist guide and navigation system [4], but many more exist [8, 10]. However, these systems rely on a central authority to maintain location information.

Our goal is to provide the user with different, user-generated services on the mobile phone based on his or her location. Thereby, the user experience should be similar to using web-based map mashups on desktop computers. For example, a convenient way to get the departure time of a bus or train is to enter your location into a search engine such as Google Maps and click on the relevant bus stop icon on the map, which may accordingly display information about next departure times. However, this approach cannot currently be extended seamlessly to other information types and cannot be applied to mobile devices without suffering in usability.

In our approach for location-based services, we do not rely on a central authority for place or service definitions but rather utilize dedicated HTML pages, which expicitly represent places and services and thus can be hosted at arbitrary web servers, be created by any party interested and be indexed and ranked by any geospatial search engine.

2 Concept

Within the scope of this paper, we define a *place* as a 2-dimensional geospatial area. Consequently, a *location* is defined as an exact point in the geospatial space. Furthermore, we define a *service* as anything that can be expressed through a web page.

In our concept, *places* are specified by their geographic shape, represented as an HTML page, which we call *place page*, and therefore identified by an URL. *Services* are also represented as HTML pages, which we call *service pages* and therefore also identified by an URL. There is no specification of the actual service – it may display static information, dynamic information or even offer an interactive web-page where the user can perform specific tasks, like for example placing an order. In order to label an HTML page as a place or service page, dedicated meta-descriptions are included in the `<head>` clause of the HTML source. These meta-descriptions are also utilized in order to bind a service page to one or more place pages, by including hyperlinks to the according places within the definition of the service. We do not specify what a place or service page should *look* like, so the user is free to use any statements in the `<body>` clause of the according page.

Since hyperlinks in the World Wide Web are unidirectional, one cannot trivially gather all the services bound to a specific place. We therefore utilise a geographic search engine, which crawls the web in order to index place and service pages, including their relations. This search engine can then be queried with a location and will subsequently return a ranked list of places close to the queried position. By selecting a place, the search engine will return a ranked list of all its associated service pages.

```
<head>
 <title>Example of a place definition</title>
 <link rel="schema.DCTERMS"
  href="http://purl.org/dc/terms/"/>
 <link rel="schema.GEORSS"
  href="http://www.georss.org/georss/"/>
 <meta name="DCTERMS.spatial"
  scheme="GEORSS.polygon" content="45.26 -110.5
  46.46 -109.48 43.84 -109.86 45.256 -110.45"/>
 <meta name="DCTERMS.type"
  scheme="DCTERMS.DCMIType" content="Place"/>
</head>
```

Listing 1. Example of a place definition

```
<head>
 <title>Example of a service definition</title>
 <link rel="schema.DCTERMS"
  href="http://purl.org/dc/terms/"/>
 <meta name="DCTERMS.type"
  scheme="DCTERMS.DCMIType" content="Service"/>
 <link rel="DCTERMS.requires"
  href="http://www.someserver.org/place.html"/>
</head>
```

Listing 2. Example of a service definition

2.1 Places

A place is represented by a dedicated HTML page, which contains specific meta-information in its `<head>` clause. Their definition is based on the standards of the Dublin Core Metadata Initiative (DCMI)[1]. In order to create a place page, we propose using `DC.Type` with the `DCMIType` vocabulary encoding and setting the value to ”Place”[2]. To specify the geographic shape, we suggest to use `DCTERMS.spatial`, a refinement of `DC.coverage`. However, as the current spatial encoding schemes of Dublin Core do not allow the creation of polygonal shapes, we are using the polygon encoding scheme of simple GeoRSS[3]. An example of a page definition is shown in List. 1. Note that the user is free to specify additional meta information, like `DC.title` or `DC.description`.

We do not specify what is displayed to the user when he requests the URL of a place, however, we suggest that relevant information of the place (like its name and its description) is displayed for usability reasons. These can for example be taken from the meta-descriptions of the page.

Since we envision a participatory approach, it is an important question how places can be defined conveniently. We suggest a web application similar to Wikimapia[4], where users can define places by drawing a polygon on a satellite image, using a mapping tool like Google Maps. Like Wikimapia, the place definitions may then also be hosted by the provider of the application, or the user may copy the geographic definition and create his own HTML page for the place.

2.2 Services

Just like places, services are represented by dedicated HTML pages, which contain specific meta-information within their `<head>` clause. In order to specify a service, one has to set the `DC.Type` of the page to ”Service”. This is straightforward and can already be realized with existing DCMI standards. In order to link the service to a specific place, one needs to specify a relation of type `DCTERMS.requires` by including a `<link>` tag of this type that targets to the URL of the according place page. An example of a service definition is given in List. 2.

As for places, we do not specify what a service should look like. However, as services are intended to be consumed on mobile devices, which usually offer only a limited amount of screen space, their presentation has to be optimized for small screens.

2.3 Search Engine

In order to find places and their associated services, we envision a geographic search engine which utilizes a crawler to find and index such pages and their relations, and provides an interface to perform a search by a given location.

As this search engine is intended for the use on mobile devices, it is important to minimize the required interaction of the user. We suggest a dedicated mobile phone application, which calls the search engine using the current position (which may be obtained from GPS, for example). A ranked lists of places nearby will then be shown to the user. When the user selects a place, the search engine will be queried for services associated with that place and return them in a ranked list. The user can then select a service, which will guide the internal browser of the mobile phone to the according web page. For usability reasons, we currently do not allow additional search terms.

The ranking algorithm is of particular importance since there may be a huge number of places and services rele-

[1] http://dublincore.org

[2] Note that there is currently no such type defined in `DCMIType`, however, we expect this to change in the future.

[3] http://www.georss.org

[4] http://wikimapia.org

vant to a specific location and users will most likely not be willing to scroll through large numbers of results. For the ranking of places, we assume that the smaller a place is, the more specific information may be associated with it, the more relevant it is in the context of the user and thus the higher it should be ranked. In the same way, the closer the current position is to a place (or its centroid), the more relevant it may be in the context of the user. In addition, places which cover the current location may be more relevant than nearby places which do not cover the current location. We are currently experimenting with a combination of these factors. The outlined ranking approach is orthogonal to the ranking algorithms used for today's web search engines, like PageRank [1], for example.

For the ranking of services, we assume that the more places a service is associated with, the more unspecific it is, thus the lower it should be ranked.

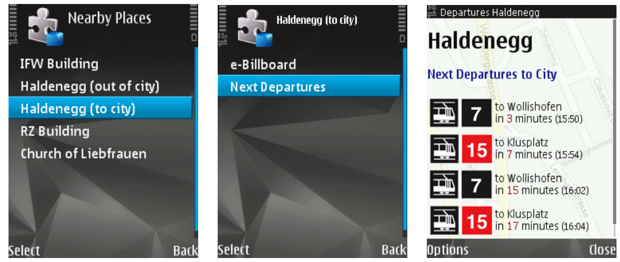
3 Prototype

We have built an initial prototype of our concept, using PHP for the server and a J2ME Midlet for the mobile phone client. Indexed data is stored in a PostgreSQL[5] database with the PostGIS[6] extension, which allows for spatial queries. Our prototype consists of three components:

1. A web-based place definition and hosting application as suggested in Sect. 2.1, where users can define places by drawing polygons on a map. Hosted places can be accessed by their URL and referenced by service pages.
2. The search engine, which returns a list of places relevant to a given location or a list of services relevant to a given place. It currently indexes only places defined with (1) and utilizes an experimental ranking algorithm based on the discussion provided in Section 2.3.
3. A mobile phone client, which determines its current location using GPS, supports the convenient access of the search engine and displays the service pages using the built-in web-browser (see Fig. 1).

Figure 1 depicts an exemplary use case, where a user is leaving his office building (denoted as "IFW Building"), heading for the next tram to the city. As he is querying the search engine the very moment he leaves the building, the place sought-after is ranked at the 3rd position. By selecting the sought-after place, the user is given a list of two services: information to the billboard located at the tram stop and a service which displays the next departures. The depicted names of the places and services are taken from

[5] http://www.postgresql.org
[6] http://postgis.refractions.net/



(a) (b) (c)

Figure 1. User requests the next tram departures at a nearby stop.

the <title> tags of the according pages. As the user selects the intended service, the corresponding service page is rendered in the built-in browser of the phone, displaying the times until the next departures of the tram line.

4 Discussion

Unlike systems such as [6] or [8], we are binding information to areal places instead of point-locations defined by coordinates, which makes our system more flexible. This enables a more sophisticated ranking of the results by the search engine and makes other location-based applications possible. There is also no restriction what size or shape a place can have or to what physical object it has to be bound. Instead of extracting location-based information from existing web pages, we rely on the dedicated definition of places and services, in order to provide a better user experience. For reasons of scalability and openness, we do not use a central database for place or service definitions (e.g., [12, 6]), but rather follow a distributed approach.

By building upon web standards, we are avoiding an isolated application, leveraging existing infrastructure and are also lowering the entry barriers for possible participants. Instead of utilizing a dedicated file format for the definition of places or services, we subtly exploit existing standards, giving the user freedom on the visualization part. Using HTML pages, we gain the advantage that they can be created easily, rendered directly in the web browser and indexed and searched by existing web search engines. Modern mobile phones feature sophisticated web browsers, which can render even complex HTML pages. The main challenge for producing web content for these devices is no longer the limited capabilities of the built-in browser but the limited size of the screen. We therefore utilize dedicated web pages for services, which display content relevant to a specific location in a compact and visually appealing way, much like it is known from today's desktop widgets.

In its current stage, our approach is limited to outdoor places, and as we rely on 2-dimensional shapes, we cannot model vertical placing. Furthermore, as the search engine may log each query, the problem of location privacy arises.

5 Related Work

To enable geospatial queries on common web-sites, [9] describes a three-stage architecture to automatically map Internet resources to geographic locations. A very similar approach is outlined in [13], where the authors discuss an algorithm that assigns location coordinates dynamically to web sites, using `whois` entries. However, with both approaches, a dedicated authority is in charge of defining location, impeding participation. Moreover, defining locations on a per-domain name base is a rather coarse-grained approach.

Another approach is to extract location information from within the contents of web pages. McCurley [10] resolves location by parsing information such as telephone numbers or zip codes from web pages, whereas the authors of [2] analyze natural language expressions that denote a location, such as "near the Eiffel tower", to construct positioning expressions. Both approaches rely on an implicit definition of locations as opposed to our approach of explicit definitions of places.

As early as 2000, the authors of [3] described ways to provide a dynamic and interactive web representation of physical places, which is automatically offered to a browsing device upon entering the place. Thereby, they scanned tags that were set up at the physical location to get the device's position. Using this location information, a custom web page is generated. Thus, the association between a tag, i.e. a location, and an URL has to be administered manually. And unlike our approach, [3] does not allow multiple services per place. Very similar to this idea is the concept described in [7]. As the authors of this paper find typing URLs on mobile devices cumbersome, they propose to associate URLs with tags that are attached to physical objects in-situ, and thereby tie web resources to locations. This however requires a "PlaceManager" to maintain directories of the resources in each place.

The system that is most similar to our approach is [12]. However, the authors do not expose places via URLs and select relevant places only by their line of sight, instead of ranking them by relevance. Another similar system is LAWS [6]. Like with our approach to provide the user with location-specific information, the user's position is sent to a web server. However, it will generate a dynamic service page on-the-fly based on the given position. Finally, [5] suggest using meta tags of HTML pages to store location information, but does not support the definition of places.

6 Conclusion and Outlook

We introduced a concept and prototype for creating location-based services by utilizing a Web of Places. Our approach is based on the combination of existing web standards by adding subtle enhancements, resulting in a gentle extension of the World Wide Web. We discuss that by using ranking algorithms based on place and service pages, finding services relevant to the current location by searching only by geographic position might be feasible even in case of large numbers of relevant places.

In a next step, we want to improve the ranking algorithm and investigate how indoor locations can be combined with our approach.

References

[1] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, Jan 1998.

[2] O. Buyukkokten, J. Cho, H. Garcia-Molina, and L. Gravano. Exploiting geographical location information of web pages. *WebDB (Informal Proceedings)*, Jan 1999.

[3] D. Caswell and P. Debaty. Creating web representations for places. *Proceedings of the 2nd international symposium on Handheld and Ubiquitous Computing*, Jan 2000.

[4] K. Cheverst, N. Davies, K. Mitchell, and A. Friday. Developing a context-aware electronic tourist guide: some issues and experiences. *Proceedings of the SIGCHI conference on Human factors in computing systems*, Jan 2000.

[5] A. Daviel and A. Kaegi. Geographic registration of html documents. Technical report, IETF, Oct 2007.

[6] A. Haghighat, C. Lopes, T. Givargis, and A. Mandal. Location-aware web system. *Workshop on Building Software for Pervasive Computing at the Object-Oriented Programming, Systems, Languages and Applications (OOPSLA) Conference*, Jan 2004.

[7] T. Kindberg and J. Barton. A web-based nomadic computing system. *Computer Networks*, Jan 2001.

[8] T. Kindberg, J. Barton, J. Morgan, and G. Becker. People, places, things: Web presence for the real world. *Mobile Networks and Applications*, Jan 2002.

[9] A. Markowetz, T. Brinkhoff, and B. Seeger. Exploiting the internet as a geospatial database. *International Workshop on Next Generation Geospatial Information*, Jan 2003.

[10] K. McCurley. Geospatial mapping and navigation of the web. *Proceedings of the 10th international conference on World Wide Web*, Jan 2001.

[11] B. Rao and L. Minakakis. Evolution of mobile location-based services. *Communications of the ACM*, Jan 2003.

[12] R. Simon and P. Fröhlich. A mobile application framework for the geospatial web. *Proceedings of the 16th international conference on World Wide Web*, Jan 2007.

[13] C. Watters and G. Amoudi. Geosearcher: Location-based ranking of search engine results. *JASTIS*, Jan 2003.