

# The Locative Web

Erik Wilde  
School of Information  
UC Berkeley  
dret@berkeley.edu

Martin Kofahl  
Geodesy and Geoinformatics  
University of Rostock  
martin.kofahl@uni-rostock.de

## ABSTRACT

The concept of location has become very popular in many applications on the Web, in particular for those which aim at connecting the real world with resources on the Web. However, the Web as it is today has no overall location concept, which means that applications have to introduce their own location concepts and have done so in incompatible ways. By turning the Web into a location-aware Web, which we call the “Locative Web,” location-oriented applications get better support for their location concepts on the Web, and the Web becomes an information system where location-related information can be more easily shared across different applications and application areas. We describe a location concept for the Web supporting different location types, its embedding into some of the Web’s core technologies, and prototype implementations of these concepts in location-enabled Web components.

## Categories and Subject Descriptors

H.3.5 [Information Storage and Retrieval]: Online Information Services—*Data Sharing, Web-Based Services*

H.3.4 [Information Storage and Retrieval]: Systems and Software—*Information networks*

## General Terms

Design, Management, Standardization

## Keywords

Geolocation, Location-Based Services, URI, HTTP, HTML

## 1. INTRODUCTION

The Web as the most successful information system ever built supports applications with very diverse data models and interface requirements. The reason for the Web’s versatility is that it provides a universally accepted language

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*LocWeb 2008*, April 22, 2008, Beijing, China.  
ACM 978-1-60558-160-6/08/04 . . . \$5.00.

for describing documents (which as dynamic documents often serve as interfaces to information repositories), and an architecture which scales well and is based on declarative languages and simple operations. Specifically, the concept of *Representational State Transfer (REST)* [11], the principle underlying the *Hypertext Transfer Protocol (HTTP)* [12], allows the Web to evolve by introducing new resource types (by introducing new representations for resources), without any change in the basic infrastructure.

Earlier proposals listed in Section 2 either use location data for specific application areas on the Web, or define a single location scheme. This paper discusses the question of how the Web can be turned into a location-aware information system. We merge existing drafts into a single location concept which can be used for several areas on the Web and refer to this new location-aware Web as the *Locative Web*. The motivation for this is the observation that “location” has become an important concept in a wide variety of application areas, some of these areas are described in Section 3.

Some application areas require a much more sophisticated model of locations than what the Web could possibly support, these areas are the ones which typically use some sort of *Geographic Information System (GIS)*. GIS provide very sophisticated spatial data models and excel at managing spatial data, but most GIS have Web-based user interfaces, but do not integrate their data into the Web. But even for areas with sophisticated location models, it makes sense to provide a limited perspective of their internal data models in a Web-friendly way, so that some of their services can be easily integrated into Web scenarios [37]. Section 4 describes the location concept for the Web that we are proposing, and Section 5 takes these concepts and describes how they are embedded into different components of Web architecture.

Sections 6 and 7 describe two implementations that use these concepts and their representation in Web technologies. We use these implementations to further explore the capabilities and limitations of the proposed extensions to the Web’s architecture, and while we do not expect that our model will be directly and completely adapted for the Web, we believe that it provides a valuable starting point for a discussion about how to realize the locative Web.

## 2. RELATED WORK

Investigations towards a locative Web are related to *semantic markup or analysis* of documents available on the Web, the extension of client/server *communications* with location information, and *applications*.

There are various resource types supporting the association with a real-world location, such as the *EXIF* header for digital pictures. Resources containing no explicit metadata may be geocoded by content and context analysis [2, 8, 31]. For the most common resource types based on *HTML* [29] or *XHTML* [26], there are several ways for geocoding: *HTML's address element* [29] is the earliest way of marking up location information, but it has neither a well-defined syntax nor well-defined semantics for its content. The *Dublin Core Metadata Initiative* defined a set of *HTML* metadata elements [36] for resource discovery containing a “coverage” element allowing substructures and various location types. The *ICBM* [30] metadata element for *HTML* is a simple coordinate pair; whereas the *geo.position* and its corresponding metadata elements [9] can additionally state a civic address following the *geopriv* location object format [27]. The *GeoRSS* [25] encoding of geographical locations and the *W3C* geocoding vocabulary<sup>1</sup> are designed for use in *RSS* and Atom feeds as well as in other *XML*-based document formats.

Client/Server communications usually are based on *HTTP* or similar protocols. There are a few drafts introducing the additional use of a client’s location: an extra *message header* in each request [7, 10, 35] is the most common way for additional metadata, such as a position. Content negotiation can then be based on that mechanism [10, 35]. However, in order to transfer desired location data, *URI query parameters* [14] can be used as well. This technique is also used by *i-mode* [16].

Beside search engines, some applications already demonstrate some advantages of the locative Web: Using *i-area* [16] (which is part of the *i-mode* environment) compatible handsets and Web sites, location-based services can be implemented on the Web. Based on *URIs* for locations [22], a Firefox browser extension supports *geographical links*. Locally available *Web presences of things* (e.g. books) may provide links to related resources or physical objects [19]. Improved *spatial queries* can be achieved using visibility and the field of view as additional query parameters [32].

### 3. APPLICATION AREAS

Sections 3.1 through 3.4 describe some application areas where the locative Web provides better support for implementing services and applications than the current Web (this overview does not claim to be complete and should only demonstrate that there are many potential application domains for a locative Web). Section 3.5 concludes this brief overview of application areas with a scenario that spans multiple areas of utilization and demonstrates how the concept of location is percolating through all interactions described in the scenario. By using a consistent location concept across all fields of the application, it is possible to obtain network effects which are impossible to attain without a location-aware infrastructure.

#### 3.1 Location-Based Services

*Location-Based Services (LBS)* are based on the assumption that many services for (mostly mobile) clients can be enhanced by customizing the service based on the client’s location. The idea originated with the widespread availability of data services for mobile phones, providing proprietary

services which work best in semi-closed architectures such as the *Wireless Application Protocol (WAP)* [34], but which are hard to extend to mobile clients with full Internet connectivity and standard browsers.

Mobile phone carriers are selling the current location of their customers to service providers, and therefore it is hard to determine a client’s location by independent providers. For a more open approach and for privacy reasons, a client should state the current (or assumed) location by itself in an appropriate way. Positioning sources may be the *Global Positioning System (GPS)* or local network services (Section 6 describes an implementation using *HTTP* for transmitting location information, Section 7.2 describes a similar implementation of a proxy server).

Location information not only can be very useful for providing services, it also is very sensitive information that should only be disclosed with the consent of the user. Privacy policies for location information and mobile devices and *LBS* still have to be developed, but there are many scenarios where *LBS* on mobile devices can provide better service than without location information.

#### 3.2 Web of Things

Pervasive [15] and ubiquitous [13] computing are fusing networked computer technology with many aspects of our daily lives. To a large extent, these fields have been concerned so far with how to take things which are more embedded into the physical world than the usual computer hardware, and make them available in ways which enable new classes of applications beyond the traditional networking scenarios. Many of the questions in these areas currently revolve around networking capabilities, such as ad-hoc networking algorithms, and the integration of computing infrastructure into the physical world.

The *Web of Things* is a vision which takes the *Internet of Things* (a common term in the pervasive and ubiquitous computing communities) one step further, so that physical objects are not only networked through ad-hoc setup of Internet connections, but instead are made available as resources on the Web, and can be used in standard Web interactions. Since physical resources always have a location, this scenario very often requires support for location-aware interactions, and thus could greatly benefit from a location-aware infrastructure.

#### 3.3 Resource-Object Associations

The scenario presented in Section 3.2 is based on the assumption that networked computer technology is embedded into everyday objects and augments the physical world. However, there also is a large set of applications which are not augmenting the real world, but still are interested to associate Web resources with it. Examples for this are all *field sciences*, which often create Web artifacts (such as images, documents, or video/audio recordings) and want to associate these with location information.

In many cases, these scenarios have layered levels of complexity. In archaeology, for example, if some ancient artifact is found, this is a physical resource which may be moved to a different location such as an archive or a museum, but it also is related to the location where it has been found, and this again can have different levels of descriptions, such as the coordinates in some grid used for an excavation project, or symbolic names for certain areas in the excavation area.

<sup>1</sup>Informally defined at <http://www.w3.org/2003/01/geo/>.

Whatever the exact information model that is used for a specific project will be, it will contain a substantial number of location-related concepts, because this information is of great significance for future interactions with the artifact.

### 3.4 Social Networking

Social networking (which often is perceived to be one of the central aspects of Web 2.0) has become an important driver for Web applications and Web content. Since social networking is yet another area where Web applications aim to represent something in the real world, location concepts often play an important role in these applications. For social networking, the question of where users are can become an important driver of making these services more valuable and more interesting, because this enables those services to match users spatially.

Many social networking sites provide some location concepts, but these in most cases are closed concepts and only useful with one particular service. In addition, an increasing number of services are implementing some social networking by providing a platform for managing information that is intended to be shared across users. Popular examples are photo sharing sites such as *Flickr* and *Panoramio*, which use geotagging of pictures to make them available in as many ways as possible (for example, both services provide interfaces where it is possible to explore pictures using a map-based interface). Currently, there are two major trends in the social networking landscape:

- Big services such as *Facebook* attempt to become the one place on the Web where social networking happens. These services typically open up a little bit by providing an API to integrate other services, but the main goal still is to be the main hub for social networking. The main motivation for this movement is the underlying business model of targeted advertising.
- A large number of new providers implement services which often are not sufficient as stand-alone social networking services, but are valuable as add-ons. These services must be able to be integrated into other services. This integration not only requires technical alignment, it also requires a data model which can be easily reused in an integration platform.

Both developments are currently active, but it is likely that the big aggregators will not last very long. The important observation is that in both scenarios, social networking happens as service aggregation, which means that shared underlying models (of all the data that should be aggregated, location being an important type among those) are essential. So far, location is not treated in great detail, though. For example, *Open Social*, Google's attempt to publish an API for social networking platforms to counter Facebook's dominance, does have location as part of an information about a person, but only allows a coordinate-based position as the only way to represent locations.

### 3.5 Scenario

The following scenario combines different aspects of a locative Web, which benefits from a standardized way of addressing locative issues. A trip to a museum is used as an example, starting with a geographical hyperlink found in an advertising email. It allows the browser to open a routing

Web site. Because of coordinates included in each request made to this Web site, it acts as a navigator. In the museum, the Web site queries a search engine with the supplied coordinates and/or derived civic address. The museum's Web address is easily found because its entry page contains geotags. After paying the fee by online payment, the device can access a local wireless network, allowing extensive access to museum resources. Since GPS is not working inside buildings, a proxy takes over network-based positioning. Thus, the Web-based visitor guide can use accurate location data and provide appropriate content and functions for controlling exhibits via the Web. Live-linked images from community sites such as *Flickr* establish a relationship to the real world where an exhibit originated.

## 4. LOCATION ON THE WEB

For the Web to evolve into the locative Web, a location-aware information system, the most fundamental issue is that there must be shared semantics for location concepts, and ideally there also should be guidelines for location syntax, so that these semantics are represented in compatible ways (if possible).

There are two dimensions to this problem. The first dimension is what the location concepts are that the Web should support. Obviously, location concepts used in applications vary widely based on the requirements of the application. The second dimension is the question of where these concepts should be embedded into the Web's architecture, so that they can be used in standard scenarios of Web-based interactions. In the following sections, we describe the location concepts that we identified as being the core concepts for location across a wide variety of applications. The question of how these should be embedded into Web architecture is then discussed in Section 5, which also discusses the syntax issues that arise because of the different environments into which the concepts should be embedded.

### 4.1 Coordinates

An objective way of talking about locations is to measure coordinates. These coordinates have to refer to some well-defined coordinate system, and the *World Geodetic System (WGS84)* [23] is the coordinate system that is most widely used and supported today. It is the coordinate system that is natively used by the popular *Global Positioning System (GPS)*, which allows satellite-based measurements of locations anywhere on earth. Of course, not all devices that might want to position themselves have GPS receivers, and there exist a number of well-known techniques for positioning in mobile phone networks and ad-hoc networks. These position types can then be mapped to WGS84 coordinates, with varying degrees of accuracy. However, regardless of what method a device uses to evaluate its position, a coordinate-based position can always be represented in an interoperable way.

Elevation is an optional part of coordinates, which might be necessary for specific applications. Since local elevations might refer to different references, the *Earth Gravity Model (EGM96)* [21] geoid must be used to map elevations to a single reference system, such as *Mean Sea Level (MSL)*.

### 4.2 Countries

Apart from coordinates as discussed in the previous section, there are few location features which are universally

used, and where some vocabulary is maintained in a well-defined and open way. One example might be continents, another one is countries. While countries are not very stable over time, at any point in time there is a consensus on which countries exist and where they are.<sup>2</sup> *ISO Country Codes* [17] are standardized codes for countries as well as for country subdivisions, and many other standards (such as the Internet’s DNS top-level domain names) use these codes and simply rely on ISO to determine what countries exist at any given time.

ISO defines three different codes for countries, *alpha-2* (two-letter codes, for example `us`), *alpha-3* (three-letter codes, for example `USA`), and *digit-3* (three-digit codes). Furthermore, ISO also defines codes for country subdivisions. However, only the alpha-2 codes are available free of charge, the other codes (although widely-used and referenced) must be purchased from ISO.

### 4.3 Places

While coordinates (Section 4.1) and countries (Section 4.2) refer to well-defined and geographically identifiable concepts of locations, there often are more loosely defined location concepts, with two typical examples being the following scenarios:

- *Centrally Controlled Vocabularies:* Frequently, place names are used in a context where it is clear how such a name should be interpreted. Examples include state names, zip codes, or city names within a certain geographic region (in the latter case, city names may need disambiguation if there are cities with the same name). In all these cases, there is some controlled vocabulary of these place names, and any name used in this context essentially is a reference into this vocabulary.
- *User-Defined Vocabularies:* For smaller user groups, it might make more sense to use a customized vocabulary with a more limited extent. For example, for social networking applications, users might only want to disclose their location in broad concepts such as “at work,” “at home,” “at school,” or “on vacation.” These concepts might or might not be associated with a concrete geographical region, but in either case they serve as a place name which might be useful for match-making between users of the social network.

When talking about vocabularies, the two main questions are how to *identify* them, and how to *define* them. Our approach is to provide a standardized way of identifying vocabularies, but no standardized way for defining them. For identification, we use the standard Web abstraction for that, which is the *Uniform Resource Identifier (URI)* [4]. This strategy is the same as for XML, which also supports the identification of *XML Namespaces* [5] by using URIs, but does not mandate any specific format for a namespace description, or for an XML vocabulary. Similarly, when using a place name for location identification, there must be an identifier for the vocabulary, either explicitly or implicitly through the context in which the place name occurs.

<sup>2</sup>This is not entirely true since countries for political reasons sometimes take different stances than the majority of the world, and because borders are sometimes contentious, but for most locations of the earth’s land surface, the attribution to a country can be done reliably.

In the same way as it is good practice to use URIs for XML namespaces which somehow describe the vocabulary, our recommendation is that a URI for a place name vocabulary should identify a Web resource which serves as a description and possibly definition of the vocabulary. We do not anticipate, however, that a single language will be universally used to define place name vocabularies, because requirements for these vocabularies will differ substantially, based on the requirements of the applications for which these vocabularies are developed. As one possibility, Section 7.1 presents the *Place Markup Language (PlaceML)*, which is a simple XML-based language for defining place names.

Depending on the application area, there can be very large place name vocabularies, such as the *Getty Thesaurus of Geographic Names (TGN)*<sup>3</sup> or *GeoNames*<sup>4</sup>, or very small place name vocabularies, such as customized place name vocabularies for use by individuals in social networks. Sometimes it might make sense to be able to exchange the complete vocabulary as part of a Web interaction, whereas for large place name vocabularies this is not practical. So exchanging vocabularies is an option (and can be covered by agreeing on some representation), but probably will not be the norm. Referring to the vocabulary to which a place name belongs, however, is essential for being able to interpret that name in the correct context.

Place name vocabularies need a well-defined policy on updates. The maintainer of a place name vocabulary should explicitly state their policy with respect to changes in the place names defined in that place name vocabulary. Otherwise, it is impossible to determine how an update policy for a local copy of the vocabulary should be implemented.

## 5. LOCATION WEB ARCHITECTURE

To use the location concepts presented in Section 4 and put them into use on the Web, they must be supported in different parts of *Web Architecture* [18]. The Web supports certain concepts natively (such as language identification and negotiation for content and transfer), and in order to implement the locative Web, the location concepts must become part of this native support as well.

The most important concepts on the Web are *identification*, *transfer*, and *content*. The main Web technologies in these areas are URIs for identification, HTTP for transfer, and HTML and XML for content. Sections 5.1 through 5.3 look at these Web technologies and describe how our location concepts are embedded into these components of Web architecture.

### 5.1 Location Resources

The *Uniform Resource Identifier (URI)* [4] is the Web’s way of identifying resources. Identification can be a useful function in its own right, as demonstrated by *XML Namespaces* [5]. As another example, the `tag` URI scheme [20] is only intended as an identification mechanism, it does not define resolution or access methods. The proposed `geoloc` URI scheme (as well as the `geo` scheme [22]) also is only defined for identification and has no resolution or access methods. However, there are semantics associated with it according to the location concepts described in Section 4.

<sup>3</sup>The thesaurus currently contains more than 1.1 million place names.

<sup>4</sup>A free database with over 8 million geographical names.

`geoloc` URIs can be used with or without a vocabulary identifier (a URI). Any URI reserved characters in names must be escaped.<sup>5</sup> The identifier can be omitted if there is a context of the URI which establishes the vocabulary. Otherwise we expect decimal WGS84 coordinates being the default context. Thus, the following first two examples shown in Figure 1 are synonymous. The third example specifies an U.S. zip code, the last one the U.S. as a country.

```

geoloc:27.988056,86.925278
geoloc:27.988056,86.925278;http%3A%2F%2Fwww. \
  opengis.net%2Fgml%2Fsrs%2Fepsg.xml%234326
geoloc:94720;http%3A%2F%2Fusps.gov%2Fns%2Fzip
geoloc:us;http%3A%2F%2Fiso.org%2Fns%2Fiso3166-1

```

Figure 1: Distinction between Locations by URIs

## 5.2 HTTP Location Metadata

The *Hypertext Transfer Protocol (HTTP)* [12] defines a number of header fields which are used for various purposes, they are broadly classified into *general*, *entity*, *request*, and *response* headers. Location information could be transferred by the client (indicating the location of the client), and the server (indicating the location of the server), and since it does not apply to the entity being transferred, a new `geoloc` header field is being defined as a general header field. It uses the same syntax as the URI described in Section 5.1, but allows whitespace around separators and does not require escaping of characters.

As an additional functionality apart from merely embedding location information, *content negotiation* for location concepts (as described in Sections 4 and 5.1) could become part of HTTP. An `Accept-Geoloc` header could specify a server’s capabilities in terms of supported location concepts using corresponding URIs. Content negotiation would work in the same way as for content types (`Accept`), languages (`Accept-Language`), content encodings (`Accept-Encoding`), and character sets (`Accept-Charset`).

HTTP content negotiation is defined sufficiently open so that a new header field could be introduced and could then be legally used as information for location-based content negotiation. HTTP’s concept of *transparent negotiation* would nicely map to the scenarios described in Section 7, where a cache can map location concepts based on its knowledge of the location concepts used by the client, and the location concepts supported by the server.

## 5.3 Resource Location Metadata

The location concepts described in Section 4 should be supported as concepts that can be used in content. The main content format on the Web is *HTML* [29] and increasingly its XML-variant *XHTML* [26], but there are also many *XML* [6] formats, which should be able to embed this information.

<sup>5</sup>It would be legal for place name vocabularies, however, to define and encode hierarchic place names, in which case they should use URI syntax for hierarchies. `geoloc:Berlin/Berlin/Tempelhof` could for example use a province, city, borough hierarchy for place names in Germany and identify the *Tempelhof* borough in *Berlin* (which is located in the province of *Berlin*).

One example for this is the *Atom* [24] format, which is a popular format for feeds.

Most previous attempts at adding location information to the Web look at specific content formats only, *GeoRSS* [25] only considers feed formats (and has been updated to cover Atom as well), and another proposal [9] only looks at HTML and XHTML. Another question is how embedding should be specified, for HTML alone there exist three main approaches, one being a custom syntax (microformats often use this approach), the second being adding *RDF/XML* [3], and the third being *RDFa* [1]. For XML, there also are various ways how reusable vocabularies can be defined, often differing substantially in their usage of elements, attributes, and namespaces.

Conciliating HTML and XML metadata standards is not as easy as it may seem at first sight, because HTML markup is often created manually and has its focus on simplicity and ease of use and robustness, whereas XML markup often is machine-generated and the focus is more on modularity and clean design and extensibility. The two most important approaches for representing location within resources is Daviel’s format [9] for HTML (which is limited to making statements about the complete Web site), and *GeoRSS* [25] (which supports rather complex GML expressions) for feeds. Both are based on different location concepts and thus are hard to use together consistently.

We propose to use *RDFa*-compliant syntax for HTML, and an attribute-based form for integration into XML-based resources. While so far we have not created concrete schemas for embedding our location concepts into resources, we believe that it is essential that there are formats for HTML and XML which are aligned with the location concepts used in other parts of the Web architecture.

## 6. GEO-ENABLED BROWSER

In order to design and test locative Web applications, we implemented a few capabilities turning a regular user agent into a geo-enabled browser. In particular, we focused on the negotiation of different location concepts. An *IE Mobile* running on *Windows Mobile 6* is used as a testbed. A vendor-specific extension using common browser APIs contains improvements to the underlying system.

The ability to retrieve location based information from the Web is basically established by querying a local location finder before modifying requests made to the Web. Currently, we use GPS, DHCP’s *GeoConf* [28], and GSM Cell ID for positioning and thus only support location types provided by these techniques. Web services for geocoding and reverse geocoding (such as *Google My Location* or *GeoNames.org*) will help to retrieve the desired location type in the future, e.g. zip code instead of geographical coordinates.

The process of retrieving localized content follows the concept described in Section 5.2, illustrated in Figure 2. As long as there are no predefined rules, the browser acts as usual. However, the browser analyzes possible `Accept-Geoloc` headers sent by the Web server. The resulting list of supported location types (each identified by an URI as described in Section 5.2) has to be compared to general privacy rules or user settings for this specific Web site. In order to apply one of the remaining location types, the available position drivers mentioned above are queried. Subsequent HTTP requests will contain the desired `geoloc` header.

An experimental feature further enhancing the locative

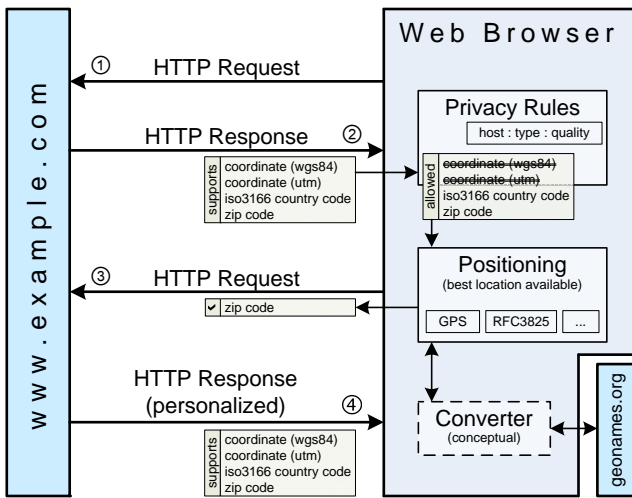


Figure 2: Geo-Enabled Browser Architecture

Web is to make location data available to scripts. Mobile devices (for example Web 2.0 city guides) can benefit from permanently available position information.

## 7. LOCPROXY

Even though mobile phone carriers often have high-quality location information for individual phones (which in the U.S. for example is mandated by E911 [33]), they usually do not make that information openly available. This means that even though mobile phones often could be located precisely (by the carrier or by GPS-enabled phones as described in Section 6), in practice this does not happen, and HTTP requests from mobile phones do not carry any location information.

While the approach described in Section 6 is more elegant because it directly updates the mobile device’s Web client to be location-aware, this requires an update to the device’s software, and also requires some sort of position information (such as a GPS position) that can be accessed by the updated software.

An alternative approach is described in this section, the approach is based on the idea to implement the location support in a Web intermediary, which then acts as the location-aware component in the HTTP request/response chain. This approach is independent from any specific client and aims at solving the chicken-and-egg problem of location-based services. While the proxy architecture described here is not feasible as a large-scale and long-term solution, it is ideally suited for prototype services. Our main goal thus is to provide an environment where location-based services can be developed without the need to update all clients. Instead, all clients use the *Location Proxy (LocProxy)*, which augments the plain HTTP requests sent by clients with location information.

One of the most important parts of the LocProxy concept is to keep the concept of locations as flexible as possible. To achieve this goal, we have developed a format for representing place name vocabularies, which is used by the LocProxy and allows users to easily define and configure a place name vocabulary.

## 7.1 Place Markup Language (PlaceML)

The *Place Markup Language (PlaceML)* is a language for representing place name vocabularies. As described in Section 4.3, our location concept does not mandate a specific format for defining place name vocabularies, all that is required is a URI-based identification. However, in many cases it is beneficial to have some agreed-upon language for representing place name vocabularies, and PlaceML is one such language.

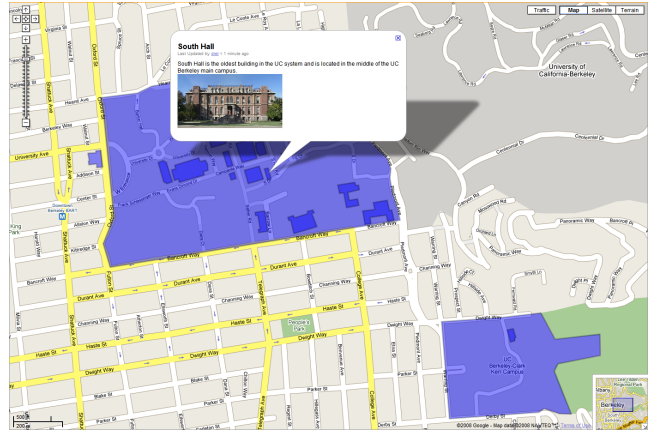


Figure 3: Google Maps Interface for Defining Places

The main properties of PlaceML are that it supports place hierarchies (which have to be acyclic graphs), inclusion and exclusion polygons and circles for defining places, aliases, place images, place descriptions, and internationalization. Since places often rely on a geographical identification of a place, we support import from KML, which is a format for representing map-related data. Figure 3 shows an example of a very simple place name vocabulary created in Google Maps, which consists of few places, all of them identified by polygons using Google Maps “My Maps.”

When exported as KML, it can be mapped to PlaceML in the way shown in Figure 4, which demonstrates some of the core features of PlaceML (the only part edited by hand is that “South Hall” is part of the “Main Campus”).

```
<placeml xml:lang="en">
  <name>UC Berkeley Campuses</name>
  <place id="MainCampus">
    <name>Main Campus</name>
    <description>The main campus has ...</description>
    <location>
      <polygon type="include">...</polygon>
    </location>
  </place>
  <place id="SouthHall">
    <name>South Hall</name>
    <parent ref="MainCampus"/>
    <description>South Hall is ...</description>
    <location>
      <polygon type="include">...</polygon>
    </location>
  </place>
</placeml>
```

Figure 4: PlaceML Example

KML and PlaceML are not perfect matches, which means that a transformation in either way most likely will be lossy. KML supports more geographic constructs (such as lines), while PlaceML allows richer descriptions (aliases, internationalization), the association of places (hierarchies), and places without geographic definitions. However, KML is popular and useful for visualization in many different applications (e.g., Google Maps and Google Earth), so we also support transformations from PlaceML to KML.

## 7.2 HTTP Proxy

PlaceML is a markup language for defining a vocabulary of place names. We use this markup language as the configuration for an experimental HTTP proxy which is used to “simulate” location-aware clients. The basic idea is that users use a simple interface for selecting their current location, and this interface is driven by PlaceML. Figure 5 shows this as the first step of interacting with the LocProxy. After configuring the location, the proxy knows a user’s identity and the associated location, which is based on a PlaceML definition.

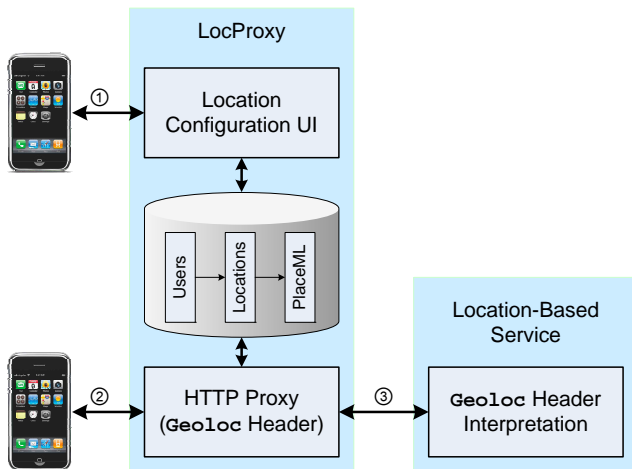


Figure 5: LocProxy Architecture

LocProxy users then have to configure LocProxy as their HTTP proxy. LocProxy uses HTTP *proxy authentication* to identify users, which means that each request going through the proxy has to carry a **Proxy-Authentication** header field. This proxy authorization information allows the proxy to identify and authenticate users. It also allows the proxy to look up the associated location information in the database (which contains information about users, their locations, and the supported place name vocabularies), and include it as a HTTP **geoLoc** header field in the request that is forwarded along the HTTP chain.

The current version of LocProxy simply inserts the user’s configured location information into the user’s HTTP request before forwarding the request to the server. This works well for a prototype and limited deployment, because the location information is based on a PlaceML definition (a place name vocabulary as described in Section 4.3), and all participants in the scenario support this specific vocabulary.

However, more generally, a location proxy could also negotiate and even map between different location notations or place name vocabularies. There are three basic scenarios:

- *Name to Coordinate*: If the place name used for configuring a location is unknown to the service being contacted, the most useful approach would be to map it to coordinates. If the PlaceML contains geolocation information for places, place names can be mapped to, for example, the geometric center of the place’s geolocation.
- *Coordinate to Name*: If the device itself supports geolocation (Section 6 describes such a device), but the service expects a place name, the coordinates determined by the device’s location service can be mapped to the service’s places. This again requires the PlaceML to contain geolocation information.
- *Name to Name*: If the user uses a certain PlaceML to configure locations, but the service expects a different place name vocabulary to be used, there can be some mapping between place names, which can either be based on conceptual knowledge,<sup>6</sup> or on mapping from a place name to coordinates using the first PlaceML, and mapping coordinates to a place name using the second PlaceML.

While the limited functionality of the current LocProxy does not support these scenarios, we believe that this scenario of negotiating between different location concepts (see our discussion of HTTP content negotiation in Section 5.2) will become an important component of handling locations.

## 8. FUTURE WORK

In order to push the development of the locative Web, a number of issues have to be discussed. The main tasks pertain the domains of location metadata in the Web’s resources (see Section 5.3) and in its communication (see Section 5.2). A consistent way about how to embed location information into HTML and XML has to be defined. The concept and the syntax of HTTP-included location data has to be discussed, and in particular how to ensure user privacy. HTTP content negotiation may help by using different forms of location. But in order to support interoperability, it is necessary to define commonly used location types (see Section 4) which every client or server should support.

Manufacturers of end-user devices might quickly close the gap between integrated positioning units and “Web 2.0” applications. To ensure a uniform evolution of the Web, a *location* object and interfaces for use in scripts should be defined.

## 9. CONCLUSIONS

The *Locative Web* is a Web which supports location as a Web-level concept. This paper describes a location concept which is based on different ways of how locations can be expressed, e.g. as coordinates, as country names, or as other place names. Locations must be qualified by identifying a vocabulary. We follow the idea of XML namespaces and require the vocabulary to be identified by a URI, but do not mandate a format for a description of the vocabulary or the vocabulary itself.

The location concepts are mapped to various components of Web architecture, specifically URIs, HTTP, and resources.

<sup>6</sup>For example, mapping cities to states, or city districts to cities.

For URIs and HTTP we present a concrete syntax, whereas for resources we only have requirements for markup which should make location information within resources easy to use and embed.

Finally, we describe two prototype implementations. The first one is a geo-enabled browser, which uses locally available location information to add location information to HTTP requests. This allows location-based Web services to deliver localized content. The other prototype uses a different approach to support location-based services on the Web. It uses an HTTP proxy where users can configure their location, which will then be added to all requests which are sent through the proxy. This has the disadvantage that the users need to manually change the location information, but the advantage that no special software is required on the client side. Both prototypes aim at providing environments within which it is possible to provide and experiment with location-based services. The novelty of supporting different location types produces advantages in compatibility and privacy but may also exclude user groups from Web services due to rejected location types.

## 10. REFERENCES

- [1] BEN ADIDA, MARK BIRBECK, SHANE MCCARRON, and STEVEN PEMBERTON. RDFa in XHTML: Syntax and Processing — A Collection of Attributes and Processing Rules for Extending XHTML to Support RDF. World Wide Web Consortium, Working Draft WD-rdfa-syntax-20080221, February 2008.
- [2] EINAT AMITAY, NADAV HAR'EL, RON SIVAN, and AYA SOFFER. Web-a-Where: Geotagging Web Content. In MARK SANDERSON, KALERVO JÄRVELIN, JAMES ALLAN, and PETER BRUZA, editors, *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 273–280, Sheffield, UK, July 2004. ACM Press.
- [3] DAVE BECKETT. RDF/XML Syntax Specification (Revised). World Wide Web Consortium, Recommendation REC-rdf-syntax-grammar-20040210, February 2004.
- [4] TIM BERNERS-LEE, ROY THOMAS FIELDING, and LARRY MASINTER. Uniform Resource Identifier (URI): Generic Syntax. Internet RFC 3986, January 2005.
- [5] TIM BRAY, DAVE HOLLANDER, ANDREW LAYMAN, and RICHARD TOBIN. Namespaces in XML 1.0 (Second Edition). World Wide Web Consortium, Recommendation REC-xml-names-20060816, August 2006.
- [6] TIM BRAY, JEAN PAOLI, C. MICHAEL SPERBERG-MCQUEEN, EVE MALER, and FRANÇOIS YERGEAU. Extensible Markup Language (XML) 1.0 (Fourth Edition). World Wide Web Consortium, Recommendation REC-xml-20060816, August 2006.
- [7] DAVIDE CARBONI, ANDREA PIRAS, STEFANO SANNA, and SYLVAIN GIROUX. The Web Around The Corner: Augmenting the Browser with GPS. In *Alternate Track Papers & Posters Proceedings of the Thirteenth International World Wide Web Conference*, New York, NY, May 2004. ACM Press.
- [8] PAUL CLOUGH. Extracting metadata for spatially-aware information retrieval on the internet. In *GIR '05: Proceedings of the 2005 workshop on Geographic information retrieval*, pages 25–30, Bremen, Germany, 2005. ACM.
- [9] ANDREW DAVIEL and FELIX A. KÄGI. Geographic Registration of HTML Documents. Internet Draft draft-daviel-html-geo-tag-08, October 2007.
- [10] ANDREW DAVIEL, FELIX A. KÄGI, and MARTIN KOFAHL. Geographic Extensions for HTTP Transactions. Internet Draft draft-daviel-http-geo-header-05, December 2007.
- [11] ROY T. FIELDING and RICHARD N. TAYLOR. Principled Design of the Modern Web Architecture. *ACM Transactions on Internet Technology*, 2(2):115–150, May 2002.
- [12] ROY THOMAS FIELDING, JIM GETTYS, JEFFREY C. MOGUL, HENRIK FRYSTYK NIELSEN, LARRY MASINTER, PAUL J. LEACH, and TIM BERNERS-LEE. Hypertext Transfer Protocol — HTTP/1.1. Internet RFC 2616, June 1999.
- [13] ADAM GREENFIELD. *Everyware: The Dawning Age of Ubiquitous Computing*. New Riders, Indianapolis, Indiana, March 2006.
- [14] AMIR HAGHIGHAT, CRISTINA VIDEIRA LOPES, TONY GIVARGIS, and ATRI MANDAL. Location-Aware Web System. In *Proceedings of the Workshop on Building Software for Pervasive Computing at OOPSLA'04*, Vancouver, Canada, October 2004.
- [15] UWE HANSMANN, LOTHAR MERK, MARTIN S. NICKLOUS, and THOMAS STOBER. *Pervasive Computing: The Mobile World*. Springer-Verlag, Berlin, Germany, 2<sup>nd</sup> edition, August 2003.
- [16] NTT DoCoMo Inc. Information on i-mode Technical Topics - GPS, November 2007.
- [17] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. Codes for the Representation of Names of Countries and their Subdivisions. ISO 3166, November 2001.
- [18] IAN JACOBS and NORMAN WALSH. Architecture of the World Wide Web, Volume One. World Wide Web Consortium, Recommendation REC-webarch-20041215, December 2004.
- [19] TIM KINDBERG, JOHN BARTON, JEFF MORGAN, GENE BECKER, DEBBIE CASWELL, PHILIPPE DEBATY, GITA GOPAL, MARCOS FRID, VENKY KRISHNAN, HOWARD MORRIS, JOHN SCHETTINO, BILL SERRA, and MIRJANA SPASOJEVIC. People, Places, Things: Web Presence for the Real World. *Mobile Networks and Applications*, 7(5):365–376, October 2002.
- [20] TIM KINDBERG and SANDRO HAWKE. The 'tag' URI Scheme. Internet RFC 4151, October 2005.
- [21] F. G. LEMOINE, S. C. KENYON, J. K. FACTOR, R.G. TRIMMER, N. K. PAVLIS, D. S. CHINN, C. M. COX, S. M. KLOSKO, S. B. LUTHCHE, M. H. TORRENCE, Y. M. WANG, R. G. WILLIAMSON, E. C. PAVLIS, R. H. RAPP, and T. R. OLSON. The Development of the Joint NASA GSF and the NIMA Geopotential Model EGM96. NASA/TP-1998-206861, July 1998.
- [22] ALEXANDER MAYRHOFFER and CHRISTIAN SPANRING. A Uniform Resource Identifier for Geographic Locations ('geo' URI). Internet Draft draft-mayrhofer-geo-uri-02, February 2008.
- [23] NATIONAL IMAGERY AND MAPPING AGENCY. Department of Defense World Geodetic System 1984. NIMA TR8350.2, Third Edition, January 2000.
- [24] MARK NOTTINGHAM and ROBERT SAYRE. The Atom Syndication Format. Internet RFC 4287, December 2005.
- [25] OPEN GEOSPATIAL CONSORTIUM. An Introduction to GeoRSS: A Standards Based Approach for Geo-enabling RSS feeds. OGC 06-050r3, July 2006.
- [26] STEVEN PEMBERTON. XHTML 1.0: The Extensible HyperText Markup Language (Second Edition). World Wide Web Consortium, Recommendation REC-xhtml1-20020801, August 2002.
- [27] JON PETERSON. A Presence-based GEOPRIV Location Object Format. Internet RFC 4119, December 2005.
- [28] JAMES M. POLK, JOHN SCHNIZLEIN, and MARC LINSNER. Dynamic Host Configuration Protocol Option for Coordinate-based Location Configuration Information. Internet RFC 3825, July 2004.
- [29] DAVE RAGGETT, ARNAUD LE HORS, and IAN JACOBS. HTML 4.01 Specification. World Wide Web Consortium, Recommendation REC-html401-19991224, December 1999.
- [30] JOSHUA SCHACHTER and ASK BJØERN HANSEN. The GeoURL ICBM Address Server.
- [31] MÁRIO J. SILVA, BRUNO MARTINS, MARCIRIO SILVEIRA CHAVES, ANA PAULA AFONSO, and NUNO CARDOSO. Adding Geographic Scopes to Web Resources. *Computers, Environment and Urban Systems*, 30(4):378–399, July 2006.
- [32] RAINER SIMON and PETER FRÖHLICH. A Mobile Application Framework for the Geospatial Web. In *Proceedings of the 16th International World Wide Web Conference*, pages 381–390, Banff, Alberta, May 2007. ACM Press.
- [33] UNITED STATES CODE. Wireless Communications and Public Safety Act of 1999 (911 Act). Pub. L. No. 106-81, 113 Stat. 1286, October 1999.
- [34] WAP FORUM. Wireless Application Protocol — Architecture Specification. WAP-210-WAPArch-20010712, July 2001.
- [35] WAP FORUM. Wireless Application Protocol — Location Protocols. WAP-257-LOCProt-20010912-a, September 2001.
- [36] STUART L. WEIBEL, JOHN A. KUNZE, CARL LAGOZE, and MISHA WOLF. Dublin Core Metadata for Resource Discovery. Internet RFC 2413, September 1998.
- [37] ERIK WILDE. The Plain Web. In *Proceedings of the Web Science Workshop (WSW2008) at WWW2008*, Beijing, China, April 2008.