Platform-integrated Sensors and Personalized Sensing in Smart Buildings

Milan Milenkovic[1], Thanh Dang[2], Ulf Hanebutte[1] and Yonghong Huang[1]
[1]*Intel Labs, Intel Corporation, Oregon, U.S.A.*
[2]*School of Engineering, Washington State University Vancouver, Washington, U.S.A.*

Keywords: Sensing Architecture, Platform-integrated Sensor, User-centric Sensing, Smart Buildings.

Abstract: We propose, implement, and evaluate a pervasive sensing system that is capable of collecting data from sensors that may surround a user in a given setting. Such systems will enable creation of new types of applications that span across devices, users, and domains based on spatial, temporal, and social aggregations of sensor data. Key innovation in our work is a sensing fabric that collects data from a variety of sensors and leverages platform-integrated sensors, which are built into hosting devices, such as laptops and tablets. These sensors can significantly improve sensing in enterprise settings and they are comparatively inexpensive to manufacture, deploy, and maintain. Our system embodies three key architectural principles: (1) support for a variety of sensor types including platform-integrated sensors for pervasive sensing, (2) use of Internet protocols for sensor connectivity, web technologies and programming model for application development, and (3) use a hybrid sensor database design with a document-oriented component to improve flexibility and performance. We evaluate our implementation in real-world pilots for several months and 73 users. Our results demonstrate that platform-integrated sensors can provide accurate sensing data, have negligible impact on operations of a hosting platform, and that our architecture can provide sensing services across users and devices over a sustained period of time.

1 INTRODUCTION

We envision a pervasive sensing system that is capable of collecting data from all available sensors that may surround a user in any setting. The data sources can come from a variety of devices including personal mobile devices as well as sensors embedded in the environment. The data sources can also be in different settings such as static sensors provided by building-management systems (BMS), mobile sensors in moving vehicles with in-vehicle information system, or public infrastructure in smart cities.

Our definition of sensors is very broad; it includes not only all types of hardware sensors, but also software sensors, sensing services, and people. Software sensors are software programs (agents) that can capture and report some conditions of interest, such as user presence detected via key clicks or mouse movements. Sensing services refer to data provided by an external source with programmatic interfaces. People as sensors refers to users providing direct input, such as comfort levels or preferences via dedicated end-user interfaces, or social network platforms.

An additional sensing opportunity, surprisingly not covered in published literature, is provided by (IT) platform-integrated sensors, which are built into personal computing devices such as laptops, phones, and tablets. They offer many advantages over existing sensors in terms of cost, deployment, and personalization. Computing devices already have computation, storage and communication capabilities, and can provide energy to attached sensors. Adding new sensors to IT devices is comparatively inexpensive as it requires adding only physical sensing probes or capabilities. Hence, the manufacturing and deployment costs for adding sensors to computing devices are small compared to standalone wireless embedded sensors or dedicated sensors.

The challenge we are addressing is to integrate data from all sensors including platform-integrated sensors across devices, users, and domains. The integration will enable new types of applications that cross-correlate data related to individuals. It will also enable comprehensive understanding of contexts and environments surrounding users. While much attention in the research community has been on designing sensing systems for dedicated infrastructure (Turc et al., 2010) or for network of standalone sensors (e.g.

Milenkovic M., Dang T., Hanebutte U. and Huang Y..
Platform-integrated Sensors and Personalized Sensing in Smart Buildings.
DOI: 10.5220/0004271300470052
In *Proceedings of the 2nd International Conference on Sensor Networks* (SENSORNETS-2013), pages 47-52
ISBN: 978-989-8565-45-7

participatory sensing (Campbell et al., 2008)), designing sensing systems that span across dedicated infrastructures, mobile phones, and laptops, tablets has not been seriously addressed to date.

In this paper, we propose, implement, and deploy a sensing system that leverages platform-integrated sensors as applied in smart buildings. The system integrates both hardware sensors and software sensors. We have deployed the system in two real-world pilots for an energy efficiency application in buildings. We evaluate the system and show that platform-integrated sensors can enable a large number of practical applications.

2 PLATFORM-INTEGRATED SENSORS AND SENSING ARCHITECTURE

In this section, we present our sensing architecture that integrates cross-domain sensors including platform-integrated sensors for smart buildings.

2.1 Platform-integrated Sensors

In our prototypes and pilot deployments, we make considerable use of platform-integrated sensors as practical enhancement to pervasive ambient sensing. These are physical sensors (e.g. light, temperature and humidity) that are integrated into common computing devices including laptops, tablets, and phones. Platform-integrated sensors have several advantages: (1) low cost, (2) ease of installation, operation, and management especially in corporate environments, and (3) personalization.

Platform-integrated sensors can use the processing power, connectivity, and energy supply of their hosts. Therefore, adding new sensors to devices requires adding only physical sensing capabilities (e.g., temperature sensors). Being essentially part of the hosting computing device, they can be installed and managed in an enterprise using existing IT processes and infrastructure. In comparison, standalone wireless sensors often require a separate communication infrastructure and may require frequent battery replacement. This makes them difficult to install and manage in commercial settings as there is no enterprise deployment processes or infrastructure to rely on, in either IT or facilities management. Wired ambient sensors in enterprises are typically provided as part of the BMS at the time of building commissioning. They tend to work only with the proprietary BMS systems and are costly to add or reprogram.

One aspect of personalization with platform-integrated sensors comes from the close proximity between users and their personal devices. Users can get real-time feedback about their own ambient conditions, such as temperature and humidity *where I am right now* as opposed to sensor data from remote or standalone sensors. Other aspects of personalization in our system include meta-data tagging for user association and correlation across devices and domains. These aspects are part of our implementation but are not described in detail in this paper.

Having realized these advantages, the computer industry is adding a variety of sensors to laptops, starting with Ultrabooks in consumers and enterprise (Intel, 2011). Smart phones started with an earlier use of sensors, but mostly to enhance user experience aspects. Platform-integrated sensors, where available, have been used to support limited vertical applications that do not make use of other sensors that may be surrounding users.

We view platform-integrated sensors as a practical and cost-effective addition to, and not a substitute for, existing sensing infrastructures. However, existing sensing systems do not integrate platform-integrated sensors well because they have several distinct characteristics and requirements including: accuracy, intermittency, mobility and localization, user experience, self-discovery, and security and privacy.

2.2 Sensing Architecture Overview

Our primary design objective is to acquire sensor data from any source and across different sensing infrastructures, including wired and wireless standalone sensors. The architecture is not limited to specific types of communications links. Figure 1 shows an overview of our sensing framework in smart buildings with existing BMS.

We assume that sensor sources are IP-enabled and can support the HTTP protocol in particular, directly or via proxies. Gateways may be deployed to bridge specialized sub-domains, such as ZigBee or 6LoWPAN sensor networks. One or more software programs called *sensor agents* monitor each IT device or group of devices. A sensor agent can represent and abstract a hardware sensor or a software sensor. The agents report readings to a sensor database. In our system, sensor data integration occurs at the database level.

The key characteristic of our system that provides cross-platform and cross-domain sensing capability; It allows integration of sensor data across sensors, devices, individual users. Sensor readings stored in our database are tagged with metadata and identifiers.

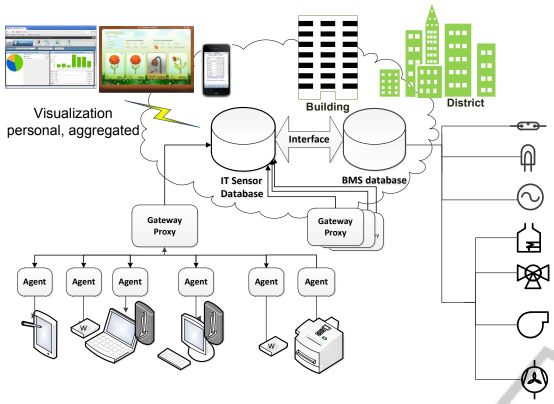


Figure 1: Sensing architecture for platform-integrated sensors.

This allows our system to associate the sensor readings with specific time and place of capture, as well as information about originating devices, and attribution to individual users. In this way, we can aggregate data of interest, that is collected by a multitude of devices, to a specific user. This is done by virtue of ownership or temporal association. For example, close proximity can be determined while users are nearby sensors embedded in the surrounding environment (domain), such as office, home, or car.

The sensor database provides an efficient way to query and aggregate data, and interact with other systems. We can perform analytics on individual data and aggregated data. The database can respond to a variety of programmed and ad-hoc queries to support applications and visualizations in various levels of detail. In addition, the sensor database also provides an interface for other systems, such as BMS database as shown in Figure 1, to exchange sensor and control information. Thus, it enables holistic building management through coordinated behavior of all control systems. Currently various control systems in buildings, such as IT and BMS, operate in isolation, resulting in sub-optimal building management.

Figure 1 is conceptual in that it depicts sensors in and around IT devices, such as laptops. In practice, we use a variety of sensors including software and hardware sensors in laptops and other mobile devices, as well as stand-alone sensors, such as power meters, embedded in the environment. Our system architecture and communication protocols are intentionally designed to accommodate these and other wired and wireless sensors, and aggregate data of interest across all relevant sensors to users. Three important features of the architecture are (1) cross device sensing using Representational State Transfer RESTful sensor agents, (2) decoupling sensor sources and sensor database using publish-subscribe communication, and (3) providing flexibility for changes by using a document-oriented (also referred to as noSQL) database for sensor data storage. The following sections describe these features in detail.

2.2.1 RESTful Sensor Agents for Cross-device Sensing

All our sensor agents implement a simple measurement and actuation protocol called sMAP (Dawson-Haggerty et al., 2010), that provides a RESTful web service. The key features of a RESTful web service that is applied in our architecture are: stateless, cacheable, and uniform interface. The sensor agents rely on HTTP-compatible communication with sensors and actuators. This allows our sensors to report data over Internet connections

Platform-integrated sensor agents convert hardware and software sensor data into sMAP format and send the data over the Internet to our sensor database. Other types of hardware sensors may be interfaced directly if properly formatted or through proxies and protocol converters. *People as sensor* inputs, such as direct observations, expressions of comfort and preferences may be collected via dedicated end-user interfaces or with appropriate gateways from commonly accessible texting and social networks.

In addition, researchers from the LoCal group at the University of California Berkeley are using a completely different set of wireless ACme sensors (Jiang et al., 2009) with microcontrollers that produce sMAP formatted data directly and report them, via a networking gateway, to a similarly structured database. This provides an independent confirmation to our claim of architectural capability to support different types of sensors across different communication networks.

2.2.2 Publish-subscribe Communication

In order to decouple consumers and producers of data, we use a publish-subscribe sensor reporting model. In particular, we treat sensor sources as publishers and consumers as subscribers. In this way, a variety of building applications can be dynamically constructed by having them subscribe to different data sources of interest.

An interesting possibility availed by this approach is that a sensor database can be implemented without any knowledge or explicit assistance from sensor agents. Namely, we create an universal receiver process that subscribes to all sensors and deposits reported data to the sensor database. This process completely decouples sensors from the sensor database, thus allowing different database implementations to be substituted at the server back-end side, without reconfiguring the sensor source at the front-end side. This kind of system flexibility stands in stark contrast to traditional proprietary supervisory control and data acquisition (SCADA) and BMS systems where adding or even modifying a few sensory inputs to a running system requires custom programming by specialized people resulting in very high end-user costs. These systems generally assume a static configuration of wired sensors, which makes them brittle and unable to support mobile users with platform-integrated sensors.

2.2.3 Document-oriented Sensor Database

On the server side, we use a variant of StreamFS (Ortiz, 2012) as a universal subscriber to all sensors to handle sensor data streams and store sensor records in the sensor database. In our view, key requirements for the sensor database are high throughput, scalability, and some flexibility in data structure. In the current implementation, we use MongoDB, a document-oriented noSQL database (MongoDB, 2011) for sensor data storage.

In document-oriented database, each data record is stored as a document. In general, document-oriented databases do not provide transactional capability with ACID (atomicity, consistency, isolation, and durability) properties. This results in lower write and little or no locking overhead, thus resulting in higher read/write throughput than traditional transactional and relational databases. The tradeoff is lack of transactional and recovery capability, meaning that some data records may be lost in case of system crashes. This is generally not a problem for streaming sensor data due to a large number and relatively minor significance of individual readings and post-processing ability to approximate the missing readings in a sequence. In addition, document-oriented noSQL databases do not require fixed formats and pre-specified schema. Therefore, new attributes can be added dynamically without affecting the whole system. This flexibility allows us to add new sensors and change sensor data format with minimal effort. Nevertheless, we also use a relational database to store associations, metadata, and to create analytic reports for users and system operators. The accessing frequency for this database is much lower than the sensor database.

2.3 Case Study: Eco-Sense Buildings (ESB)

The initial use case of our platform-integrated sensing architecture is eco-sense buildings (ESB). The prototype and end-user testing target is energy efficiency in office buildings. Our project goal is to expand the role of IT to participate in increasing energy efficiency of office buildings, while improving comfort and safety of their occupants. We are working with eco-system partners worldwide towards meeting the most aggressive energy target, net-positive energy buildings (GIE, 2011).

We choose buildings for both their opportunities and challenges. Buildings are a major energy consumer of energy. Today, buildings consume 42% of total energy and 72% of electricity (OSC, 2011). Given the increasing demand on energy and its impacts on the environment, it is critical to achieve energy efficiency in buildings. Design-time energy optimization, while important, is static. To maintain actual desired operational efficiency throughout a building lifetime, a process of constant commissioning that continuously monitors and adjusts the building's operation and control settings is necessary.

2.3.1 ESB Implementation

In our initial prototype, we focus on leveraging IT infrastructure to measure personal energy consumption and personal ambient conditions (i.e., temperature, light, and humidity wherever the user with the instrumented laptop happens to be). The system includes sensor DB server, PC clients and other devices. In the forthcoming end-user deployments, we plan to use office laptops with built-in ambient sensors. For initial prototyping, we use an external sensor kit attached to each laptop via an USB port (Figure 2). In addition to platform-integrated sensors, we also apply soft sensors and commercial meters to monitor energy consumption of devices. All sensors and sensor agents communicate with the database over existing Ethernet and Wi-Fi infrastructure, thus, simplifying provisioning and deployment.

For personal power and energy measurement, we use a combination of custom-made software sensors and commercial meters. We developed a software sensor that is installed on a monitored platform. The software agent accurately estimates the platform energy consumption (currently PCs and laptops) by tracking occupancy of computer power states defined by EnergyStar (EnergyStar, 2011) and ECMA-383 (ECMA-383, 2011), such as short idle, long idle and sleep. Taking into consideration battery status and modeling the appropriate power draw, which can increase by a factor of $2-3$ when the battery charge is low, provides the added accuracy. This allows us to eliminate the need for dedicated hardware power meters to measure PC power draw and energy consumption in real time.

We also measure energy of other devices such as external monitors and printers at device and user levels by using externally attached commercial energy meters. These meters are network-enabled. They post data using existing corporate intranet.

Figure 2: Platform-attached prototype sensor hub used in our experiments.

3 EVALUATION

We have implemented and deployed the architecture in two buildings with real office users in France and Japan for approximately three months. The building in France has 5 floors with the total area of $27,146$ square feet. The pilot site in Japan occupies a single floor of $42,305$ square feet in a multi-story commercial building. There were 73 participants in total with 73 PCs and 21 printers instrumented with energy meters. Each of the participant's IT-supported client system was supplemented with a prototyped platform-integrated sensor. The sensor collected ambient data including temperature, humidity, and light. Furthermore, a software energy sensor was installed in each PC to monitor the PC energy consumption. A commercial power meter was installed for each printer to monitor its energy consumption. Raw sensing data were collected every 30 seconds and stored in a sensor database.

We analyze sensing data collected from sensors that were attached to a platform (e.g., laptop) and a reference standalone sensor that was placed nearby. The reference sensors were calibrated by the National Institute of Standards and Technology(NIST). The results and analysis are in Section 3.1.

To investigate whether the architecture is capable to collect data over a large scale and for a long period of time, we collected three months of data from 73 PC users and 21 printers from two pilot sites. We remove the outliers and perform the statistical analysis at individual and population levels. For ambient data, we calculate the statistics, mean and standard deviation (std), for each individual user.

3.1 Accuracy of Platform-integrated Sensors

Figure 3 shows temperature profiles during off hours in a real office deployment. A sensor was attached to a laptop at the upper right hand corner of the display lid. The laptop was placed on an office desk. The reference sensor was placed nearby. Both sensors show the gradual temperature drop at night and fast temperature rise in the morning. The statistical analysis of this profile yield for the reference sensor $24.28C+/-1.35C$ and $24.04C+/-1.56C$ for the platform attached sensor, which is a relative difference in the mean of less than 1%. We can see that the platform-integrated sensors follow the reference sensor trending very closely. We ran similar tests to evaluate our platform-integrated sensors with the reference sensor on humidity and light, and obtained similarly accurate results.

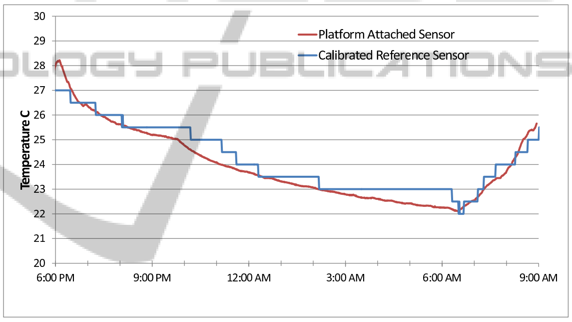
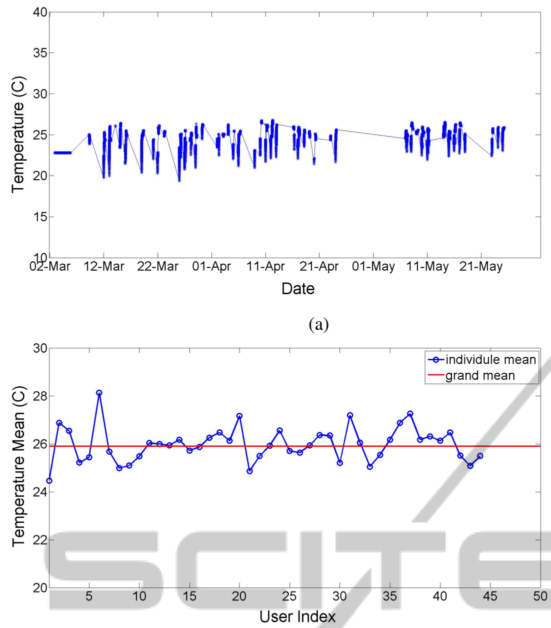


Figure 3: The accuracy of our platform-integrated sensor prototype compared to a NIST calibrated reference sensor on temperature during off hours in an office deployment.

3.2 Pilot Data Analysis

We deployed our platform-integrated sensors to real-world pilots to evaluate its feasibility in long-term data collection for smart buildings. The sensors monitor the ambient condition, including temperature, humidity, and light. Figure 4 shows the temperature data collected from the platform-integrated sensors in one pilot. Figure 4(a) shows the temperature raw data collected by the platform-integrated sensors for one randomly selected user from March to May 2012. There are small variations of temperature for each day and across different days for this particular user. The temperature for this user is at the range of 20C to 27C, which is within the comfortable range. Figure 4(b) shows the statistics of the temperature data for 45 users. The individual mean and the population mean are shown. The grant mean temperature is $26C$. We use similar approaches to validate the humidity data and light data. These results demonstrate



(a)

(b)

Figure 4: The temperature data collected from the platform-integrated sensors in one pilot. (a) The temperature data for a randomly selected user. (b) The mean of the temperature data for 45 users. The blue dot is the temperature mean for an individual user and the red line is the grand mean across all users.

that platform-integrated sensors are capable to collect continuous ambient data including humidity, temperature, and light over a long period of time for a large-scale deployment. The data provide meaningful information for a building management system to adapt building operations to improve energy efficiency and user comfort in buildings.

4 SUMMARY

We have presented the case for platform-integrated sensors as a component of pervasive sensing in office buildings. We describe our sensing fabric that supports cross-user and cross-device sensing. Three key architectural principles in our implementation are: (1) leveraging platform-integrated sensors for personal sensing and crowd-sourcing (2) using of Internet protocols for sensor connectivity, web technologies and programming model for application development, and (3) using a hybrid sensor database design with a document-oriented component to improve maintenance flexibility. Our experiments in real-world pilots confirm the accuracy, feasibility, and applicability of the architecture in practice. Our work is a step closer to realizing a vision of a pervasive sensing system capable of collecting data from all available sensors that may surround a user in any setting. Our current work continues with personalizing user sensing experience and crowd-sourcing (described elsewhere) and in addressing new requirements of platform-integrated sensors due to their distinct characteristics, including host-attachment constraints, mobility, and reporting intermittency.

ACKNOWLEDGEMENTS

Many people contributed to this work, including Scott Shull, Mark Chang, Yves Aillerie, Sylvain Sauty, Tomm Aldridge, and individual and corporate members of the Positive-Energy consortium (GIE, 2011) (Bouygues, Intel, Lexmark, Philips, Schneider Electric, Sodexo, Steelcase, Tenesol) and Oregon Sustainability Center (OSC) (OSC, 2011).

REFERENCES

Campbell, A. T., Eisenman, S. B., Lane, N. D., Miluzzo, E., Peterson, R. A., Lu, H., Zheng, X., Musolesi, M., Fodor, K., and Ahn, G.-S. (2008). The rise of people-centric sensing. *IEEE Internet Computing*, 12(4):12–21.

Dawson-Haggerty, S., Jiang, X., Tolle, G., Ortiz, J., and Culler, D. (2010). smap: a simple measurement and actuation profile for physical information. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 197–210, New York, NY, USA. ACM.

ECMA-383 (2011). Standard ecma-383: Measuring the energy consumption of personal computing products.

EnergyStar (2011). Energystar, http://www.energystar.gov/.

GIE (2011). Gie (consortium) eep, positive-energy challenge,, http://www.enjeu-energie-positive.com/.

Intel (2011). Ultrabook inspired by intel, http://www.intel.com/ content/www/us/en/sponsors-of-tomorrow/ultrabook.html?cid=sem117p8085.

Jiang, X., Dawson-Haggerty, S., Dutta, P., and Culler, D. (2009). Design and implementation of a high-fidelity ac metering network. In *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, IPSN '09, pages 253–264, Washington, DC, USA. IEEE Computer Society.

MongoDB (2011). Mongodb, http://www.mongodb.org/.

Ortiz, J. (2012). Stream fs, http://streamfs.cs.berkeley.edu.

OSC (2011). Oregon sustainability center, http:// www.oregonsustainabilitycenter.org/.

Turc, T., Gligor, A., Dumitru, C., and Morar, A. (2010). Development of extensible virtual instruments for scada applications. In *Proceedings of the 2010 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR) - Volume 03*, AQTR '10, pages 1–5, Washington, DC, USA. IEEE Computer Society.