Design of a Web-based Distributed Location-Aware Infrastructure for Mobile Devices

Vlad Trifa, Dominique Guinard
Institute for Pervasive Computing, ETH Zurich
and SAP Research CEC Zurich
8092 Zurich, Switzerland
trifa@inf.ethz.ch

Philipp Bolliger, Samuel Wieland
Institute for Pervasive Computing, ETH Zurich
8092 Zurich, Switzerland

***Abstract*—Since GPS receivers have become a commodity anyone could access and use location information simply and freely. Such an easy access to ones' location is instrumental to development of location-aware applications. However, existing applications are static in that they do not model relations between places and mobile things. Moreover, these applications do not allow to easily map the physical location of mobile devices to virtual resources on the Internet. We attempt to bridge this gap by extending the base concepts that make up the Internet with the physical location of devices, in order to facilitate the development of Web-based location-aware applications for embedded mobile devices. In this paper, we describe and evaluate a simple infrastructure for the "Web of Things" that extends the existing Web to enable location-aware applications. By introducing the concept "here/*", we enable a naturally hierarchic way to search for location-aware devices and the services they provide.**

***Keywords*-Web of Things; REST; HTTP; API; location-based services.**

I. Introduction

In the last decade, a fundamental paradigm shift has been taking place in the field of computing technologies. Thanks to the miniaturization of embedded systems, tiny computers with wireless communication abilities can be easily integrated into everyday objects, thus turning them into *smart objects*. According to the IP for Smart Objects Alliance (IPSO)[1], an increasing number of embedded devices will be supporting the IP protocol, so that many physical objects will be connected to the Internet. The convergence of physical computing devices (Wireless Sensor Networks, mobile phones, etc.) and the Internet provides new design opportunities, as digital communication networks will soon not only contain virtual data (images, text, etc.), but also real objects. On top of that, these objects will very likely become first class citizens of the World Wide Web, which will make them linkable, discoverable, searchable, therefore usable just like any other data available on the Web.

On the Web, the location of data is irrelevant, since a powerful mechanism is in place for accessing data regardless of where it is stored (URI). In contrast, physical things are always somewhere, and their location can change quickly. In addition, people also often change location (home, office, car, etc), and the use of centralized repositories to store the current location of all mobile things would not scale. To fully leverage the physical nature of objects one needs to know their location. Localization of objects and people has always been a tedious technical challenge, and only recently GPS receivers have become a commodity which allowed people to localize objects more easily. While GPS can be used for localization in outdoor environments in most situations, there are still some limitations to this technology. When it comes to indoor applications, GPS becomes inherently unusable, thus indoor localization system using WiFi fingerprinting [1], [2] have been particularly popular as they require no new hardware infrastructure installation for sites that already have WiFi, and with resolution to a few meters, it can support room-level localization. As it has been shown in surveys of ubiquitous computing, it is sufficient to localize a user within room-level accuracy for almost all applications [3]. Although spatial localization techniques improve over time, wide spreading of location-aware applications is still prevented by the lack of robust and open standards for modeling and representing locations on the Web beyond geographical coordinates [4]. Due to the lack of support for modeling the physical location of things, discovering devices present in a location and interacting with them in an ad-hoc manner is a complex problem that requires customized applications, especially because the available low-devices and protocols are usually incompatible. While solutions such as Bluetooth, Apple's Bonjour or Universal Plug and Play do offer powerful mechanisms for locating similar devices on the same network; a common ground on which devices using different protocols could interact globally and transparently is still missing.

To overcome this problem, we propose a Web-based infrastructure that simplifies the interconnection of heterogenous embedded devices and that takes into account the hierarchical structure of places and the mobile devices therein. For this, we propose using lightweight software entities called gateways that are used to extend the Web to the real-

[1] `http://www.ipso-alliance.org/`

world by offering a Web interface to all kinds of embedded devices that do not necessarily support the IP protocol. Gateways are designed to be linked with other gateways to form a hierarchical tree structure mapped to physical locations (for example place concepts in a building such as floors, rooms, etc) which allows users to interact directly with all the devices present in any particular location. Based on this Web-based hierarchical place model, we introduce the concept `here/*` which allows any Web client to use URIs as a flexible context-aware search method to find and use things in specific locations. Finally, we describe a prototype to illustrate how such an infrastructure allows users to develop quickly location-based services for physical things, only by using the Web as development platform.

II. Related work

There have been several attempts to integrate physical objects into the Web [5], [6], [7]. Unfortunately, most of previous projects merely proposed solutions to link real objects to their representation on the Web without using the founding principles of the Web as core concepts. Because of that, early work reduced the role of the Web only as a transport protocol, instead of using it as an application protocol. This prevented devices and their functionality to considered as *real* Web resources that can be searched, browsed and used just like any other Web page. More recent projects [8], [6] have investigated how embedded devices can accessed while using the REST paradigm, but they mainly focused on isolated systems and didn't address the problem of scalability of the system. The term Sensor Web refers to a global network of Web-connected sensors, and several projects have been proposed to build such a worldwide Sensor Web, as for example MSR Senseweb [7], Irisnet [9]. In these projects, a unique endpoint is used to register and store data collected by many physical sensors, however such a central point of failure is against the distributed nature of the Web. While still a centralized server, Pachube [10] offers a promising solution which is simpler to use as sensor data can be published and reuse sensor data by using RSS feeds a RESTful interface.

A major problem when it comes to integrate physical object to the Web is that no Web standard explicitly designed for physical object is available. Because of that, the Open Geospatial Consortium has propose the Sensor Web Enablement [11] initiative that focuses on developing standards to enable the discovery and exchange of sensor observations between Web-connected sensors.

While GPS can be used for localization in outdoor environments in most situations, there are still some limitations to this technology, as for example it's inaccuracy in urban canyons. When it comes to indoor applications, GPS becomes inherently unusable. Thus, indoor localization systems have been popular for several years [12], [13], [1], [14]. In this domain, WiFi fingerprinting [12], [1] has been particularly popular as it requires no new hardware infrastructure installation for sites that already have WiFi, and with resolution to a few meters, it can support room-level localization. To achieve such high accuracy from the noisy WiFi signal, however, such systems require prior manual calibration. To overcome this problem Bhasker et al. first formulated the idea of collecting calibration data during use [15]. Our own system Redpin [2], also collected calibration information from end-users, but in contrast to Bhasker et al. omits a geometric algorithm beforehand. Using a modified version of k-nearest neighbors, Redpin generates *symbolic* location identifiers with room-level accuracy. As it has been shown in surveys of ubiquitous computing, it is sufficient to localize a user within room-level accuracy for almost all applications [3], [16], [17].

Although there is no formalized standard on how to model indoor locations, one can find many location models in the literature [18], [19], [20]. However, most of these location models have been designed to match specific application need as they were designed solely for specific projects. The NEXUS project proposed an open platform for context-aware applications, with a special focus on location [18]. Their location model is explicitly defined and supports hierarchical naming schemes and different levels of detail for indoor and outdoor applications.

To represent location in Carnegie Mellon's AURA project [21], Jiang and Steenkiste introduced a hybrid location model and a formal representation that combines the advantages of symbolic and geometric location models [22]. The clear separation of model and representation is what separates this approach from others. Consequently, the *AURA Location Identifier (ALI)* uses a formatted, Universal Resource Identifiers (URI) compliant string to represent all the above concepts. Other approaches for location modeling on the Web have been proposed to build a geographic web that merges abstract information with geographical [23], [24], unfortunately they were designed primarily for spatial locations and are not suited for hierachical models. Ubisworld [25] describes an interesting Web-based hierarchical model for locations, however not explicitly designed for real devices, and the project seems discontinued. Recently, the Locative Web [4] has offered interesting insights towards turning the Web into a location-aware infrastructure.

III. Web-oriented Infrastructure for Physical Things

The success of the World Wide Web is rooted in its particular software architecture called Representational State Transfer (REST) [26], which emphasizes scalability, generic interfaces, and a loose coupling between components. On the Web, the primary abstraction of information and functionality are *resources* identified by Uniform Resource Identifiers (URIs) and are manipulated using the HTTP verbs GET, POST, PUT, DELETE. Although HTTP was designed as an application protocol with particular strengths (and weaknesses), many Web applications use HTTP as a transport protocol by using only a small fraction of it. This prevent to take full advantage of the Web because it requires to define specific application layers for application. For example, Web applications that rely upon Web services based on SOAP and WSDL use only one operation of HTTP (POST) to call API methods offered by a few URI-identified endpoints, thus hiding the actual resources being manipulated.

The term *Internet of Things* refers to networked devices with an emphasis on interoperability at the data transport layer. We propose the notion *Web of Things* as a refinement of the Internet of Things where emphasis is shifted towards interoperability at the application layer. In the Web of Things, the founding principles of the Web architecture (REST) are fully leveraged to support a location-aware infrastructure for networked devices. Based on the success of Web 2.0 mashups, we propose a similar lightweight approach for interacting with embedded devices using HTTP to manipulate URI- identified resources, therefore enabling the creation of *physical mashups*.

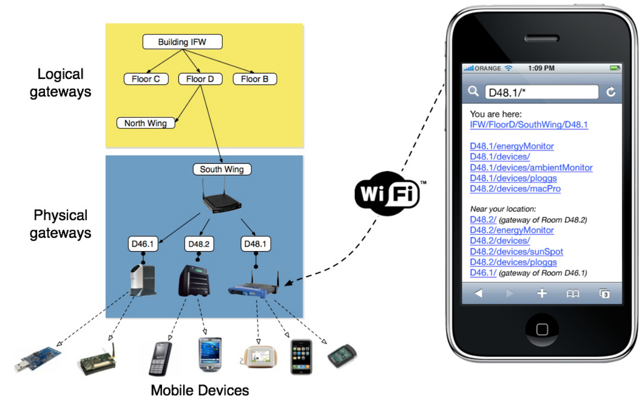

Figure 1. Example gateway hierarchy from a building. The top gateway covers the gateways for each floor, and is composed only of logical gateways. The South Wing gateway is implemented on a wireless router, which is connected to the computer sub-network of that area and has access to all the gateways present in the rooms (which are implemented on computers located in rooms). The gateways in each room have different physical interfaces to access mobile devices.

A. Hierarchical Location Modeling

Embedded devices have usually limited resources, therefore require optimized protocols to exchange data. As HTTP or IP might not be available or would be computationally expensive, we propose to use gateways to enable uniform Web-based interaction with low-power devices. A gateway is a lightweight software module that is used to connect to the Web many kinds of embedded devices connected to the gateway using an appropriate physical interface such as Bluetooth or Zigbee. To realize a scalable and distributed location-aware infrastructure for devices, gateways were designed so that they can be linked together to form a virtual network of nodes that can then be mapped onto physical places to model relations between real places. As illustrated in Figure 1, using such a tree of nested gateways one can easily create a hierarchy of places with different levels of granularity (country, region, city, street, building, floor, room, object). Each level of the tree offers an abstract layer for interacting with the devices and services under it, and each node refines its parents by offering a finer granularity.

We distinguish two types of gateways; logical and physical gateways. The same software is used in both cases, however logical gateways do not necessarily need to be installed on a computer physically located in the area it is mapped to. For example, the logical gateways tree presented in Figure 1 can be fully distributed across servers anywhere in the world in a transparent manner, as long as the logical structure of the tree is maintained. In contrast, physical gateways (or *terminal* gateways) must be embodied into a physical computer located in the area it corresponds to, and can posses different interfaces to connect with devices in the real world. The gateway software was designed to be easily implemented on any programmable computer with a TCP/IP connection, such as open wireless routers, Network Attached Storage (NAS), media players, etc.

The mapping process that assigns the logical place name (room 44, floor D, east wing, etc.) to gateways must be done manually by the developer at setup time. Fortunately, since gateways are not mobile and the structure of their connections is rather static, little effort is required to maintain the tree structure. Terminal gateways can discover mobile devices in their surroundings and make them dynamically available as Web resources accessible over HTTP. This allows to navigate the tree by following links to surrounding gateways simply by clicking the links on a Web page or typing a URL in any Web browser.

Using this approach, we can easily build a system that supports range and lookup queries for mobile devices. Unlike other hybrid models for spatial queries, our approach does not require any centralized component or database to store information about the system. This is possible because the gateways are loosely-coupled and independent components responsible for managing all the devices (and gateways) located in the area they are associated with. The higher in the hierarchy, the more seldom things change, which naturally forms an efficient load-balancing system because users only access gateways located in the area of interest without soliciting the rest of the system. By leveraging well-known Web technologies, the infrastructure is more scalable while supporting mobile devices that can change location dynamically.

B. Localization

Once the spatial hierarchy formed by the gateways has been setup, the problem of mapping the current location of a user to a particular node in the tree persists. The spatial localization process is not part of our project and for our requirements one can rely on an indoor localization system to determine one's location at the room level (such as RedPin [2]).

When gateways are sparse enough so that their radio coverage does not overlap, the mapping between a device and a place can be determined simply by the connectivity between the device and the nearest physical gateway. However, when several gateways are located nearby, how does the system know which one to choose? We call this the bootstrap problem, and a simple method to retrieve the URL to connect to the correct gateway is necessary. To avoid specific software or hardware, we enable gateways to be *discoverable* on the local network. Given that HTTP does not define any discovery mechanism (on the Web resources are discovered by following URIs), we have implemented a lightweight discovery protocol used by our gateways, similar in function to Apple's Bonjour. We have developed a plugin for Firefox that searches for gateways on the local network and the user simply clicks on the name of the gateway he wants to be associated with, and the respective address gets loaded into the Web browser.

C. Representing Locations on the Web

Given that many different localization techniques are available for different applications, the representation of location information must be kept independent of technology to allow a sufficiently loose coupling, thereby allowing different localization systems to be used. Although many formats to represent outdoor locations have been developed recently, there is no standard way to represent indoor location information, and certainly none that complies with the existing Web technologies. As geographic coordinates are not practical for dealing with the concept of a location used in everyday life, as for example a room's number or a building wing's name, a flexible model that supports user-generated symbolic annotations of places is needed. Sharing semantics of places can be a tedious problem in case a central authority must maintain a repository of place names, which would be against the Web's decentralized nature.

To solve this problem, we propose to use the Web itself as a lookup service to find and explore locations as well as to obtain information about places and the devices therein. Following the idea formualted in [22], we use universal resource identifiers (URI) to represent locations and their containment relations as a logical path according to the URI definition. Consequently, RESTful URIs can be created dynamically by navigating the hierarchical tree formed by the gateways. For each URI, both machines and people should be able to retrieve a description about the identified resource. This is an essential mechanism for a establishing a shared understanding about the location identified by the URI, where machines can retrieve semantic data (RDFa) while people can retrieve a human readable representation (HTML).

In addition to using URIs, we propose a concept we call `here/*`. This concept is based on the idea that a fixed string, namely `here/`, is used to identify the current location of a user or device on the gateway tree. This is conceptually similar to a dynamic bookmark that always points to the URI of the gateway associated to that location. A relative `here/*` URI can be constructed dynamically to allow for search and exploration across physical locations:

```
URI = http://host/{location"/"}[keyword]
```

Thereby, the `host` denotes the (network) location of any Gateway (IP address or network name). To traverse the location structure, `{"/"location"/"` is used to represent a path of arbitrary length. Finally, by specifying a `keyword`, the user can search for devices and services that match the expression.

D. Dynamic Search of Mobile Devices (scoping)

Searching for devices that meet a certain criteria is an important feature of every modern distributed architecture. For our purpose, devices and gateways can be tagged with specific keywords and then be searched based on their associated keywords. As the search algorithm per se was not our focus, a simple search algorithm has been used. Search starts by matching a search expression with all devices and direct neighbors on a gateway (the gateways that can be reached within one hop). The search query also contains a time-to-live (TTL) that is decreased on each intermediate searching hop. If the TTL reaches zero the token will not be routed any further. The cost of this search algorithm grows exponentially according to the size of the network thus is not very scalable, but is sufficient for most pervasive and ad-hoc interaction scenarios where one is primarily interested to interact with devices located in the same room or floor.

By using the URI syntax defined in the previous section, the URI becomes a flexible search bar. For example, the wildcard character "`*`" can be appended to the URI of any gateway, which instructs the gateway to generate a webpage with all the links this particular gateway posses (devices or other gateways). For example, to find all devices tagged with the keyword `phone` located on the same floor, one can simply write the following URL in a browser:

```
http://here/floor/phone/*
```

This will be resolved by the gateway the user is associated with, and will get routed to the gateway of the corresponding floor. This is possible because the links between gateways are tagged semantically using RDFa. Subsequently, a HTML page with links to all the devices that match the query and are "under" the floor gateway at that time will be generated dynamically. Needless to say, the same URI will yield different answers depending on the floor the client is located on.

IV. Prototype Implementation: Mobile Ambient Meter

As proof of concept we illustrate how simple location-aware applications can be built by using our Web-based infrastructure. Our system is composed of a mobile energy meter that displays the energy consumed in the room it is located in.

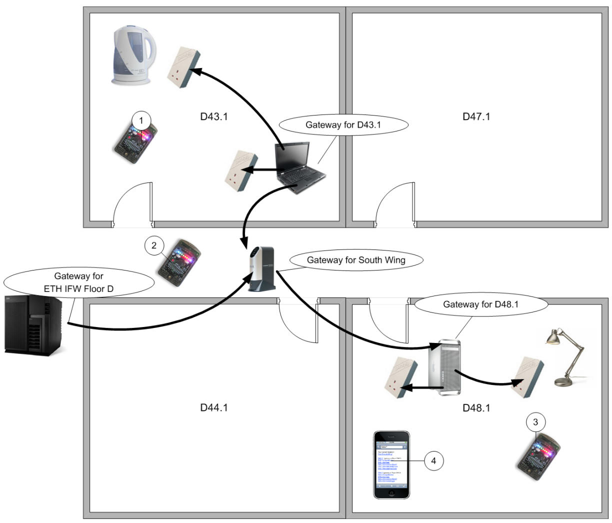

Figure 2. The prototype deployment on the floor of our building, with three physical gateways in two different rooms and one in the hallway. Two energy meters are connected to the corresponding gateways. As the Ambient Meter moves from one room to the other its color changes according to the level of energy consumption in the room (1-3).

As shown on Figure 2, two gateways are deployed in office D43.1 and D48.1, and communicate wirelessly with the Ambient Meter. The Ambient Meter is a SunSPOT sensor node[2] that runs an embedded Web server so that its functionalities are available as URI-identified resources. This enables the mobile user to access the sensors' resources simply by entering their URI in any Web browser. The energy consumption of electric appliances are monitored using Ploggs[3], which are sensor nodes that combine an electricity meter plug and a data logger that can be accessed with Bluetooth. Ploggs are also connected to the gateway so that they become URI-identified resources.

As shown on Figure 2, the meter is located in room D43.1 and is connected with the gateway of that room ① (we assume that only 1 gateway is within signal reach in each room). The device gets the energy consumption of the electric appliances in that room from the gateway that aggregates the individual consumption of all Ploggs in the room (white boxes next to the laptop and the kettle). Depending on the total amount of energy consumed, the Ambient Meter changes its color from green (little energy is consumed) to red (a lot of energy is consumed). The meter retrieves this information by periodically issuing a HTTP GET request for the `here/energyMonitor` resource. Since the gateway knows its location, it automatically resolves this URI to: `http://192.168.99.6:8081/energyMonitor`. The meter is then taken towards another room. On its way, the meter connects to the gateway of the left wing of the floor, so that it displays the energy consumed in the area ②, which is the aggregation of the individual consumption of each room. The Ambient Meter is then taken to room D48.1, where it reconnects automatically to the gateway of this other room ③. The consumption of a lamp and a desktop computer located in D48.1 is then displayed. Note that the ambient device does not actually deal with any explicit location since it only asserts the energy consumption its current location - which depends on the closest gateway.

In the last part of this scenario a user enters room D48.1 ④. Using a Firefox plugin, the user connects to the gateway of the room he is located in: D48.1. As shown on Figure 1 (right), he gets back a webpage containing links to either related locations (e.g. D48.2, etc.) or to resources in the current room. By clicking on the `http://D48.1/energyMonitor/` he retrieves the amount of energy consumed by the lamp and the desktop computer. As the Ambient Meter is also currently located in this room the user can click on its link and access its services. For instance this link can be used to retrive the temperature currently sensed by the Ambient Meter for room D48.2. This illustrates how users can leverage the gateways' structure and the concept of Web of Things to browse (and bookmark) their physical location as they would with other Web pages.

V. Conclusion

We have presented how the design principles of the modern Web architecture can be applied to build a location-aware infrastructure for physical objects. In our approach, devices are made available as resources on the Web by regardless of actual protocols used by the devices. This is made possible by using gateways that enable access to devices through a uniform interface based on same architectural style that is used in the World Wide Web. This allows physical objects to be searched, accessed, browsed, and linked together exactly like any other Web resource. A hierarchical structure of physical locations can be created by linking gateways, and this infrastructure can be used to search and use things based on their location and other contextual information. We have shown with a prototype implementation that simple and lightweight location-based services called physical mashups can be implemented with any mobile device that supports HTTP. Security and privacy have not been addressed in this paper, however we are investigation the usage of HTTPS and

[2] `http://www.sunspotworld.com/`
[3] `http://www.plogginternational.com`

OAuth to enable authenticated and secure communication between mobile clients and gateways. Future work will also include a more detailed performance and scalability analysis of the whole system.

References

[1] T. King, S. Kopf, T. Haenselmann, C. Lubberger, and W. Effelsberg, "Compass: A probabilistic indoor positioning system based on 802.11 and digital compasses," *Proceedings of the First ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and CHaracterization (WiNTECH)*, Aug 2006.

[2] P. Bolliger, "Redpin - adaptive, zero-configuration indoor localization through user collaboration," *Workshop on Mobile Entity Localization and Tracking in GPS-less Environment Computing and Communication Systems (MELT), San Francisco*, 2008.

[3] G. Abowd and E. MYNATT, "Charting past, present, and future research in ubiquitous computing," *ACM Transactions on Computer-Human Interaction*, Jan 2000.

[4] E. Wilde and M. Kofahl, "The locative web," in *Proceedings of the first international workshop on Location and the web*. Beijing, China: ACM, 2008, pp. 1–8.

[5] T. Kindberg, J. Barton, J. Morgan, G. Becker, D. Caswell, P. Debaty, G. Gopal, M. Frid, V. Krishnan, H. Morris, J. Schettino, B. Serra, and M. Spasojevic, "People, places, things: web presence for the real world," *Mob. Netw. Appl.*, vol. 7, no. 5, pp. 365–376, 2002.

[6] C. Prehofer, J. van Gurp, and C. di Flora, "Towards the web as a platform for ubiquitous applications in smart spaces," in *Second Workshop on Requirements and Solutions for Pervasive Software Infrastructures (RSPSI), at Ubicomp 2007*, 2007.

[7] A. Kansal, S. Nath, J. Liu, and F. Zhao, "SenseWeb: an infrastructure for shared sensing," *IEEE Multimedia*, vol. 14, no. 4, pp. 8–13, 2007.

[8] T. Luckenbach, P. Gober, S. Arbanowski, A. Kotsopoulos, and K. Kim, "TinyREST - a protocol for integrating sensor networks into the internet," *in Proc. of REALWSN*, 2005.

[9] P. Gibbons, B. Karp, Y. Ke, S. Nath, and S. Seshan, "Iris-Net: an architecture for a worldwide sensor Web," *Pervasive Computing, IEEE*, vol. 2, no. 4, pp. 22–33, 2003.

[10] O. Haque, "Pachube," Online at `http://www.pachube.com`.

[11] O. G. C. Inc., "OGC Sensor Web Enablement: Overview and High Level Architecture," White paper OGC 07-165, 2007.

[12] P. Bahl and V. Padmanabhan, "Radar: an in-building rf-based user location and tracking system," *INFOCOM, Tel Aviv, Israel*, Jan 2000.

[13] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, "The anatomy of a context-aware application," *Wireless Networks*, vol. 8, no. 2-3, pp. 187–197, Jan 2002.

[14] B. Schilit, N. Adams, R. Gold, M. Tso, and R. Want, "The parctab mobile computing system," *Workstation Operating Systems*, Jan 1993.

[15] E. Bhasker, S. Brown, and W. Griswold, "Employing user feedback for fast, accurate, low-maintenance geolocationing," *Pervasive Computing and Communications (PerCom)*, Jan 2004.

[16] G. Chen and D. Kotz, "A survey of context-aware mobile computing research," Dartmouth College, Hanover, NH, USA, Hanover, NH, USA, Tech. Rep., 2000.

[17] M. Weiser, "Some computer science issues in ubiquitous computing," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 3, no. 3, 1999.

[18] M. Bauer, C. Becker, and K. Rothermel, "Location models from the perspective of context-aware applications and mobile ad hoc networks," *Personal Ubiquitous Computing*, vol. 6, no. 5-6, pp. 322–328, Dec. 2002.

[19] M. Beigl, T. Zimmer, and C. Decker, "A location model for communicating and processing of context," *Personal Ubiquitous Computing*, vol. 6, no. 5-6, pp. 341–357, Dec. 2002.

[20] B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer, "Easyliving: Technologies for intelligent environments," in *Handheld and Ubiquitous Computing: Second International Symposium (HUC 2000)*, ser. LNCS, vol. 1927. Bristol, UK: Springer, September 2000, pp. 12–27.

[21] D. Garlan, D. P. Siewiorek, A. Smailagic, and P. Steenkiste, "Project Aura: Toward distraction-free pervasive computing," *Pervasive Computing*, vol. 21, no. 2, pp. 22–31, May 2002.

[22] C. Jiang and P. Steenkiste, "A hybrid location model with a computable location identifier for ubiquitous computing," *UbicOmp*, Aug. 2002.

[23] "An introduction to georss: A standards based approach for geo-enabling rss feeds," OGC 06-050r3, july 2006.

[24] A. Mayrhofer and C. Spanring, "A uniform resource identifier for geographic locations ('geo' uri)," Internet Draft draft-mayrhofer-geo-uri-02, Feb. 2008.

[25] C. Stahl and D. Heckmann, "Using semantic web technology for ubiquitous hybrid location modelling," in *1st Workshop on Ubiquitous GIS, in conjunction with 12th International Conference on Geoinformatics*, 2004.

[26] R. T. Fielding and R. N. Taylor, "Principled design of the modern web architecture," *ACM Trans. Internet Techn.*, vol. 2, no. 2, pp. 115–150, 2002.