HBCI: Human-Building-Computer Interaction

Jeff Hsu, Prashanth Mohan, Xiaofan Jiang, Jorge Ortiz,
Sushant Shankar, Stephen Dawson-Haggerty, and David Culler
Computer Science Division
University of California, Berkeley
{jeffhsu,prmohan,fxjiang,jortiz,sushant,stevedh,culler}@berkeley.edu

Abstract

Buildings account for a large portion of the world's total delivered energy consumption. With smartphones becoming increasingly ubiquitous and sensor networks growing more mature, buildings can provide personalized and context-aware services to an occupant while minimizing energy consumption. This paper proposes the architecture of a Human-Building-Computer Interaction system that connects the building to its occupants by bridging the gap between the digital and physical worlds. We present our instantiation of an HBCI system, which is composed of an Android mobile application, a number of RESTful services in the cloud, and physical objects co-located with QR tags. We show that, with this system, a user can increase personal comfort within a building while reducing energy usage.

Categories and Subject Descriptors

H.4.3 [**Information Systems Applications**]: Communications Applications

General Terms

Standardization, Design

Keywords

Building Monitoring, Energy, Sensor Networks

1 Introduction

Buildings account for one fifth of the world's total delivered energy consumption [7]. A crucial factor of this statistic is the general lack of awareness occupants have of their energy consumption and their inability to optimize it. To meaningfully reduce energy, a building must actively involve the occupants of a building by providing easily accessible services and individualized feedback. Today, with the popularity of the mobile phone and the proliferation of cheap and ubiquitous sensors, we can finally connect the building to its occupants and optimize for both occupant comfort and energy efficiency (Figure 1). In our vision, users are able to meaningfully and easily interact with their physical environment as a set of objects via a network of sensors and actuators, effectively bridging Human to Buildings, Buildings to Computers, and Humans to Computers, thus completing the H-B-C interaction triangle: Human-Building-Computer Interaction (HBCI). With energy as an economical incentive, HBCI can serve as the core fabric to connect occupants to the building by bridging the gap between the digital and physical worlds.

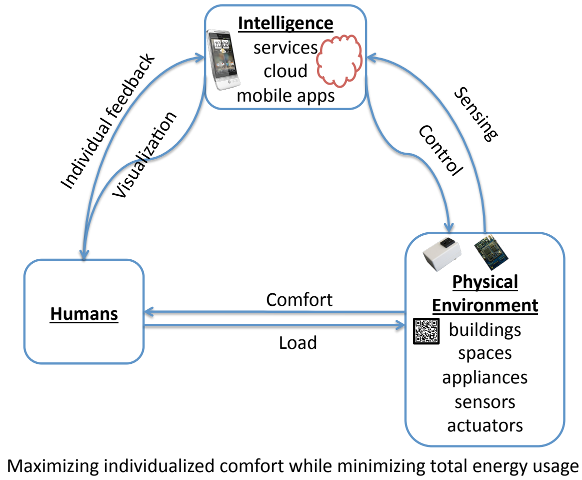


Figure 1. HBCI strives to optimize for both occupant comfort and energy efficiency.

In this paper, we build upon many years of previous research on ambient intelligent buildings while acknowledging the importance of energy efficiency. We propose the architecture for an HBCI system that allows a user in a building the ability to visualize data from numerous sensor streams, remotely and actively configure personal devices, and lastly, access context aware and personalized services provided by any object. Our implementation of the HBCI architecture strives to meet two goals at once: maximize a user's comfort and minimize total energy usage via remote monitoring and control of devices.

2 Related Work

Ambient intelligent buildings, including smart homes and office spaces, have been extensively studied in the literature. The Adaptive House [19], ComHOME [15], House_n [18] and the Aware Home project [16] are all examples of real working prototype labs and homes that have been built along the lines of smart spaces. All of these projects make extensive use of sensors in their setup to perform occupancy detection and also to infer user activities. Instead of specifically focusing on a single problem, the HBCI architecture presented in this paper allows third party developers to visualize the building as a central, physical deployment application framework. The architecture is extensible and flexible for various use cases. The prototype implementation involves an application of energy conservation among the occupants of a campus building. The HBCI architecture provides the necessary spatial, temporal, communication and security primitives that are required to enable development of such applications. The physical objects in a building are modeled hierarchically similar to the approach taken in [22]. An *HBCI Object* can have any number of child HBCI objects and any number of parent HBCI objects.

The use of mobile phones as the interface medium for smart environments has been found to be extremely beneficial in previous literature [11, 12]. The mobile smartphone has multiple sensors including an accelerometer, gyroscope, microphone, radios, etc. Since the phone is a personal device of the user it can be used to infer his/her activity and contextual setting. Inferences can be coupled with various services.

Our implementation of the HBCI architecture utilizes RESTful services, as they have been encouraged to be used for monitoring energy usage in Smart Buildings by [24, 21]. [24] proves that low power consumption of sensors can be achieved with REST. While [21] uses a RESTful web application to monitor sensor streams, in our implementation, we additionally provide a mobile application for energy footprint monitoring.

This paper does not address security, metering, or scheduling, as explored in [5, 23, 3, 17, 4]. Instead, we explore the design of an HBCI architecture that provides a set of primitives to enable participatory applications that reduce energy consumption in buildings. Analysis of security issues in mobile phone based sensing is outside the scope of this paper and depends upon existing mechanisms to provide a trustworthy platform.

3 Architecture

The HBCI architecture (shown in Figure 1) manages the interaction between the physical entities (HBCI Objects) and online web services (HBCI Services) offered. This is enabled by using localization of the physical entity and providing drivers for interfacing with various sensors and actuators already available in the building. It should be noted that the localization of objects in the building does not require absolute localization (GPS coordinates) but instead depends on *semantic* localization (e.g. Room 415 in Floor 4). The architecture is also extensible allowing any authorized application developer to associate new services to any object.

3.1 HBCI Objects

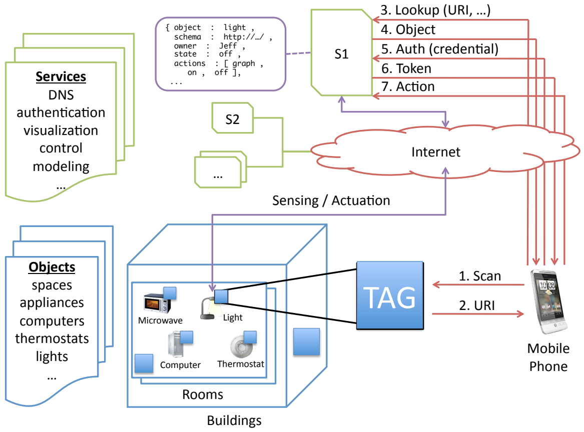
The concept of *Objects* is central to our HBCI architecture. An object is any physical entity that has a bridge into the digital domain. Objects can refer to spaces, sensors, appliances, lighting, computers, etc. In the current prototype, the bridge is enabled by placing tags on these entities. The tags can be scanned by mobile phones and resolved in the cloud to provide additional information about the object. To become an HBCI object, a physical entity is required to have a corresponding identifier in the HBCI database. Anybody can register objects in the system and attach identifiers to an entity. The ability to incrementally add objects and their corresponding metadata to the database reduces the bootstrap overhead of the system.

3.2 HBCI Services

Each object offers multiple *HBCI Services*. The URI's for these services are obtained by querying the HBCI directory with individual object identifiers. The services in turn offer multiple actions that individuals can perform on the object. Each object has at least one service attached to it, i.e. the "Query" service which returns a list of all other services attached to the object. Typically, the URI for the "Query" service is embedded in the reference tag, otherwise the application will fall back on a set of default query service providers. The identifier for the "HBCI object" is required to be unique within the namespace of an individual "Query" service. Thus, if the URI to the query service is not available, the user has to manually resolve the right namespace to use.

Examples of services associated with HBCI objects include the ability to stream sensor information, actuation of electrical outlets, etc. For instance, the HBCI object of a physical room could have the following services – "Room Temperature", "Lighting", "Occupancy detection", etc. Examples of actions provided by the "Room Temperature" service will include "Visualization of room temperature history" and "Room temperature control."

Developers can attach services to individual objects by adding a service to the list of URIs returned by the query service. The object owner determines the restriction of an object's services. While an unrestricted ability to add services to objects will result in the introduction of "Service Spam", existing spam filtering mechanisms and user ratings can be used to reduce this.



```
{ object  :  light ,
  schema  :  http://…/ ,
  owner  :  Jeff ,
  state  :  off ,
  actions  : [ graph ,
     on ,  off ],
 ...
```

Figure 2. Process flow of the HBCI Architecture.

3.2.1 *HBCI Drivers*

Drivers are a variety of HBCI Services. Drivers provide the functionality to interface with individual sensors and actuators. The driver should typically be written by the owner of the sensor and will expose certain functionalities that can be accessed by other services.

3.3 HBCI Process Flow

Figure 2 shows the individual steps performed by the application in order to interface with the HBCI architecture. (1, 2) The phone finds the identifier code for the object - either by scanning the tag or by using suitable localization, etc. (3) If the query service URI is present on the identifier, then the service is contacted. (4) The query service returns a set of service properties, actions, and metadata. The metadata contains the list of URI's to other services available for the object. Services may require authentication to execute certain actions. Authentication is performed by an authentication service associated with the object. (5) The "user authentication" service uses the object identifier and user credentials for verification and (6) returns a signed blob (including the object namespace and identifier), a token string and a token expiration time. (7) This token can be re-used for subsequent interaction with the service. End-to-end encryption such as SSL is used in order to prevent man-in-the-middle attacks.

3.4 Security

While the ability to view the history of room temperature information might not be considered sensitive, the ability to change the set point of the air conditioner could potentially be abused. Access to the air conditioning actuation service should either be restricted to the building manager or the service could provide a "poll" mechanism that takes input from the room occupants and sets the temperature that maximizes comfort. Both options require user authentication. The authentication service is similar to Kerberos [20] and OAuth [9], wherein an authentication token is generated and is passed on to other services.

HBCI services are essentially web applications that ship the presentation layer to the mobile phone. While the presentation layer may be shipped in HTML and displayed through the web browser, the phone can choose to optimize the presentation layer with native code. In addition, some services may *require* shipping executable code for accessing sensor data. The code could be executed to fetch the necessary sensory information to infer user context. For instance, it could access accelerometer data from the mobile phone to decipher whether the user is moving. An execution framework like PRISM [5] could be used to execute untrusted programs in a sandbox, providing a level of immunity to security vulnerabilities.

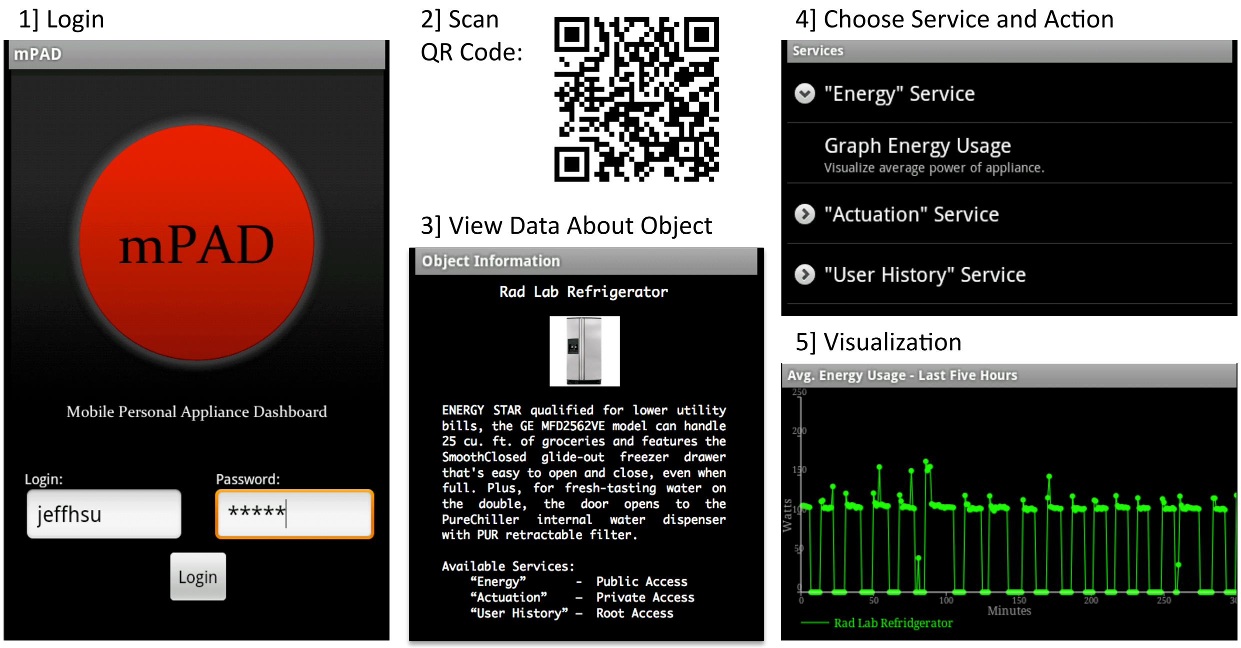
4 Implementation

We chose QR codes as the choice of tags to identify objects for the implementation. Not only are QR codes physically compact, but also they can hold up to 4,296 alphanumeric characters and be decoded at very high speeds [13]. QR codes can be easily printed on paper and attached to any object, reducing the upfront costs of adding an object into the HBCI database. Finally, many smartphones provide a QR code decoder. We use the ZXing decoder [2] on the Android mobile operating system. The advantages of high decoding speed, small size, and high density make QR codes an ideal choice for storing any URI metadata.

In order for the tag to be easily accessible to a user, a QR code is placed on a visible part of an object. For objects that refer to rooms, the QR code is placed on any doors that lead to the room. For example, various barcodes have been put on the front door of UC Berkeley's Computer Science Building (Soda Hall), the front door of the Rad Lab office in Soda Hall, desk lamps, desktop computers, televisions, a coffee machine, a refrigerator, and other appliances.

Our mobile application, the "mPAD" (Mobile Personal Appliance Dashboard), can be installed on any Android OS phone. Figure 3 shows screenshots of the application, following the process flow shown in Figure 2. To initially use the application, the user first enters his/her login credentials on the splash screen. The login data is locally recorded. When a user identifies an object of interest (such as a refrigerator), he/she can scan the object's attached QR code, which encodes a URI. The web resource identified by the QR code's URI is then acquired RESTfully by the mobile device. The resource contains general data of the services offered, which is then presented to the user on the phone. The user may select an action, which may require authentication using the login information. In Figure 3, the "Energy" service's "Graph Energy Usage" action is selected, which allows the user to view a graph of the average amount of energy consumed by the refrigerator in the recent past.

We chose JSON as the data representation for transmitting structured data between a server and a smart-



Rad Lab Refrigerator

ENERGY STAR qualified for lower utility bills, the GE MFD2562VE model can handle 25 cu. ft. of groceries and features the SmoothClosed glide-out freezer drawer that's easy to open and close, even when full. Plus, for fresh-tasting water on the double, the door opens to the PureChiller internal water dispenser with PUR retractable filter.

Available Services:
“Energy” - Public Access
“Actuation” – Private Access
“User History” – Root Access

Figure 3. Screenshots of our mobile application, the mPAD (Mobile Personal Appliance Dashboard).

```
Name :  Jeff's Desktop Computer
ID : 361
Owner :  jeffhsu
Type :  Object
Subtype :  Desktop Computer
Smap URL : http://smap.berkeley.edu/…
Description :  2.66 Ghz Intel Core 2…
Space : Rad Lab
Subobjects: Dual LCD Monitor,  PC Tower
Services :
      [{ Service:  Energy,
          Actions :  [Graph Energy Usage] ,
          Permission : Public Access } ,
        { Service:  Actuation,
          Actions :  [Turn On, Turn Off] ,
          Permission : Private Access } ,
        { Service:  User History ,
          Actions :  [View Logs] ,
          Permission : Root Access } ]
```

```
Metadata: {
       { Energy Service:
              # Energy usage per min (watts)
              { 08 June 2010 :
                     [0, 0, 0, 519, 747, 1150,
                      1097, 1026, 1018, 979…] }
              { 07 June 2010 :
                      …    }
       { User History Service:
              # Log of user actions
              { 08 June 2010 :
                      User :  jeffhsu ,
                      Action :  Turn On ,
                      Timestamp :  1275998800 }
              { 07 June 2010 :
                      User :  prashmohan ,
                      Action : Graph Energy Usage,
                      Timestamp :  1275939909 }
              …          }                    }
```

Figure 4. A simplified version of the JSON schema of a desktop computer.

phone. It could easily be in other representations such as XML or binary, where appropriate. The JSON schema of an object's respective web resource stores information about the services it provides to users. Figure 4 illustrates a simplified version of the schema of a desktop computer. A privately owned resource, the desktop computer has multiple services: the “Energy” service, the “Actuation” service, and the “User History” service. To increase awareness of a user's energy footprint, the “Energy” service allows a user to analyze a real-time graph of the average power used by the object, while the “Actuation” service allows a user to remotely turn on or off the HBCI object. Both services are enabled by technologies we designed at UC Berkeley: the ACme [14], a wireless IPv6-based plug-load meter and actuator, and sMAP, an application layer protocol to retrieve data streams from sensors [6]. In a complex environment such as a commercial building where there are nonuniform classes of sensors, meters, and actuators monitoring thousands of sense points at various granularities, sMAP abstracts these myriad data sources and allows uniform access, enabling HBCI services such as the “Energy” service to rely on a certain set of semantics. When a user chooses the “Energy” service to graph the energy usage of a certain object from the menu, the mobile application queries the sMAP web service for an array of data points of energy usage. This data is then visualized onto the phone using the open source library AChartEngine [1]. Using the “Actuation” service for an object involves sending a JSON object (either State : On, or State : Off) to its sMAP service in order to control the on/off state of the device. An exception for ACme actuation are computers; in our implementation, Auto Shutdown Manager [8] is used to intelligently start and shutdown computers. Finally, the “User History” service of an object displays a historical log of user actions performed onto the object. As seen in Figure 4, any logged information by a service is recorded in the “Metadata” portion of the JSON schema.

There are three types of access scopes: public access, private access, and root access. Basic HTTP authentication is used to allow the user to perform an action of a service. As shown in Figure 3's “object information” screenshot, the “Energy” service is publicly accessible, allowing all users to analyze the energy usage data of the refrigerator. However, while the owner of the object is allowed to actuate the object, only the root user is able to access the user history logs of the refrigerator.

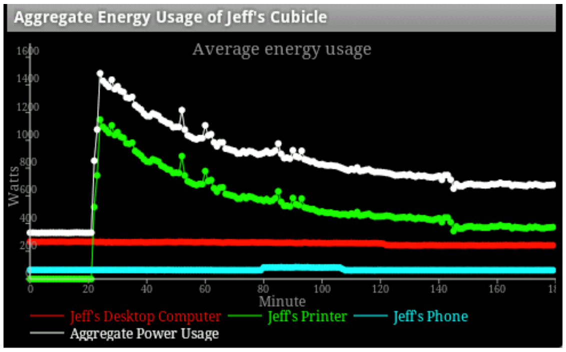


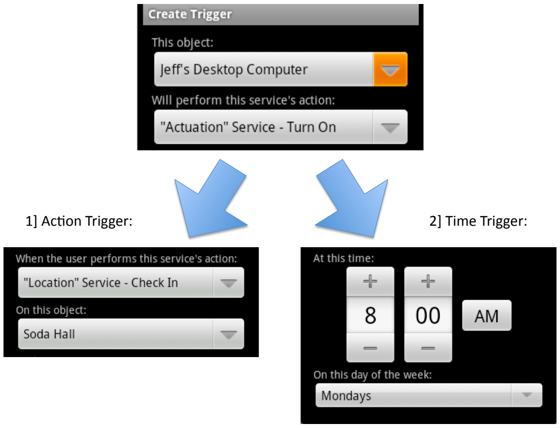
Figure 5. Using the "Energy" service of a space object, a user can graph the aggregate energy usage of all appliances contained within the space (in this case, a work cubicle). The top line represents the total energy usage of all three appliances.

Objects that refer to spaces also provide multiple services. The "Energy" service of a work cubicle allows the user to graph the aggregate energy usage of all objects inside the cubicle – an emulated graph is shown in Figure 5 . The "Occupancy" service of the object representing the Rad Lab shows the current occupancy of the research lab and allows for initiating historical queries. Primarily designed for a building manager, this service aggregates all "User History" service logs of every object in the laboratory. The "Location" service allows occupants to "check in" and "check out" of spaces, which serves as a crucial component for our system of triggers, explained next.

The mobile application provides multiple user-specific functionalities that serve to maximize a user's individualized comfort while minimizing total energy usage. The application provides 1) remote automated actuation through triggers and 2) energy consumption analysis through apportionment.

A trigger is defined as an automated action that is triggered whenever a certain condition is fulfilled. Through our system of triggers, every user has the ability to manually create his/her own user profile to automate appliances. For instance, Figure 6 displays the configuration of an *action trigger* that involves turning on a desktop computer whenever the user checks in to a building using the "Location" service. Inversely, the user can create a trigger that involves automatically turning off their desktop computer when checking out of the building. The system also allows for triggers to be set off based on conditions of time. For instance, as shown in Figure 6, a user is able to set a *time trigger* that turns on his/her cubicle lights every morning at 8:00AM in advance, right before the user arrives to the building for work. The advantage for the trigger system is twofold: 1) comfort automation and 2) energy optimization.

The second functionality of the mobile application allows a user to analyze his/her own energy consumption through accurate apportionment of energy. The mobile application can provide an accurate summary of a user's daily energy consumption. The "Energy" service logs the energy consumption of all objects in the metadata portion of the JSON schema, and a record of a user's owned objects is tracked. In combination with the energy consumption of all owned objects, attributing a share of the energy consumed by a shared resource to a user forms the total aggregate energy used by a building occupant. To increase awareness of a user's energy footprint, a summary of this information is presented to the user in Figure 7. Apportionment is a strong tool for incentivizing a user to reduce his/her energy consumption [10].


Create Trigger

This object: Jeff's Desktop Computer

Will perform this service's action: "Actuation" Service - Turn On

1] Action Trigger:

When the user performs this service's action: "Location" Service - Check In

On this object: Soda Hall

2] Time Trigger:

At this time: 8 00 AM

On this day of the week: Mondays

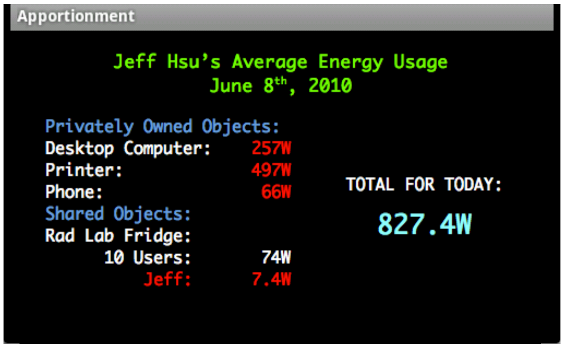
Figure 6. There are two different types of triggers a user can create: 1) an action trigger that is set off by another service's action 2) a time trigger which is set off during a set time.

We emulated a study of a user using the mobile app. In his work cubicle, ACme meters are connected to his desktop computer, dual LCD monitors, and cubicle light. We make multiple assumptions about the user: 1) The user only utilizes all owned devices only from 9AM to 5PM on weekdays, but does not turn them off outside these hours and 2) all devices in the cubicle are only used by him. Using the apportionment and trigger features, the user is able to decide which devices to automate to save energy. Under these assumptions, we calculated that, with the mobile application, this user can roughly save 67% of one week's worth of energy usage by automating the actuation of his appliances.

While accurate apportionment and display of energy can motivate a user to save energy, triggers minimize a user's energy waste by automatically managing the states of their objects. It also allows users to automatically manage their comfort.

5 Conclusion and Future Work

This paper demonstrated the architecture and implementation of a Human-Building-Computer Interaction



Apportionment

Jeff Hsu's Average Energy Usage
June 8th, 2010

Privately Owned Objects:
Desktop Computer: 257W
Printer: 497W
Phone: 66W
Shared Objects:
Rad Lab Fridge:
10 Users: 74W
Jeff: 7.4W

TOTAL FOR TODAY:
827.4W

Figure 7. Accurate apportionment of a user's energy footprint can lead to increased awareness and ultimately energy reduction.

system offering anticipatory, adaptive, and personalized services to the user. The implementation of the mobile application provides an iteration of this architecture that offers multiple services to a building occupant to maximize his/her comfort while minimizing energy waste.

Currently, a real working setup of the implementation has been deployed in our research lab. Multiple users have smartphones with this mobile application installed. In addition, numerous appliances are already connected to ACmes and they are sMAP enabled. Future work involves implementing a large scale user study of building occupants using this system and analyzing its impact on occupant comfort and energy consumption. To increase the scope of the HBCI system, our lab has multiple developers working on creating new services offered by any object. Because sMAP serves as an uniform application layer for all types of sensor streams (temperature, humidity, motion, sound, etc.), numerous innovative services can be engineered to support building occupants in their everyday activities.

We also plan to utilize the logs provided by services as a feedback loop for user profiling. For instance, context learning algorithms can use the logs from the "Energy", "Occupancy", "Actuation", and "Location" services to automatically infer intentions of a user through anticipating his/her location, routes, and activities with a high degree of accuracy. The same logs can also be used to suggest informed recommendations to users for decreasing overall energy usage without compromising occupant comfort.

Finally, a crucial aspect to be further developed is improving the security of our implementation. Currently, very basic HTTP authentication is used for security; the next step is to implement the authentication token system explained in the architecture section to prevent malicious events from occurring.

Acknowledgments

Special thanks to Albert Goto for much helpful assistance. This work is supported in part by the National Science Foundation under grants #CNS-0932209, an NSF Graduate Fellowship, the FCRP MySyC Center, Intel Corp, and Samsung Corp.

6 References

[1] Achartengine. `http://www.achartengine.org/`.

[2] Zxing ("zebra crossing"). `http://code.google.com/p/zxing/`.

[3] A. Arora, E. Ertin, R. Ramnath, M. Nesterenko, and W. Leal. Kansei: A high-fidelity sensing testbed. *IEEE Internet Computing*, 10:35–47, 2006.

[4] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos. Anonysense: privacy-aware people-centric sensing. In *MobiSys '08: Proceeding of the 6th international conference on Mobile systems, applications, and services*, pages 211–224, New York, NY, USA, 2008. ACM.

[5] T. Das, P. Mohan, V. N. Padmanabhan, R. Ramjee, and A. Sharma. Prism: platform for remote sensing using smartphones. In *MobiSys '10: Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 63–76, New York, NY, USA, 2010. ACM.

[6] S. Dawson-Haggerty, J. Ortiz, X. Jiang, J. Hsu, S. Shankar, and D. Culler. Enabling green building applications. In *6th Workshop on Hot Topics in Embedded Networked Sensors (HotEmnets 2010)*, 2010.

[7] *Energy Outlook 2010*. Energy Information Administration, `http://www.eia.doe.gov/oiaf/ieo/index.html`, 2010.

[8] EnviProt. Auto shutdown manager. `http://www.enviprot.de/`.

[9] E. Hammer-Lahav, D. Recordon, and D. Hardt. The oauth 2.0 protocol. Technical report, Internet Engineering Task Force, July 2010.

[10] S. Hay and A. Rice. The case for apportionment. In *Proceedings of the FIrst ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings (BuildSys 2009)*, 2009.

[11] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen. The gator tech smart house: A programmable pervasive space. *Computer*, 38:50–60, 2005.

[12] S. S. Intille. The goal: smart people, not smart homes. In *In Proceedings of the International Conference on Smart Homes and Health Telematics, IOS*. Press, 2006.

[13] ISO/IEC. ISO 18004: Automatic identification and data capture techniques — QR Code 2005 bar code symbology specification. Final draft, ISO/IEC, 31.-August 2006.

[14] X. Jiang, S. Dawson-Haggerty, P. Dutta, and D. Culler. Design and implementation of a high-fidelity ac metering network. In *The 8th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN2009) Track on Sensor Platforms, Tools, and Design Methods (SPOTS 2009)*, 2009.

[15] S. Junestrand, K. Tollmar, S. Lenman, and B. Thuresson. Private and public spaces: the use of video mediated communication in a future home environment. In *CHI '00: CHI '00 extended abstracts on Human factors in computing systems*, pages 16–17, New York, NY, USA, 2000. ACM.

[16] J. A. Kientz, S. N. Patel, B. Jones, E. Price, E. D. Mynatt, and G. D. Abowd. The georgia tech aware home. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pages 3675–3680, New York, NY, USA, 2008. ACM.

[17] H. Lu, N. D. Lane, S. B. Eisenman, and A. T. Campbell. Bubble-sensing: Binding sensing tasks to the physical world. *Pervasive and Mobile Computing*, 6(1):58 – 71, 2010.

[18] MIT. House n. `http://architecture.mit.edu/house_n/`.

[19] M. Mozer, R. Dodier, D. Miller, M. Anderson, J. Anderson, D. Bertini, M. Bronder, M. Colagrosso, R. Cruickshank, B. Daugherty, M. Fontenot, O. Ikeako, P. Kooros, D. Lukianow, T. Moyer, C. Myers, T. Pennell, J. Ries, E. Skorpen, J. Sloss, L. Vidmar, and M. Weeks. The adaptive house. *IEE Seminar Digests*, 2005(11059):v1–39–v1–39, 2005.

[20] B. C. Neuman and T. Ts'o. Kerberos: An authentication service for computer networks. volume 32(9), pages 33–38, Sep 1994.

[21] L. Schor, P. Sommer, and R. Wattenhofer. Towards a zero-configuration wireless sensor network architecture for smart buildings. 2009.

[22] X. Wang, J. S. Dong, C. Chin, S. Hettiarachchi, and D. Zhang. Semantic space: An infrastructure for smart spaces. *IEEE Pervasive Computing*, 3:32–39, 2004.

[23] G. Werner-Allen, P. Swieskowski, and M. Welsh. Motelab: a wireless sensor network testbed. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, page 68, Piscataway, NJ, USA, 2005. IEEE Press.

[24] D. Yazar and A. Dunkels. Efficient application integration in ip-based sensor networks. 2009.