Adaptive Wireless Services for Augmented Environments

Xing Liu
Technical University Berlin
10587 Berlin, Germany
Email: scripper@cs.tu-berlin.de

Tansu Alpcan
Deutsche Telekom Laboratories
10587 Berlin, Germany
Email: tansu.alpcan@telekom.de

Christian Bauckhage
University of Bonn and Fraunhofer IAIS
Sankt Augustin, Germany
Email: christian.bauckhage@iais.fraunhofer.de

Abstract—This paper presents a system that combines mobile services, ubiquitous computing, and augmented reality concepts in order to bring the kind of information-rich environment, which is currently limited to computer screens, to the physical world. This objective is achieved by constructing an architecture based on existing wireless Web application and sensor network technologies. Users of the system as well as objects in the environment are associated with individual sensors. The system collects proximity information between objects and users, which is then used by the Web application server to provide adaptive services. The users then access these location-based services over the wireless network using a standard Web browser on their mobile device, e.g. smart phone. Unlike existing GPS-based solutions, this approach allows for a proximity resolution of approximately one meter inside buildings and rooms. Thus, the augmented objects in the surrounding environment of the users become local hubs of interactive services, creating an information-rich environment that is similar to the one on the Internet. The system presented, which can be built using already existing technologies, has widespread potential applications in, for example, elderly care, education, and commerce.

I. INTRODUCTION

Recent developments in mobile computing and wireless networking technologies resulted in an explosion of mobile and location-based services. Early technology adopters are already surrounded by ubiquitous computing services, which are destined to become an integral part of the daily life of the general public in the next decade. Today, however, mobile applications are mostly static, platform-specific, and closed when compared to the services we all enjoy within the virtual environment of the Internet. While this can be attributed partly to hardware (e.g. battery, bandwidth) limitations, vendor lock-in, and form-factor issues, the main reason is arguably the lack of an open architecture similar to the one on the Internet. Fortunately, with recent developments such as rapidly advancing mobile computing devices (in the form of smart phones or netbooks), improvements in bandwidth with next generation wireless networks, and the arrival of the mobile Internet, this limited situation is expected to change.

This study presents a system, which combines mobile (location-based) services, ubiquitous computing, and augmented reality concepts. The goal is to bring the kind of information-rich environment, which is currently limited to computer screens, to the real world. This is achieved by constructing an architecture based on existing wireless Web applications and sensor network technologies.

The users of the system and objects in the environment are associated with individual sensors. The received signal strength indicator (RSSI) values of the sensors are used to compute proximity information between objects and users, which is then utilized by the Web application server to provide adaptive services over wireless Web. The users then access these location-based services over the wireless network using a standard Web browser on their mobile device such as a smart phone or Internet tablet. Unlike solutions based on global positioning (GPS), the adopted approach allows for a proximity resolution of approximately one meter inside buildings and rooms.

Individual objects in the environment thus become hubs of local, interactive, and adaptive services, creating an information-rich and interactive environment similar to the one on the Internet. Possible applications of the proposed system are numerous and far reaching. For example, such an environment can be used for elderly care, education, or commercial purposes. At the same time, the proposed system can be seen as a precursor to futuristic concepts such as the Internet of things or virtual reality. In this paper, however, the emphasis is on building the described augmented environment using today's existing technologies.

Recent developments in the area of wireless sensor networks make them easier to program [1] and more usable than ever before. At the same time numerous sensor network solutions and platforms are emerging [2]. Due to favorable properties such as their small size, low cost, and flexibility, sensor networks can be used to make the physical world more interactive [3]. For example, sensor networks can be used to monitor the health condition of patients in hospitals [4]. In other cases, wireless sensors are deployed as a signal broadcasting agent to address tracking problems or provide various services in either indoor or outdoor environments [5]–[7].

With the emergence of the Web 2.0, there is concurrently an explosion in dynamic Web platforms making development and deployment of Web-based applications much easier than before. For example, the Turbogears [8] Web programming framework used in our prototype is based on the Python [9] Web framework which facilitates sophisticated and flexible prototyping in very short amount of time without requiring high level of expertise from the developers.

The **contributions** of this study, which are presented in detail in the next sections, include:

1) an architecture that combines existing technologies to create a working augmented environment based on wireless Web services and sensor networks,
2) a fully functioning prototype as a source of usage data for further research and development,
3) development of adaptive services that are mobile, location-based, and interactive such as Twitter and Wiki.

Background

Modern *smart phones* have become powerful mobile computing devices. Despite their small sizes they are equipped with a multitude of sensors and network interfaces allowing them to communicate with other devices over different channels. In addition, communication technologies such as Bluetooth, Wi-Fi, Infrared, and UMTS facilitate a location-based exchange of personalized (multimedia) data when combined with GPS or RFID tags. At the same time, wearable, head-mounted displays have become available at affordable prices. The architecture presented in this paper is inspired from the convergence of these trends.

Mobile services are at the center of the envisioned system. Although first conceptualized about a decade ago, mobile services on smart phones were, for a long, time confined to primitive Web browsing (e.g. WAP). Recently, rich Web applications such as *Google maps* have extended the range of mobile services towards more advanced and location-based use cases. Another popular Web application, *Twitter*, bridges online communities of cell phone users and bloggers but is still missing a closer integration with the physical world.

The idea of *ubiquitous computing*, too, has been around for years. However, it appears that it still has not matured as a technology that is accessible by the general public. The omnipresence of cell phones and progress in the area of ad-hoc networking, on the other hand, suggest that nowadays the basic technologies are finally available to make ubiquitous computing a reality.

The paradigm of *augmented reality* (AR) is yet another concept that has been discussed for years [10] but has not yet become a technological commodity. The predominant paradigm, where the environment is passive and the AR interface has to actively analyze its current surroundings to produce an enriched view for the user, requires sophisticated signal processing algorithms for auditory and visual analysis of signals recorded by, say wearable microphones or cameras. However, corresponding algorithms that are flexible and robust enough to be deployed in everyday environments and can be used by ordinary people do not yet exist. On the other hand, if everyday environments and objects were to become active information sources themselves, existing technologies could easily process this information and display them to the user. Two dimensional QR codes, which have been successful in some communities can be seen as a primitive proof-of-concept application of this alternative approach.

Finally, *location based services* are among the hottest topics in mobile computing in recent years. Location information not only distinguishes mobile computing from the existing Internet, but also bridges virtual and physical environments. GPS-based traffic assistance devices, albeit being specific purpose, are precursors of a rich ecosystem of potential services and have been quickly adopted by the general public.
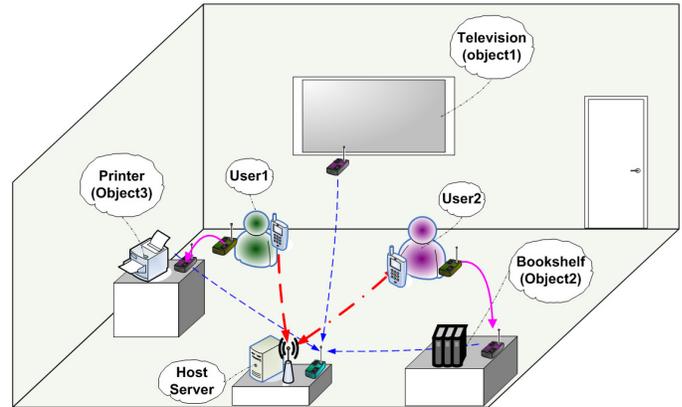
II. System Architecture

Fig. 1. A representation of the proposed augmented environment.

The system operation can be summarized as follows. Each user carries a mobile device such as a smart phone as well as a wireless sensor (Figure 1). In addition, each augmented object in the environment is associated with its own wireless sensor. The sensors determine which object is closest to the user at a given moment and pass this information to a server, which provides specific Web services such as Wiki and Twitter for each individual object. These object- or location-based services may also be adapted to the individual user profile. The users then access the individual Web services of the closest object over a wireless network using a standard Web browser on their smart phone.

The system architecture consists of two main parts that operate in parallel and complement each other. The first one provides proximity information on the object level, i.e. with a granularity of approximately one meter, in closed environments such as building floors and rooms. Such a proximity measurement between users and individual objects is necessary to realize the envisioned augmented environment. Clearly, this cannot be achieved through GPS or Wi-Fi triangulations as their resolution is too low (on the order of meters) to be effective on the object level. Alternative methods for obtaining the desired proximity data include barcodes (classical 1D or 2D such as QR), RFID tags, and wireless sensors.

The architecture presented in this paper applies wireless sensors to obtain proximity information mainly due to their off-the-shelf availability, ease-of-use and well-documented programming tools. On the other hand, it is not argued here that these sensors always provide a better solution than other

means. One can envision applications and settings where any of the above alternatives is better suited than others.

The second part provides the adaptive mobile (wireless) services deployed in the form of rich, location-based and interactive applications. Taking into account potential compatibility issues at the client-side and the burden of supporting multiple mobile operating systems and versions, a Web-based approach is adopted instead of a client-server solution. This decision is supported by the fact that recent browsers and AJAX Web technologies already enable seamless and interactive mobile applications. The wireless communication between the Web application server can be established using a 802.11 Wi-Fi network as well as other alternatives such as 2G and 3G wireless solutions.

Combining these two parts, the current implementation consists of a hybrid networking architecture combining a wireless sensor network and a Web application server operating over 802.15 and 802.11 wireless networks, respectively. Individual wireless sensors are attached both to physical augmented objects as **relays** and carried by users as **beacons** to obtain the proximity between users and objects (see Figure 1). Specifically, the proximity of a user to a certain object is computed by measuring the received RSSI values of the user beacon broadcast at the relays attached to objects. Then, these values are compared with each other at the server (Host) to determine which object is nearest to the user.

Once the proximity information of users and objects is obtained, a variety of services can be implemented in the form of wireless Web applications. The objects are given both a presence in the virtual environment (e.g. Internet) and associated with mobile services. Hence, the system creates a rich interactive *augmented environment* that is similar to the one on the Internet. In this setting, the location or proximity information provided by the sensor network bridges the physical world with the virtual one. At the same time, it organizes data and services in a very natural way, namely by location, and thus decreases the existing and potential information overload associated with virtual environments.

Putting immediate feasibility before other considerations, the architecture proposed is a flexible, user-friendly, and cost-efficient way for realizing an augmented environment. The system consists of off-the-shelf components such as a wireless sensor network of TelosB motes using TinyOS [1], a 802.11 Wi-Fi access point, an open source Web application server (TurboGears), and smart phones as end user devices. The system allows users and developers to continue to use the tools they are familiar with. In addition, the modular and standard-based approach brings a lot of flexibility. The services are provided to the users as Web applications, which are supported by a wide range of end user devices through mobile browsers. Each of the three main components (mobile devices, application server, and sensor motes) providing proximity information can be replaced by alternatives. For example, on a scale beyond object-level, GPS can be used to obtain proximity information rather than the sensor network. If desired, virtual reality goggles may be used instead of phone screens as user displays. Finally, the application server, too, can be easily replaced by another one.

The developed system collects a variety of location and usage information which it stores in a database. This data is post-processed in order to adapt the services to individual user needs and provide adaptive applications using machine learning and optimization methods. Privacy of individual users is preserved as multiple users can use the same sensor mote at different times and may have multiple virtual identities (e.g. login names) which might not be linked to real world identities.

In the following sections, the sensor network deployment, the Web application implementation, as well as their integration are presented in detail.

III. WIRELESS SENSOR NETWORK

A standard wireless sensor network is deployed in order to obtain proximity information between users and augmented objects. Specifically, TelosB [11] wireless sensor motes are chosen as part of the architecture. Each TelosB mote contains an 8 MHz Texas Instruments MSP430 microcontroller and a IEEE 802.15.4 compliant RF transceiver (CC2420) with a globally compatible ISM band between 2.4 and 2.4835 GHz capable of 250 kbps data transmission rate. In addition, it has 10KB RAM and 1MB external flash-based storage capacity for data logging. Compared to other wireless sensors, TelosB distinguishes itself through low-power consumption, hardware write protection, and flexible and robust configuration of the programming module. One of its most attractive properties for our system is its use of the IEEE 802.15.4 standard, which facilitates wireless communication in a more energy-efficient way.

Following the analysis in [12], the received signal strength indicator (RSSI), which characterizes link quality, is used here as an inexpensive and feasible method to estimate the relative distance between the sensor nodes. Thus, assuming that the link quality between sensor nodes closely correlates with their distance, proximity information between sensor nodes (objects and users) is obtained with a resolution of approximately one meter in indoor environments. Next, we describe the details of the wireless networking structure in our experiments as well as the deployment steps.

A. Network Structure

The deployed sensors (motes) on the wireless sensor network shown in Fig. 2 can be divided into three categories of distinct functionalities. They are respectively:

a) Beacon: A Beacon mote is carried by the mobile user and periodically transmits a signal for the system to determine the relative location of the user. Using the IEEE 802.15.4 wireless channel, it broadcasts a well-defined radio packet periodically, whose message stack includes a unique mote ID with which each mobile user could be accurately identified.

During experiments, it is empirically observed that, due to radio irregularity, not all the broadcasted packets from the beacon successfully reach the target. This situation results
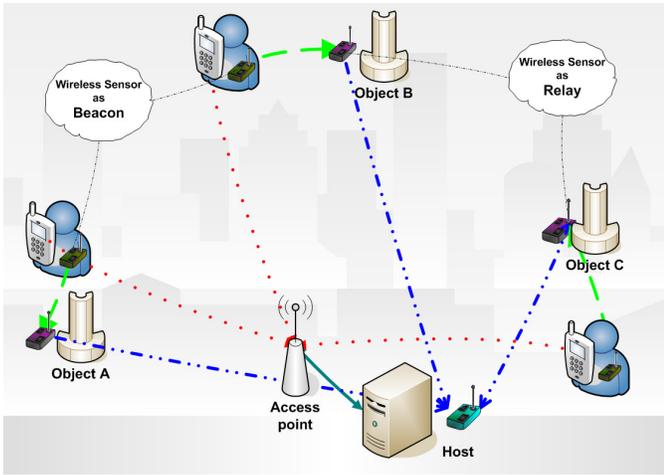
Fig. 2. The wireless sensor network as part of the augmented environment.

in consecutive packet loss and deteriorates the proximity detection performance. In order to decrease this potential transmission failure, the RF sending power of a Beacon is set to the maximal level as a temporary solution.

b) Relay: Relay motes differ from Beacons by their message stack and functionality. Each Relay mote is associated with a specific object in the environment. As shown in Figure 3, the message stack of Relay motes in our experimental environment consists of 4 bytes in total with three block items, indicating Beacon ID (1 byte), Relay ID (1 byte), as well as the RSSI value (2 bytes). In general, a Relay measures the RSSI value of each incoming radio packet and also receives the Beacon ID in order to identify the source of the radio packet. Together with its own Relay ID, the received radio packet is forwarded to the remote Host.
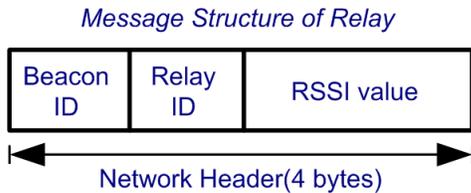

Fig. 3. The message stack of Relay motes.

c) Host: All radio messages sent by a Relay are received by the Host and eventually forwarded to the host server for post-processing. In particular, the Host mote not only collects radio messages from the Relays but also acts as a gateway or bridge between the application server and sensor network streaming the collected RSSI values. In the current implementation, after the host mote receives each radio packet sent by a Relay, the RSSI information is posted immediately to the connected server via a USB interface. The specific program handling this task is implemented in Java as part of the existing TinyOS tools.

The Beacon, Relay, and Host motes described above are the

basic building blocks of the sensor network. However, in order to meet essential requirements such as good tracking accuracy or low power consumption, the motes have to be programmed appropriately to achieve a robust system performance. During our experiments, we have identified several issues which crucially impact the system's performance and need to be addressed in detail. The first of these is the timer setting in Beacon motes.

B. Beacon Timer

The timer is a component provided by TinyOS that allows for programming of synchronous events. For a Beacon in particular, it is necessary that the radio packets are broadcast to nearby Relays periodically.

Using the trigger provided by the timer, the Beacon broadcasts radio packets periodically, which are used to infer proximity information of users. The timer interval is an important parameter for the successful operation of the system. On the one hand, setting a long timer interval in a Beacon may lead to tracking inaccuracy when users are moving quickly between objects. On the other hand, choosing a very short period is again not an optimal choice, since the tracking accuracy also depends on the Relay behavior. Furthermore, excessive transmission produces redundant RSSI values and increases energy consumption of the Beacon. Therefore, the timer should be set such that power consumption is low but proximity information of moving users can be determined fast and accurately.

C. Power consumption of Relays

Having discussed the power consumption of Beacons and since the Host, which is attached to the server, is electrically powered, the main issue in the radio power management of motes is caused by the Relays attached to objects. Generally, a Relay mote is regarded as a "message carrier", receiving incoming radio packets and forwarding them to the next destination immediately. In our experimental setup, the Relay motes run on batteries and receive RSSI values from user Beacons at the object they are attached to. This requires a Relay to keep its radio continuously alive in order not to miss broadcast packets from Beacon motes. However, in this way, the batteries of motes will deplete rapidly thus leading to forwarding failure. Having an electrical power supply for each Relay is not always possible or practical. Hence, we deploy the low-power listening technique provided by TinyOS which allows for a low power radio operation during listening. In low-power listening mode, a mote keeps its radio alive long enough to detect a carrier on the wireless channel. If it detects a carrier, it will stay on to detect further packets. Should it then receives a packet, the mote will stay awake long enough to receive a second packet and received packets will be directly forwarded to the next node instead of sending individual packets at a fixed constant rate.

By applying the Low-Power Listening technique, the mote can be put into sleeping mode in order to save energy. However, depending on the specific system requirements, the

sleeping interval has to be set properly such that a good balance between tracking accuracy and power consumption is achieved.

Since all Relay motes in the system receive, process and finally forward each Beacon's packet to the Host, a growing number of users in the environment leads to higher packet traffic between Beacon and Relay, which will then consume more energy. A possibility for the Relay is to send multiple Beacon messages to the Host bundled in a batch to save energy and decrease traffic. However, this also introduces a certain amount of delay for proximity detection. In summary, the number of users, the timer period of Beacon together with sleeping interval of Relay constitute an optimal control problem within the sensor network, where the objectives of tracking accuracy of users and power consumption of sensors have to be balanced.

IV. MOBILE WEB SERVICES

Compared with the sensor network architecture discussed, the wireless (Wi-Fi) network over which the Web services are provided is easier to install, program, and configure. It consists of a server working in conjunction with the Host of the sensor network and smart phones as clients. Since the services are Web applications, the clients are merely plug-and-play devices without any need for configuration. The application side of the framework is then implemented as a database-driven Web application running on the state of the art Python based Web framework Turbogears [8]. On the server side, user proximity information as well as activity on certain services are recorded, time stamped and stored in the database and then used for post-processing. On the client side, each mobile user receives periodically refreshed contents and adaptive services in the form of interactive AJAX-based Web applications, which are dynamically generated based on his location and data profile. In the following subsections, we present the particular structure of our experimental system as well as the techniques applied in its development.

A. Location-Based and Adaptive Applications

The wireless network provides the (research-oriented) technical platform, on which mobile Web applications are designed and deployed. Currently, the applications include a couple of social-networking services, implemented as simple Twitter and Wiki clones, as well as other dynamic contents such as targeted advertisements. These services are designed to be location-based and adaptively rendered to the end-user over the wireless network.

The services currently deployed on the system are summarized in Table I. Twitter has been characterized as the next generation social-network platform and has constantly gained popularity over the last few years. It is a micro-blogging platform that enables users to broadcast short messages (tweets) and read other users' updates. A Wiki is a collaborative working platform and knowledge sharing management system, where anyone can access, contribute or modify the content. Both these applications have an important place in modern

TABLE I
SERVICES ON OBJECTS

Service	Description
Info	Basic info about the object, e.g. a manual.
Wiki	Collaborative knowledge sharing on the object.
Twitter	Object's own public twitter portal for information dissemination and communication.
Advertisement	Targeted and personalized advertisements.

social networking and enable new ways of interaction between people. Our goal is to explore and adapt these two popular social networking services to our system such that they are location-based. Hence, users may interact and communicate with each other via augmented objects. Furthermore, we propose to use proximity information and data histories of users recorded by the Web-based applications as a mean of filtering out redundant and unfavored information or services, hence improving the quality of interaction in the augmented environment.

Recall the system structure shown in Figure 1: relay motes are attached to certain objects and users' relative proximity to individual objects is computed using received RSSI values from Beacon broadcasts in near real-time at the Host. Whenever a user (carrying a Beacon mote) approaches an object, the Beacon mote ID as well as the received signal strength is forwarded to the Host by the Relays. The Web application server, in turn, provides an object-based service and user-specific content to users smart phone over the Wi-Fi network.

The use case shown in Figure 4 demonstrates user activities in an illustrative example scenario. First, the mobile users register an account through the Web interface on their mobile phone. During login, they provide a username/password as well as the ID of the Beacon mote they are carrying. Subsequently, they are redirected to a Web page, where user-specific contents are displayed. At the same time, object-based services are provided, as long as the user location can be determined through RSSI information received by the Host and server. Once the user moves from the vicinity of an object A towards another object B, the corresponding object-based services of object B will be updated and displayed on the mobile phone without explicit user intervention utilizing AJAX technology. Therefore, if several objects in an environment are augmented (with Relays), a user can obtain and use various services which are specific to the nearest object.

Providing dynamic applications with respect to spatial location of the user is one aspect of adaptive services. Another notion of service adaptability is based on the activity of individual users over time. Consider again the above scenario. If user interactions with augmented objects and their respective services are recorded in the system database, behavior and usage patterns and preferences can be processed and analyzed by the system to create a series of adaptive and dynamically
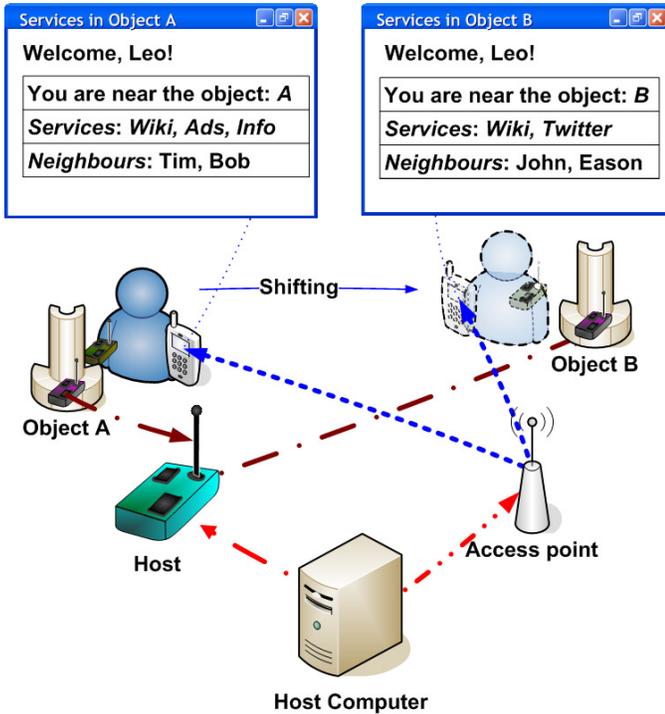
Services in Object A `_ ⬜ ✕`

Welcome, Leo!

| You are near the object: *A* |
| Services: *Wiki, Ads, Info* |
| Neighbours: Tim, Bob |

Services in Object B `_ ⬜ ✕`

Welcome, Leo!

| You are near the object: *B* |
| Services: *Wiki, Twitter* |
| Neighbours: John, Eason |

Shifting

Object B

Object A

Access point

Host

Host Computer

Fig. 4. An illustration of system usage.

generated services in the augmented environment. Note that the successful and widely used Adsense service of Google operates on similar principles.

B. Programming Environment

In order to effectively minimize the programming effort for our testbed applications, we employ Turbogears as a state-of-the-art Web programming framework. Turbogears [8] is considered as one of the most rewarding Python [9] Web frameworks, allowing for sophisticated and flexible prototyping in a very short amount of time. During the development of our system, we have observed Python to be a very graceful language with numerous powerful features, which not only enhance the code productivity but also assure its quality as well. Between the Host mote running TinyOS and the Python-based TurboGears Web application server, a small Java program is deployed, in order to send the obtained RSSI values to the host computer over HTTP and thus to bridge the data stream between the Host mote and the Web application server.

C. Privacy and Security

Privacy and security play an important role in determining the success of augmented environments. At this initial stage of experimentation, we apply two principles to ensure the privacy of the subjects taking part in our experiments. Firstly, the digital identity of users is independent of their real identities. Hence, users can decide on their own privacy level in the system and maintain anonymity by having multiple digital identities. Secondly, each user can choose a different Beacon mote from a common pool for each separate session by

temporarily associating the mote with the respective digital identity during login. Hence, tracking of a certain mote does not yield information about a certain user.

We rely on well-established Web application security methods for the overall security of the system. Although the computing power of the sensor network (or in general the location information system) is limited, modern smart phones and Web application servers have enough computational power to run state-of-the-art encryption schemes. Since all envisioned services in the augmented environment are Web-based, it is possible to reuse existing security mechanisms of Web applications without any modification.

D. Deployment

After the initial design of both the sensor and the Wi-Fi 802.11 network, we have deployed the entire experimental system in the cafeteria of our lab in order to evaluate its operation and usability. The cafeteria was chosen for it provides a space where people are socializing, exchanging their ideas, and sharing their technical experience. It can be therefore regarded as an ideal interactive setting, where we can experiment with and demonstrate our augmented environment.

There are several fixed objects in the cafeteria such as, for instance, the coffee machine, the game console, the foosball table, and a small book shelf. These are natural candidates for augmentation and deployment of the sensor network. Each of the selected augmented objects is assigned a Relay mote running on batteries, which receives and forwards proximity information of users who carry a Beacon mote. As long as users move within the accessible radio range of the augmented objects, the broadcast radio packets from their Beacon are received by the deployed Relay motes. Through experiments, we have found that RSSI values suffice to obtain reliable proximity information between an individual object and a user, even if there is background noise (other radio signals) and channel attenuation. The anecdotal evidence shows that the system's proximity resolution is around one meter inside a room.

The specific algorithm used in current prototype to determine proximity of users to the objects is as follows. The RSSI values obtained from the broadcast of a specific user Beacon are received by multiple Relays at the objects. The median of the last three received RSSI values at each object is chosen as the representative current RSSI value of the user. This simple windowing scheme ensures that sudden fluctuations due to fading effects are filtered out and acts as a low-pass filter. Finally, the maximum of these RSSI values determines the object at which the signal is strongest, i.e. the user is nearest.

As an example, Figure 5 depicts the RSSI values of a user as received by two separate objects in the environment. At *location1*, the user is closer to *object1*. Then, the user moves to the *location2*, which is between the objects. Finally, in *location3*, the user is closer to *object2*. Since the Relays forward the received Beacon messages to the Host non-periodically and due to variability of wireless channel, the number of RSSI values received from a user by one object

may differ from the ones by another in a given time period. Therefore, the actual time instances of user movements are marked with vertical lines in the figures.
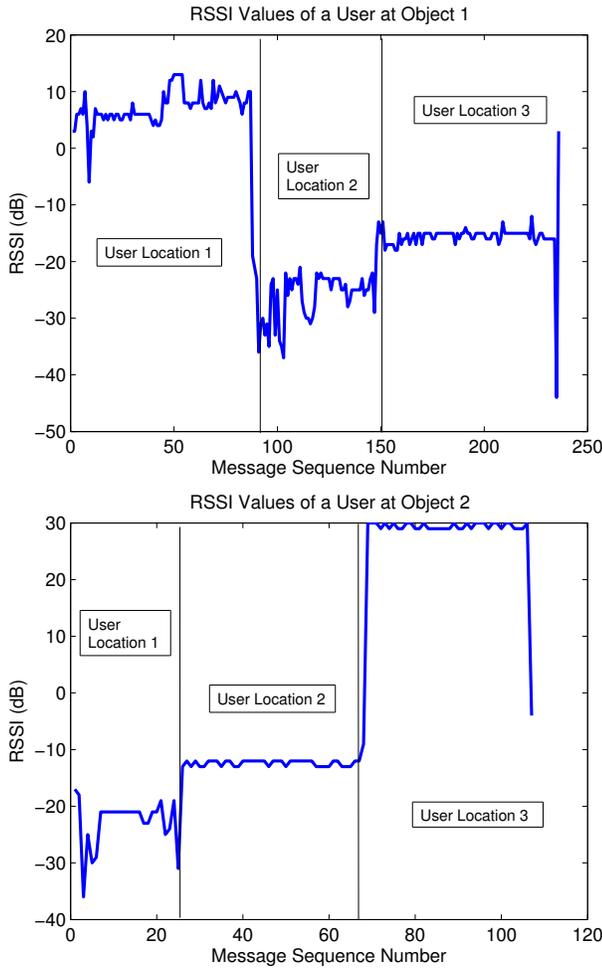

Fig. 5. RSSI values of a user as received by object1 (top) and object2 (bottom) for three different user locations. The number of messages (values) received from a user by each object may not be equal as illustrated in this case.

The implemented Web-based mobile applications are deployed on the host computer which is also located in the cafeteria. This computer is connected to a Wi-Fi Router so that the users can access the application through any Wi-Fi enabled mobile phone without any additional cost. A wireless sensor mote, the Host, is attached to the same computer via a USB interface. It gathers radio packets sent by the Relay motes, which contain RSSI values and user IDs. All of the incoming RSSI values are then parsed by this gateway mote and then posted to the Web application. For programming convenience the final proximity decisions are made by the Web application. A screenshot of the Twitter application from the actual system is shown in Figure 6.

V. DATA COLLECTION

Collection of usage data is crucial for generating user behavior models as a basis for adaptive services. The data
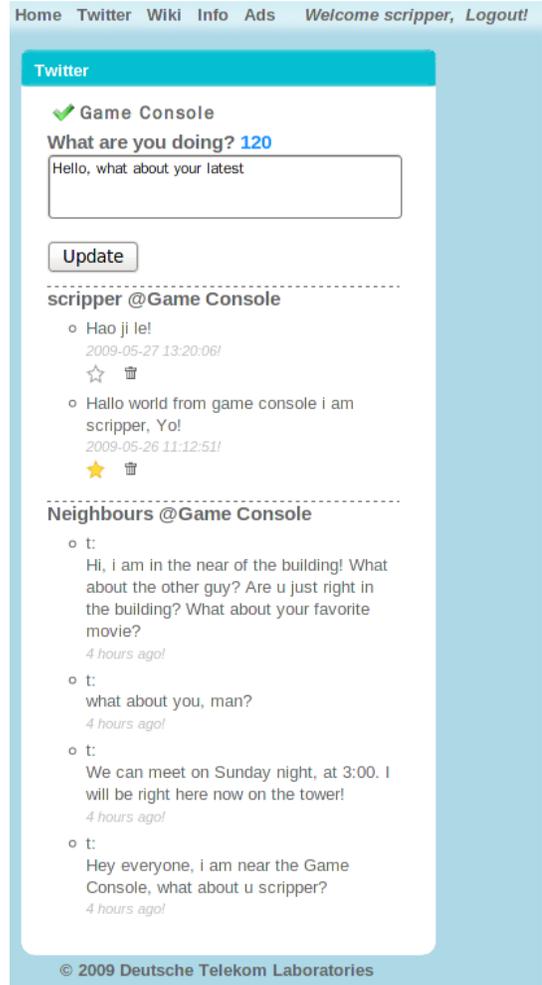

Fig. 6. A screenshot of the Twitter application as an example location-based messaging/micro-blogging service.

collected consists of two parts. One of them is the raw RSSI or signal strength values recorded by the wireless sensors. The other one is the data from the Web applications.

The RSSI values received from all objects are stored to a specific table in the database in order to analyze and evaluate proximity detection mechanisms deployed. Another table records time stamped actions of the users which include current closest object as well as service-specific actions such as editing a specific object Wiki, posting a Twitter message, etc. Hence, time stamped data sequences containing the relative path of the mobile user and the used services are periodically stored to the database by the server. By processing this dataset on the server side using various machine learning methods, our system can filter out unfavored services and has the potential of providing more flexible and intelligent applications.

Combining the two types of networks (sensor and Wi-Fi network) and making them work collaboratively is critical for the success of data collection efforts. During authentication, users receive a UserID which helps identifying them for data collection purposes. For simplicity, each user is assumed to

have a single UserID. In addition, each Beacon mote carried by a user during a session is allocated a unique MoteID. These two identifiers facilitate post-processing of datasets collected during experiments.

Finally, the server-side can continuously evaluate the collected data from spatially distributed sources (objects) to extract and observe usage patterns. With such patterns available, a series of user-specified options and services can be designed such as data set management, multi modal interaction, and anomaly detection.

VI. CONCLUSION AND FUTURE DIRECTIONS

In this paper, we have proposed a virtual augmented environment and implemented a prototype as an illustrative example. We have described a hybrid wireless network architecture which aims at providing adaptive, location-based Web services for the mobile users of the system. Our framework relies on RSSI (Received Signal Strength Indicator) values measured by wireless sensors motes in order to determine proximity between users and augmented objects in the environment. Experimental results have shown that this approach performs well in practice with a resolution of approximately one meter inside buildings, even if considerable amount of background noise disturbs data transmission. From the point of view of system performance, we have discussed the trade-off between power consumption and tracking accuracy. This leads to an optimal control problem formulation which will be derived and analyzed in future work.

By incorporating the wireless sensors together with a Wi-Fi 802.11 Network, a functioning prototype of the system has been designed and implemented. The services have been realized as Web-based applications employing the rapid prototyping Web framework Turbogears. Specifically, we have designed and developed several social-network like Web applications to provide adaptive wireless services, which are based on multiplexed information such as the time stamp of activities, locations and user-profiles.

Next, we have deployed the entire co-existing network system of both wireless sensors and Wi-Fi in the cafeteria of our laboratory (T-Labs). Through a Wi-Fi enabled mobile phone and a Beacon mote, a user can directly access the Web-based application and use various location-based services. A series of experiments are planned to collect a dataset for further examination. The data collected from spatial distributed sources will be evaluated on the server-side. Subsequently, a series of user-specific and location-based services will be adaptively generated in the environment, all based on user-dataset and his location.

As additional future research directions, we intend to exploit the utility of wireless sensors as well as their impact on the augmented environment. Formulation of an optimal control problem regarding the trade-off between power consumption and tracking accuracy in wireless sensor network and its further study will analytically improve the system performance. On the other hand, the data model of user activities collected during experimentation need further exploration. Using statistical machine-learning methods, we intend to uncover user patterns and common behaviors in the application in order to achieve more adaptive and personalized wireless location-based services.

REFERENCES

[1] P. A. Levis, "TinyOS: An open operating system for wireless sensor networks," in *Proc. of 7th Int. Conf. on Mobile Data Management.* IEEE, 2006.
[2] J. Polastre, "Sentilla pervasive computing kit (PERK)," May 2008, http://blog.sentilla.com/2008/05/.
[3] CrossBow, "Extending our senses into the physical world," August 2008, http://blog.xbow.com/xblog/2008/08/.
[4] V. Shnayder, B.-R. Chen, K. Lorincz, T. R. F. Fulford-Jones, and M. Welsh, "Sensor networks for medical care," in *Proc. of 3rd Int. Conf. on Embedded Networked Sensor Systems (SenSys'05).* ACM, 2005.
[5] E.-E.-L. Lau and W.-Y. Chung, "Enhanced RSSI-based real-time user location tracking system for indoor and outdoor environments," in *Proc. of 2007 Int. Conf. on Convergence Information Technology (ICCIT'07).* IEEE, 2007.
[6] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli, "Wireless Sensor Networks for Environmental Monitoring: The SensorScope Experience," in *20th IEEE Int. Zurich Seminar on Communications (IZS 2008)*, 2008.
[7] M. Gauger, P. Marron, M. Handte, O. Saukh, D. Minder, A. Lachenmann, and K. Rothermel, "Integrating sensor networks in pervasive computing environments using symbolic coordinates," in *Proc. of 3rd Int. Conf. on Communication System software and Middleware (COMSWARE 2008).* Bangalore, India: IEEE, January 2008.
[8] M. Ramm, K. Dangoor, and G. Sayfan, *Rapid Web applications with TurboGears: using Python to create Ajax-powered sites.* Upper Saddle River, NJ: Prentice Hall Press, 2006.
[9] G. V. Rossum, "An introduction to python for UNIX/C programmers," in *Proc. of NLUUG (Dutch UNIX users group) najaarsconferentie.*, 1993.
[10] R. Hainich, *The End of Hardware: A Novel Approach to Augmented Reality*, 2nd ed. BookSurge Publishing, 2006.
[11] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *Proc. of 4th Int. Symp. on Information processing in sensor networks (IPSN '05).* IEEE, 2005.
[12] K. Srinivasan and P. Levis, "RSSI is under appreciated," in *Proc. of 3rd Workshop on Embedded Networked Sensors (EmNets)*, 2006.