Location-aware Content Delivery Service Using Publish/Subscribe

Alisa Devlić and Ivana Podnar
Department of Telecommunications
Faculty of Electrical Engineering
University of Zagreb
Unska 3, HR-10000 Zagreb, Croatia
[alisa.devlic, ivana.podnar]@tel.fer.hr

Abstract – The delivery of location-aware content tailored to user location and personal requirements is going to be the premier service in next generation mobile networks. This service should provide mobile users with up-to-date information according to their preferences and current geographical location. In this paper we propose an architecture for the location-aware content delivery service. The architecture comprises a publish/subscribe component which supports push-based delivery of data to interested users, a presence service for defining and updating user presence information, and a profile handler for storing and updating user profiles. The presence service interacts with the location management component that can locate the applied user terminal. First, we explain the concept of publish/subscribe and discuss its applicability to location-aware content delivery. Next, we describe the presence and profile management service with the applied solution for location management. Additionally, we define the structure of presence information and personal user profiles. Finally, we present a scenario showing the interaction between our components and conclude with the discussion of service implementation.

Keywords: location-aware service, content dissemination, publish/subscribe

I. INTRODUCTION

Location-aware content delivery seems to be an inevitable part of the future personalized services for mobile users. The first step towards the realization of such services is the delivery of personalized content that conforms to user subscription preferences, the current location and presence information, and the applied end user terminal. All the information that guides the process of content delivery is stored in a user profile. The profile contains user subscriptions that define what type of content in interesting to a user at different locations, user presence information about the current state and activity, and device capability information about the characteristics of applied devices. As the user moves, the application constantly tracks his/her location and decides weather or not to send the available content to the applied end terminal. The application can also get the information about the user's preferred communication means using the presence information which knows where to send the appropriate information. The presence information offers means for personal mobility.

To enable efficient and scalable content delivery to numerous users, and at the same time offer content filtering, we propose the use of publish/subscribe middleware. The publish/subscribe infrastructure enables publishers to publish the information through an intermediary, and allows subscribers to declare interest in certain types of information.

In this paper we present the service architecture for location-aware content delivery. In Section II we describe the concepts of publish/subscribe. The components that are necessary for service personalization, locating the user and tracking the presence information are outlined in Section III. The detailed description of our architecture and the interaction between the components is given in Section IV. In Section V we compare our approach with the related work and give our conclusions and directions for future work in Section VI.

II. PUBLISH/SUBSCRIBE CONTENT DELIVERY

The publish/subscribe approach to content delivery enables the delivery of information from information sources to numerous users across a wide area network. The content being distributed is non-real time multimedia content such as plain text, image, or video clips. The publish/subscribe delivery service involves two types of users: publishers and subscribers. Publishers define the content that is submitted to the service for the subsequent delivery to subscribers. Subscribers define subscriptions that describe the type of content they are interested in receiving. The publish/subscribe service can therefore be visualized as an information bus that joins a publisher when publishing some content with interested subscribers, as depicted in Figure 1.

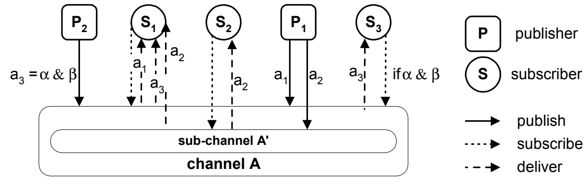

Figure 1 Publish/subscribe interaction

Figure 1 shows two publishers P_1 and P_2 that publish notifications on channel A with a corresponding sub-channel A'. Subscriber S_1 subscribes to all the content published on channel A. When publisher P_1 publishes a notification a_1 on channel A, a_1 will be delivered to the subscriber S_1.When P_1 publishes a notification a_2 on the sub-channel A', a_2 will be delivered to S_2 that is subscribed to the sub-channel A', but also to S_1. A weather channel "Croatia weather" with a sub-channel "Zagreb weather" is a real-world example demonstrating this type of subscription. In literature they are referred to as *subject-based*, *topic-based*, or *channel-based* subscriptions [8].

Subscriber S_3 in Figure 1 is selective regarding the content it wants to receive from publishers and it specifies its subscription as a condition. When publisher P_3 publishes a notification a_3 that matches the condition specified by S_3, a_3 will be delivered to S_3 but also to S_1 since S_1 is interested in all the content published on channel A. A user interested in extreme weather conditions in Croatia would define a subscription to the channel "Croatia weather" with a filter matching snowy conditions and temperature values below zero. With such a subscription the user would only get notifications containing the word "snow" and satisfying a condition that the temperature is less then zero. This type of subscription describes the content of the notification that should be delivered to subscribers and is therefore called *content-based* subscription [8].

In the content delivery service we combine channel-based and content-based subscriptions and assume that a user subscribes to a channel with a list of keywords that will be used for filtering the content published on the channel. Our subscriptions are location-aware: the subscription contains a list of locations for which it is valid. Each subscription for our location-aware content delivery service is composed of a channel identifier, an optional list of keywords, and an optional list of locations.

For example, a subscriber to "Croatia weather" can specify the subscriptions presented in Figure 2. The subscriber wants to receive notifications published on the channel "Zagreb weather" when he/she is in Zagreb and Dubrovnik, but he/she is interested in notifications published on the channel "Dubrovnik weather" that contain the word "rain" only when visiting Dubrovnik.

```
<subscription>
       <channel_name>Zagreb weather
       </channel_name>
       <location>Zagreb</location>
       <location>Dubrovnik</location>
</subscription>
<subscription>
       <channel_name>Dubrovnik weather
       </channel_name>
       <keyword>rain</keyword>
       <location>Dubrovnik</location>
</subscription>
```

Figure 2 Example of a location-aware subscription

When implementing a publish/subscribe push-based service, it is important to note that the service needs to keep track of the processes running on subscribers terminals that can receive data packets carrying notifications. When delivering a notification to a subscriber's terminal, the service becomes a client that contacts a server running on the terminal. In case the terminal changes its attachment point in the IP-based network, its IP address will change – unless using Mobile IP – and the service needs to update this information. It is also possible that a user employs different devices for receiving notifications: In this case the service needs to support personal mobility [11], i.e. regard a user as an end communication point and keep track of the device he/she is currently using. Additionally, to implement a location-aware service, we also need a component that can locate the subscriber. Therefore in our system we use a presence component that tracks the employed user device and the current user location.

III. PRESENCE AND PROFILE MANAGEMENT

The presence service accepts, stores, and distributes the presence information about mobile users - presence entities. The presence information describes the current status of a presence entity regarding its communication capabilities, for example, a user can be "online" or "offline". The presence status of a user is necessary for location-aware content delivery because the content needs to be delivered to the currently applied user device in accordance with the current communication status. For example, if a user is currently in a meeting and the presence status is "busy", the content delivery service can decide not to send the time-sensitive information to his/her terminal.

The presence information is also important for instant messaging, a new medium for communication over the Internet [5][3], [6]. The currently developed applications that implement presence and instant messaging functionalities use non-standardized and non-interoperable protocols developed by various vendors. The goal of the Instant Messaging and Presence Protocol (IMPP) working group within the IETF is to define a standard protocol and message format in order to support independently developed applications that can interoperate across the Internet. We have decided to follow the recommendations of the IMPP regarding the presence service. The instant messaging functionality is not under the scope of this article.

The presence functionality works as follows: the presence entities (*presentities*) provide their presence information to the presence service. The presence service accepts, stores, and distributes the presence information to everyone subscribed to get the notifications about presence changes. The so called *watchers* receive the presence information from the presence service. Presentities and watchers communicate by exchanging presence events: presentites are publishers in the publish/subscribe model, and watchers are subscribers that are interested in presence events.

The presence events are given in the form of XML documents that comply with at least a *common base format* for presence information: the presentity's unique identifier (the "pres" URL), and a list of presence "tuples" as depicted in Figure 3. The <tuple> elements are used for segmenting the presence information, because the components of the full presence information can come from distinct devices, or different applications at the same time. A <tuple> element must contain a unique 'id' attribute to distinguish itself from other "tuples" in the same XML document. Each <tuple> element consists of one mandatory basic <status> element which describes the status information of the presentity, e.g. online, offline, busy, away, or, do not disturb. The status element is followed by a number of *optional extension attributes* from other namespaces, and an optional <contact> element, which describes the communication address of the presentity, e.g. an email address, telephone number, postal address, with relative priorities of preferred communication means. Further on follows a number of optional <note> elements containing human readable comments about the tuple or presentity, and an optional <timestamp> element, which describes the time of the change of this tuple.

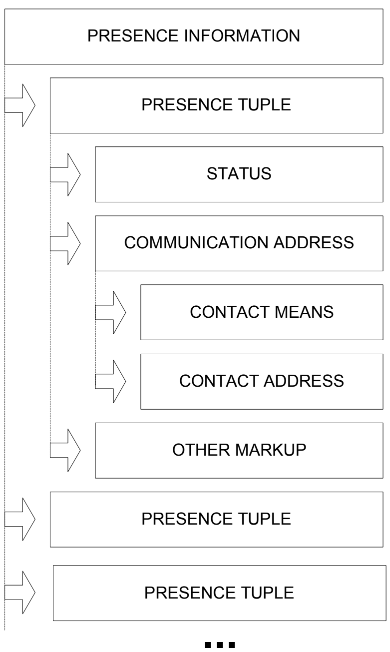

Figure 3 Presence information structure

The Wireless Village Initiative [18] has defined a list of DTDs for presence attributes initiated from client, server or presence service which can be part of the *extended presence information format*. A presence attribute in general contains the presence information intended for the user. It may also contain the meta-information for machine-to-machine communication between a publishing client and receiving clients.

An example of a presence attribute usage in our application is shown Figure 4. The `PresenceSubList` is a list of attribute names with the empty values, used as a reference when requesting/subscribing presence attributes. Our example contains a subset of the list of attributes defined with the Wireless Village Initiative, needed for the purpose of the application.

```
<PresenceSubList
 xmlns=
 "http://www.wireless-village.org/PA1.0">
      <OnlineStatus/>
      <ClientInfo/>
      <GeoLocation/>
      <FreeTextLocation/>
      <PLMN/>
      <UserAvailability/>
      <PrefferedContacts>
      <PrefferedLanguage>
<PresenceSubList>
```

Figure 4 Example of presence attribute usage in our application

The listed attributes contain the information about the logged status of the client, the type of the device he/she is currently using, current location information, PLMN name of the mobile network where the client device is registered, the availability for communication, preferred contacts, and preferred languages. Those attributes are stored as optional extension attributes from other namespaces in the presence information following the <status> element. Here the subscriber provides user-specific parts of the presence information, some parts are obtained automatically from the network or user terminal, and the location management provides the location information at regular intervals.

The location management component determines the location of a subscriber. It is developed as a communication between a client and a location management server that exchange HTTP requests for location information and replies [7]. The parameters needed to start the procedure for getting the location information are username, password and a mobile phone number. This data is encapsulated in an XML-formatted request, the HTTP connection is opened and the request is send via a HTTP POST method to a Mobile Positioning Server. The Mobile Positioning Server receives the request, determines the location and sends it back to the client in an XML positioning result complying with the WGS-84 (World Geodetic System 1984) format [19]. The Mobile Positioning Server can also be configured to send the location information via a push mechanism in predefined intervals, and therefore track user's moving.

When the location information changes, the presence information will be modified and the profile database has to be notified and updated with this new presence information. Because the location information is presented in the WGS-84 format, the location to name mapping has to be performed either via interaction with a GIS database or by defining a mapping the range of latitude and longitude values with location names.

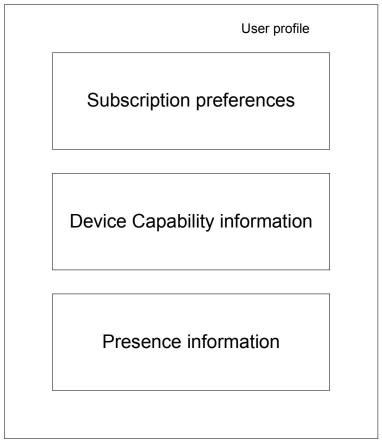

Figure 5 User profile

A user profile, as shown in Figure 5, consists of subscription preferences regarding location, device capability data and presence information, following the CC/PP document [4] and WAP User Agent Profile [17]structure and semantics defined with RDF schema [16]. The matching process of the current user subscription with the location name of the positioned area will be performed in order to determine and process his active subscription. This subscription is then needed in publish/subscribe content delivery service.

IV. SERVICE ARCHITECTURE

The service architecture is composed of two main parts: the location-aware presence and profile management part, and the publish/subscribe-based content delivery part. They can function independently as two separate services, if needed.

The location-aware presence and profile management consists of three components that take care of user's subscriptions and presence information: the presence service, the location management component, and the profile handler. The presence service stores and updates the presence information about registered users and communicates with the location management component that is in charge of locating users. The profile handler stores user profiles: In case of content delivery profiles contain user subscriptions. Each subscription contains the name of the subscribed channel, an optional list of keywords for content-based subscription to the specified channel, and an optional list of locations for which this subscription is valid.

Presence information is introduced because a subscriber can use various devices at different locations and the application needs to track him/her when delivering the information. The subscriber is identified with the unique identity ("pres" URL), while the information about the device he/she is currently using is written in the "ClientInfo" attribute in presence information. The presence service will filter out the periodical notifications about the user location and notify other system components when the user location significantly changes.

The interaction between the components that are responsible for presence and profile management is given in Figure 6. When first starting to use the location-aware service, a user defines his/her profile and subscriptions regarding location. The profile handler stores this information in the user profile that contains previously defined device capability information. The profile handler wants to be notified about the changes of the user presence information, and therefore subscribes to the channel "Presence update" giving user id as a keyword. In the next step the user registers with the presence service and provides his/her presence information attributes. The presence service component will send a request to the location management component to constantly track the user and to inform the presence component through its push mechanism about the current user location. The presence service component will publish each significant change of user location together with the associate presence information using the publish/subscribe middleware. The profile handler will be notified about the changes of presence information and, as a result, it will update its profile.

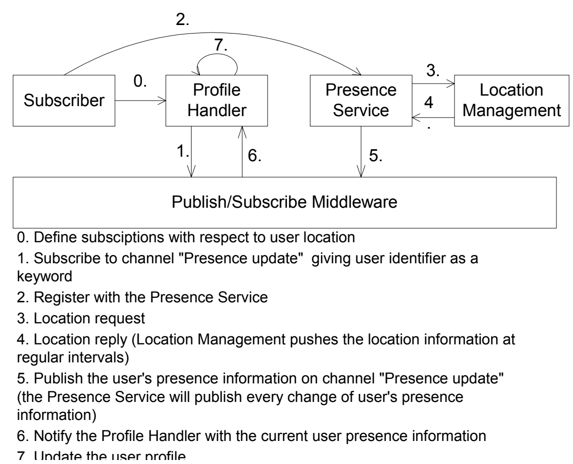

Figure 6 Location-aware presence and profile management

The content delivery part of our architecture involves the publish/subscribe infrastructure and the profile handler. The publish/subscribe infrastructure is responsible for efficient content delivery from content publishers to subscribers, and the profile hander is responsible for storing and updating user subscriptions, and presence information. The delivery functionality can be split into two phases: subscription activation during application startup and subscription update for an existing subscription during application runtime.

The process of subscription activation is depicted in Figure 7. It can be seen as a pull mechanism, where a user (subscriber in terms of publish/subscribe) requests the subscription data from the profile handler component. The profile handler replies with the subscription data that depends on the current user location. The data in the profile are kept current using the mechanisms described in Figure 6. Next, the subscriber subscribes to a channel using the received subscription. When a publisher publishes some new content on the stated channel, the notification will be delivered to the subscriber.

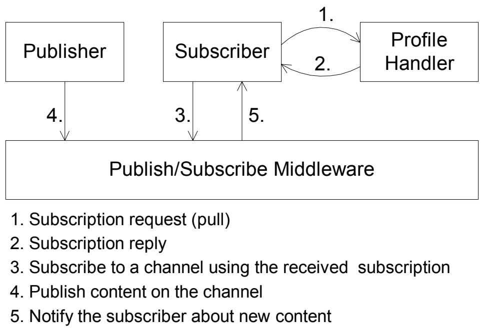

Figure 7 Subscription activation – pull style

The subscription update process for an active subscription is implemented through the push mechanism as presented in Figure 8. The subscription update is needed when a user changes the location which influences his/her subscription. In this case the application running on the user's terminal needs to be informed about the change of subscription. The profile handler will push the new subscription to the subscriber; the subscriber will terminate the existing subscription and subscribe to another channel according to the new subscription. The subscriber will be notified when the content published on the channel(s) matches the new subscription.

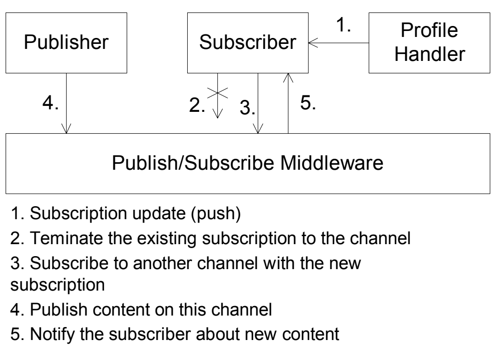

Figure 8 Subscription update – push style

V. Related Work

The location-aware content delivery is gaining more interest with the appearance of personalized services that adapt the content to individual user needs and current locations. However, to our knowledge only a few systems are designed for selective personalized location-aware delivery of content to mobile users.

The position paper [10] argues that such services are important and valuable, and sketches the architecture of the selective content dissemination prototype that is based on the publish/subscribe interaction. The main component of the presented architecture is the filtering engine for efficient matching of notifications to user subscriptions. The architecture assumes the existence of a location server and the use of different transport mechanisms for content delivery. However, the description of the location service is not provided. The authors predict that the subscription language and the efficient filtering engine are crucial to the service, however, they do not address the specifics introduced by mobility and location management.

The detailed discussion of problems related to the implementation of a publish/subscribe-based content delivery in mobile environments is given in [15]. The paper proposes an architecture that is personalized and mobility-aware. This paper further extends the proposed architecture by making it location-aware.

The GigaMobile project from Telematica Institut [9] focuses on the customization of end-user mobile services, which includes adaptation of end-user services to personal preferences, location, network and terminal capabilities, and the entire context in which the service is used. They have developed a concept called the Personal Service Environment (PSE), combining three generic middleware services to form the core components for the personalization of end-user services in mobile and wireless networks. These generic services are: personalization (location-awareness, user preferences and profile service adaptation), QoS adaptation (adaptation of the service to the quality of service of the network and the terminal of user) and brokerage (matching the user preferences with the service characteristics).

The concept of PSE is part of the VHE specifications of 3GPP [1], which is a concept for service personalization focused on portability of personalized information across network boundaries and between terminals. Their concept of PSE has been extended from GigaMobile project, including service discovery and adaptation.

The work on (semi-) standards that incorporate profiles has been done from the following organizations: 3GPP/VHE (3GPP, 2001) [1], MPEG Community (MPEG-7) [12], PAM Forum (2001) [13], and Cameleon (1998) [3]. 3GPP/VHE specifies the distinction of two groups of profiles: User_Interface_Profile and User_Service_Profile. Profiling is still an ongoing work within the 3GPP. The MPEG Community defines multimedia content schemes (metadata) with possibilities of tailoring content to terminal characteristics and user preferences, personalized filtering and searching and browsing of audio-visual content. PAM Forum has defined a set of specifications for Presence and Availability Management, with a goal of reaching a standard on digital identities, characteristics and presence's status of agents, capabilities and state of entities, and availability of entities for various forms of communication. CAMELEON is a European project, carried out between 1998-2000., VHE concept compatible, that made use of agent technology for personalizes service portability. They cover adaptive profile management, advanced directory service, virtual address book, flexible financial services, and personal telecommunication assistant.

In the telecommunications field the problems related to mobility and location-awareness are well understood. However, wireless networks are closed systems that offer limited possibilities for the design, implementation, and deployment of novel personalized services such as location-aware content dissemination. Currently, the Short Message Service (SMS) for content distribution to mobile phones is widely used in cellular mobile networks. SMS offers the so-called *cell broadcast service* [14] for transmitting notifications such as traffic conditions, weather forecasts, or stock quotes. The cell broadcast can distribute messages to users who subscribe to a particular channel, but offers no filtering. SMS is an extremely popular messaging service, but limited by the low bandwidth communication channels and closed proprietary solutions. Multimedia Messaging Service (MMS) [2], the successor of SMS, provides non-real time multimedia messaging in 2.5G and 3G networks. The MMS specification currently focuses on one-to-one usage scenarios, but considers one-to-many communication scenarios that offer content delivery to numerous users. We consider using SMS and MMS as transport mechanisms for content delivery that need to be combined with the location and presence information to offer a personalized and selective location-aware content delivery.

VI. Conclusion and Future Work

We have presented the architecture for the location-aware content delivery service that enables the delivery of personalized content to users depending on their location and preferences. The architecture uses the publish/subscribe infrastructure as an efficient and customizable content transportation mechanism which supports the multicast-type communication between content publishers and subscribers. The architecture provides service personalization by enabling users to define personal profiles that will store user subscriptions, device capability information, and presence information. The service presence and location-awareness is realized by designing the presence service that communicates with the location management component.

Up to now we have designed and implemented the location management component that is described in the paper and are currently developing a dissemination component that is based on the principles of publish/subscribe. We plan to implement the proposed presence service and profile handler, and integrate components into a prototype system for personalized location-aware content delivery.

References

[1] 3GPP TSG SA, *The Virtual Home Environment*, 3GPP, 3G TS 22.121 version 5.00. (version 5.1.0: ftp://ftp.3gpp.org/Specs/2001-09/Rel-5/22_series/22121-510.zip)

[2] 3GPP TS 23.140. Multimedia Messaging Service (MMS): Functional description. Stage 2, Release 4, March 2001.

[3] CAMELEON project: C.Gőrg, et al., Specification of Prototypes, Release 2, Deliverable D08, July 1998., See: http://www.comnets.rwth-aachen.de/~cameleon/cameleon/D08_FINAL.zip

[4] *Composite Capabilities/Preference Profiles*, World Wide Web Consortium, http://www.w3.org/Mobile/CCPP/

[5] M. Day, J. Rosenberg, and H. Sugano. A Model for Presence and Instant Messaging, RFC2778, February 2000.

[6] M. Day, S. Aggarwal, and J. Vincent. Instant Messaging/Presence Protocol Requirements, RFC2779, February 2000.

[7] A. Devlić, A. Graf, and D. Huljenić. Developing a Framework for Mobile Positioning Client on PDA Platform, *Proceedings of the Conference on Telecommunications (MIPRO), 2002.*

[8] P.Th. Eugster, P. Felber, R. Guerraoui, and A.-M. Kermarrec. The Many Faces of Publish/Subscribe. Technical report TR-DSC-2001-04. Distributed Programming Laboratory, Swiss Federal Institute of Technology, January 2001.

[9] GigaMobile Project, Telematica Institut, http://www.telin.nl/Middleware/GIGAMOBILE/ENindex.htm

[10] H.-A. Jacobsen. Middleware Services for Selective and Location-based Information Dissemination in Mobile Wireless Networks. *Advanced Topic Workshop on Middleware for Mobile Computing*, (Heidelberg, Germany), Nov. 2001.

[11] P. Maniatis, M. Roussopoulos, E. Swierk, K. Lai, G. Appenzeller, Xinhua Zhao and M. Baker. The Mobile People Architecture. *ACM Mobile Computing and Communications Review*, **3**(3), July 1999.

[12] MPEG-7., International organization for Standardization, MPEG-7 Overview (version 8), ISO/IEC JTC1/SC29/WG11/N3705, July 2002. http://mpeg.telecomitalialab.com/standards/mpeg-7/mpeg-7.htm

[13] PAM (Presence and Availability) Forum , PAM Specification Document, version 1.0, 11 September 2001., http://www.pamforum.org/Docs/PAMspec_1.0.pdf

[14] C. Peersman, S. Cvetkovic, P. Griffiths, and H. Spear. The Global System for Mobile Communications Short Message Service. *IEEE Personal Communications*, **7**(3):15—23, June 2000.

[15] I. Podnar, M. Hauswirth, and M. Jazayeri. Mobile Push: Delivering Content to Mobile Users. Proceedings of the International Workshop on Distributed Event Based Systems in conjuction with International Conference on Distributed Computing Systems. IEEE CS Press, Los Alamitos, Calif., pages 563—568, (Vienna, Austria), July 2002.

[16] *Resource Description Framework,* World Wide Web Consortium, http://www.w3.org/RDF/

[17] Wireless Application Forum, http://www.wapforum.org/

[18] Wireless Village Initiative: Presence Attributes, DTD and Examples, Version 1.1, http://www.openmobilealliance.org/wirelessvillage/docs/WV_PA_DTD_v1.1.pdf

[19] World Geodetic System 1984, http://www.wgs-84.com