Towards a Wiser Use of Intelligence: Fieldwork in the Application of Information Technology in a Commercial Building

Carl Blumstein and Therese Peffer
California Institute for Energy and Environment, University of California

ABSTRACT

Information technology can increase energy efficiency by improving the control of energy-using devices and systems. Awareness of this potential is not new—ideas for applications of information technology for energy efficiency have been promoted for more than 20 years. But much of the potential gain from the application of information technology has not yet been realized. In an earlier paper one of the authors discussed some reasons for the slow exploitation of information technology's potential to increase energy efficiency. The earlier paper also suggested that a combination of new requirements for the operation of the electricity system and the development of new technology could cause a rapid increase in the pace of adoption. In this paper we describe an application of these ideas to the operation of a commercial building. First, we review basic concepts with emphasis on an open software-architecture. Then we describe the components of this open software-architecture and its ability, for example, to nimbly add sensors and add control algorithms within a proprietary Building Automation System (BAS). Finally, we describe results from this application to Sutardja Dai Hall at the University of California Berkeley. Results include demonstration of reduced peak demand by 20 -30% and continuous energy savings of 20%.

Introduction

This paper is about *open software-architecture*[1] for the control of energy use in buildings. Open software-architecture is a way of organizing the software that links together the physical elements of a building control system to allow the addition of other systems or components. The reason we are concerned about open software-architecture is that open software-architecture is the key to creating an environment that supports innovation. Proprietary and closed systems, which are prevalent today, typically create barriers to innovation.

To make this clear, consider a commercial building with a control system for its Heating Ventilating and Air Conditioning (HVAC) system. If you want to control the lighting in the building, the technology currently used for HVAC control cannot easily be modified for lighting control—in practice you need to add a completely separate control system for lighting. Further, if the control system for lighting includes occupancy sensors and you want to use occupancy to control HVAC, you cannot, as a practical matter, use the lighting control system's occupancy sensors. Still further, if you develop new software for detecting faults in the HVAC system, you cannot easily install the new software in the existing building control software. These are all problems that can be solved with open software-architecture.

[1] Readers should be careful to distinguish between open software-architecture and open-source software. Open software-architecture does not necessarily involve open-source software.

We will have more to say about how we addressed these problems in Sutardja Dai Hall on the UC Berkeley campus later in this paper. First we discuss in more detail the idea of open software-architecture, drawing on lessons from the Internet.

Lessons from the Internet[2]

The most important lesson from the Internet is interoperability—the ability of the Internet to accommodate diverse devices and systems and enable them to work together. The practical effect of interoperability is that equipment suppliers and software developers can compete to supply established needs and can innovate to create new uses. This environment has fostered both cost reductions and rapid innovation. So, one may well ask, can we make building monitoring and control systems look like the Internet? The answer is, yes we can.

Doing this is facilitated by using the Internet's open architecture and Internet protocols. The critical step is to move from a vertical architecture to a horizontal architecture—an essential element of open architecture. Figure 1 provides a simplified representation of horizontal layered architecture[3] to help explain the concept. Each layer is independent, and thus creates modularity. The bottom layer in Figure 1, here called the sensor/actuator layer, is the interaction with physical data and systems; the monitoring and control system interacts with the building environment, gathering data and executing control actions. The middle layer—data management—organizes, stores, and transmits data from the sensor/actuator layer and instructions from the application layer. The top layer, here called the application layer, has software applications that operate on data provided through the data layer to provide outputs in the form of information on the state of the building and instructions for the control of building systems. Not all control is initiated on the application layer; some happens autonomously on the sensor/actuator layer—for example, lights might be directly controlled by an occupancy sensor. And not all instructions from the application layer are accepted. For example, a smoke alarm may override an instruction to open a damper. Within the layers a variety of languages (protocols) may be used for communication, but between layers a single language (protocol) is used for communication—Internet protocol (IP). To make this more concrete, consider a building appropriately equipped with sensors, actuators, and applications. Suppose that the operator of the building wishes to minimize energy use during the peak time on a hot day by precooling the building so it can ride through the peak time. An application in the application layer contains a model of the building that can predict the best time to turn on the chillers based on the outdoor temperature, the indoor temperature, the weather forecast, and other variables all of which are resident in a database in the data layer. The application gets the data from the database and predicts the best time to turn on the chillers, say, 7:00AM. If sensors and controllers in the sensor/actuator layer determine that operation is safe, the chillers will be turned on at 7:00AM.

The difference between horizontal and vertical architecture is not in the functions that need to be performed. Sensing and actuating, data management and applications need to happen in monitoring and control systems regardless of the architecture. The difference is in the separation of these functions. In a vertical system a "black box" might, for example, have hard-

[2] This section adapted from Blumstein 2011

[3] The phrase "layered architecture" does not refer to spatial relationships among the system's components; rather, it refers to logical relationships. The "layers" are an abstraction. Here we are using the word "layers" as a heuristic; it has more specialized meanings in other contexts.

wired connections to sensors and actuators and have applications with built-in data structures that were inaccessible to other applications. Horizontal layered architecture keeps the functions from becoming entangled and allows devices and software from different suppliers to inter-operate.

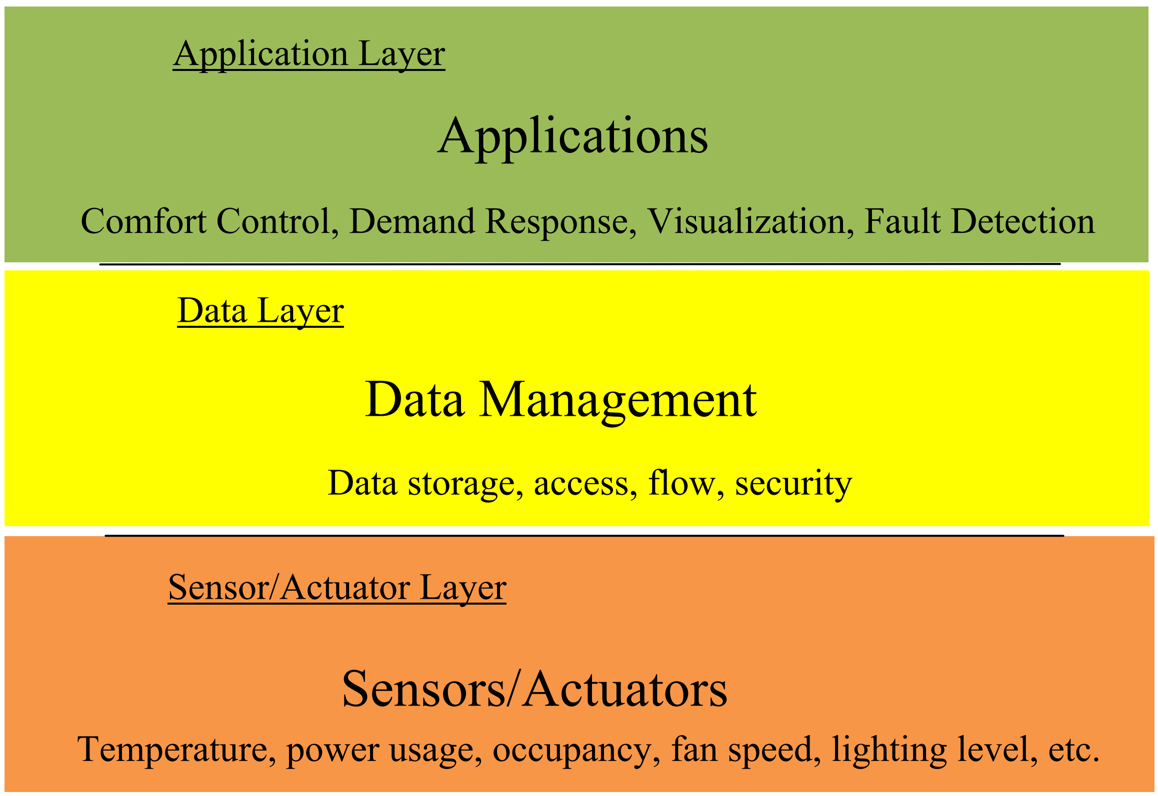

Figure 1. A simplified representation of layered architecture for building monitoring and control.

Sutardja Dai Hall—A Living Laboratory

Over the past four years, several researchers have implemented a prototype open software-architecture for energy management software in a building at UC Berkeley. Sutardja Dai Hall (SDH) is the headquarters of the Center for Information Technology Research in the Interest of Society (CITRIS). The building has approximately 141,000 gross square feet of space that houses laboratories for collaborative research, open plan and private offices, a 149-seat auditorium, conference rooms on each of seven floors, state-of-the-art classrooms, a data center, and 12,000 square feet dedicated to the Marvell Nanofabrication Lab. The building also houses the Main Distribution Center for the northeast quadrant of campus.

The SDH HVAC system is Variable Air Volume (VAV) with reheat. The HVAC system is controlled by a Siemens Apogee Building Automation System (BAS). The building has two 600 ton Trane chillers—one an absorption chiller and the other a centrifugal compressor chiller. The absorption chiller was designed to use steam in warm months (April through October) when steam on the UC Berkeley campus from the 30MW co-generation facility is not in high demand for heating. The centrifugal compressor chiller with hot gas bypass was designed to be used in winter (November through March).

The WattStopper lighting system (overhead fluorescent) in the open plan offices (found on floors 4-7) has tri-level dimming capability and is on a timed schedule. The private offices have Lutron wall switches with dimming and an occupancy sensor.

Sutardja Dai Hall also has 30 revenue grade DEM 2000 power submeters with Ethernet connections on most subpanels (including submeters for lighting and receptacle power on each floor). These measure energy (kilowatt-hours), voltage, current, power factor, and peak demand. The chilled water has flow meters and temperature sensors to determine thermal energy consumption.

Monitoring and Trending

The Apogee BAS and the WattStopper lighting system have limited capacity for data archiving and visualization; in addition data from other sensors in the building are required to obtain a comprehensive picture of energy use in SDH. To bring all of these data together, in 2010 graduate students in the Electrical Engineering Computer Science (EECS) Department at UC Berkeley implemented an open software-architecture solution. The EECS team made the data from the Apogee system, the WattStopper system and the other SDH submeters available using open-source software known as sMAP (simple Monitoring and Actuation Profile) (Dawson-Haggerty *et al*. 2011). The EECS team designed and implemented a data acquisition architecture and wrote a BACnet-to-sMAP converter that makes HVAC-related streams available. This has enabled all of the BACnet data points from the building to be monitored continuously and made accessible via an open interface (http://new.openbms.org/plot/). The WattStopper lighting control system was integrated with the Siemens Apogee system through a BACnet interface, and all of the BACnet data points likewise enabled.

sMAP spans the connection between the sensor and actuation level and data management level in Figure 1. The sMAP drivers provide the communication and the time series database is the repository for many different kinds of data from sources in the sensor/actuator layer. Querying and external visualization of the data is provided by plotting software in the application layer in Figure 1. sMAP provides several features: a specification for transmitting physical data and describing its contents, a large set of free and open **drivers** communicating with devices using native protocols and transforming it to the sMAP profile, and tools for building, organizing, and querying large **repositories** of physical data. The core object in sMAP is the *Timeseries*, a single progression of (time, value) tuples. Each Timeseries in sMAP is identified by a UUID (universally unique identifier), and can be tagged with metadata; all grouping of time series occurs using these tags. These objects are exchanged between all components in this ecosystem (Dawson-Haggerty *et al*. 2011).

At the sensor/actuation level, we have added additional submetering points, also accessible through sMAP. The EECS team deployed an IPv6 wireless sensor network of ACMe plug load receptacle power meters and control relays using the Berkeley Low Power IP stack (BLIP 2.0) created by the EECS team. In addition, they deployed additional data loggers for monitoring environmental conditions during test runs. There is an extensive mote-based wireless test bed throughout multiple floors of the building. Finally, a recent project with the Korean Micro Energy Grid (KMEG) installed over 500 Korea Electronic Technology Institute (KETI) environmental sensors (temperature, light, humidity, carbon dioxide, motion (passive infrared)) throughout the building, in a 6lowPAN wireless mesh network, also accessible through sMAP.

Open Building Control Architecture—the BOSS Example

The EECS team developed the Building Operating System Services (BOSS) (Krioukov *et al*. 2012, Dawson-Haggerty *et al*. 2013), similar to the simple three layer model in Figure 1, with a flexible layered multi-service open software-architecture. This architecture dramatically reduces the effort to add new applications at the top layer (shown below as the Building Application Environment, (Taneja *et al*. 2013)), and supports sensor and actuator access at the bottom layer. Note that in this instance of BOSS, the HVAC control system (Siemens) and the lighting control system (WattStopper) are in the sensor/actuator layer. Of course, these control systems are not simple sensors or actuators—they are full stacks, including the application, data and sensor/actuator layers. This overlay is a "workaround" that addresses the problems we discussed in the Introduction, that is, to allow information from the sensors from one system to inform the actuation of another system. The middle layer (Building System Services) includes access management, metadata, archiving, and discovery of new devices, as well as multiple simultaneously executing programs. Its layered open architecture provides interoperability while preserving reliability (Figure 2).

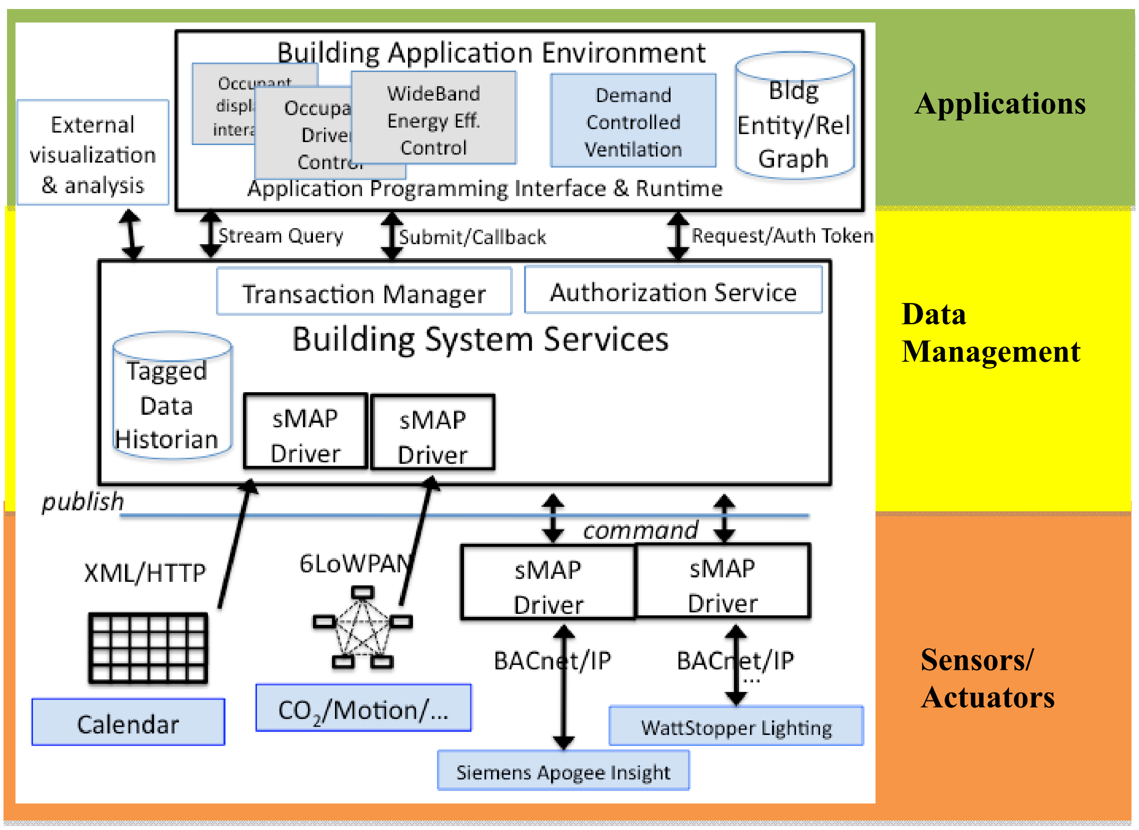

Figure 2. Building Operating System Services (BOSS): An open, layered distributed structure to bring advanced control and integrated operation into existing and new building stock. Note that in this instance BOSS is an overlay on the Siemens Apogee system and the WattStopper Lighting system, which are vertically integrated stacks. *Source:* adapted from Taneja *et al*. 2013.

Application of BOSS and Results

The sMAP data repository allowed us to report on whole building energy consumption; the external visualization/analysis in the top layer accesses the data historian in the middle layer, which contained data from the sensors in the bottom layer. The average electrical demand of Sutardja Dai Hall in Academic Year 2011-2012 (July-June) was approximately 894 kW when the building used the steam-driven absorption chiller and 964 kW when the building used the electricity-driven centrifugal chiller. The office portion of the building uses about a quarter of the whole building energy; (the nanofabrication laboratory is quite energy-intensive). The figure below shows the whole building load from the two main substations, MSA and MSB, beginning with the first sMAP feeds in May 2011 through December 2012. Over this time period, many factors affected the energy consumption: which chiller[4] was running, the gradual installation and use of tools in the nanofabrication lab, the addition of laboratories and other rooms to the first floor, and energy efficiency measures, such as the addition of Variable Frequency Drives (VFDs) to the chilled and condenser water pumps, and a dynamic ventilation regime (minimum ventilation rates dynamically changed based on economizer and assumed occupancy to maintain 15 cfm of outside air per person). In addition, daily factors, such as outdoor air and solar loads and occupancy played a role.

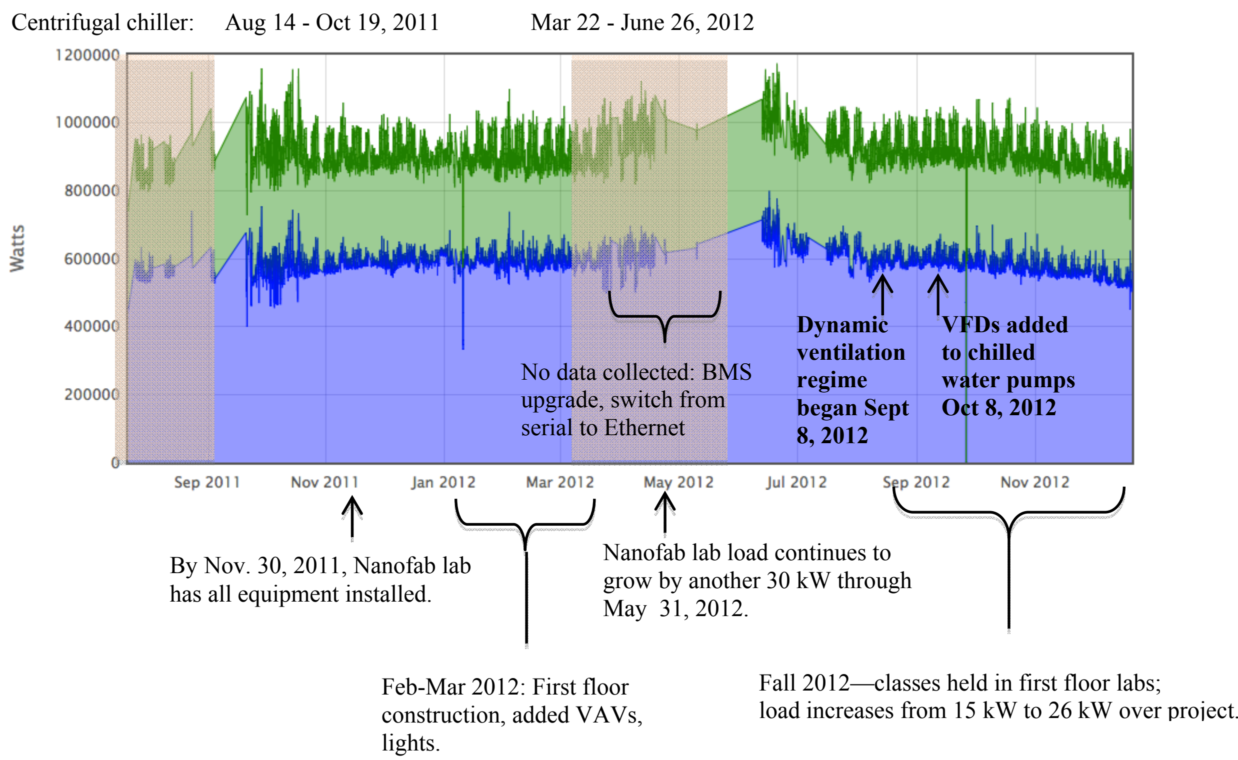

Figure 3. Whole building electrical load of Sutardja Dai Hall from May 22, 2011 to Dec 31, 2012 (MSA upper/green, MSB lower/blue) *Source*: Peffer 2012.

[4] The absorption chiller broke down in mid-August 2011, and thus the centrifugal chiller ran in August-October 2011, but we discovered this chiller was short-cycling. As soon as the absorption chiller was fixed, the building was switched to this chiller while the problem with the centrifugal chiller was addressed.

The BOSS system provided an accessible research test framework as well as improving the day to day operation of the building. The accessibility of the data using sMAP was useful in diagnosing issues. Graduate students found a stuck damper and faulty cooling valve, and helped diagnose the short cycling problem of the centrifugal chiller. In looking at the data, we determined many energy efficiency measures to implement. For example, the building was originally operating under a single 70°F setpoint (heating at temperatures below 70°F and cooling above 70°F). The existing BAS did not provide a means of developing a deadband, so in July 2012, we implemented a deadband (heating below 70°F and cooling above 74°F) using JSON scripts that called BACnet commands. Occupant surveys had indicated the building was too cool, so we also reduced the minimum ventilation rate by 30% and increased the Supply Air Temperature to 58°F. See Figure 4.

```
["SDH.S4-09:HEAT.COOL", 0],
["SDH.S4-09:CTL STPT", 74]

['SDH.S4-09:CTL FLOW MIN', 135]
```

For each of the 130 VAVs, the mode was set to cool (0), then a new control setpoint was sent (e.g., 74°F). (If the order was reversed, the system might heat to the setpoint instead.) The minimum ventilation rate (e.g., 135 cfm) was also adjusted in a similar way.

Figure 4: Scripts to control the building's BAS.

A secure website was developed that allowed researchers to upload tests and request approval from the facilities manager (Figure 5).

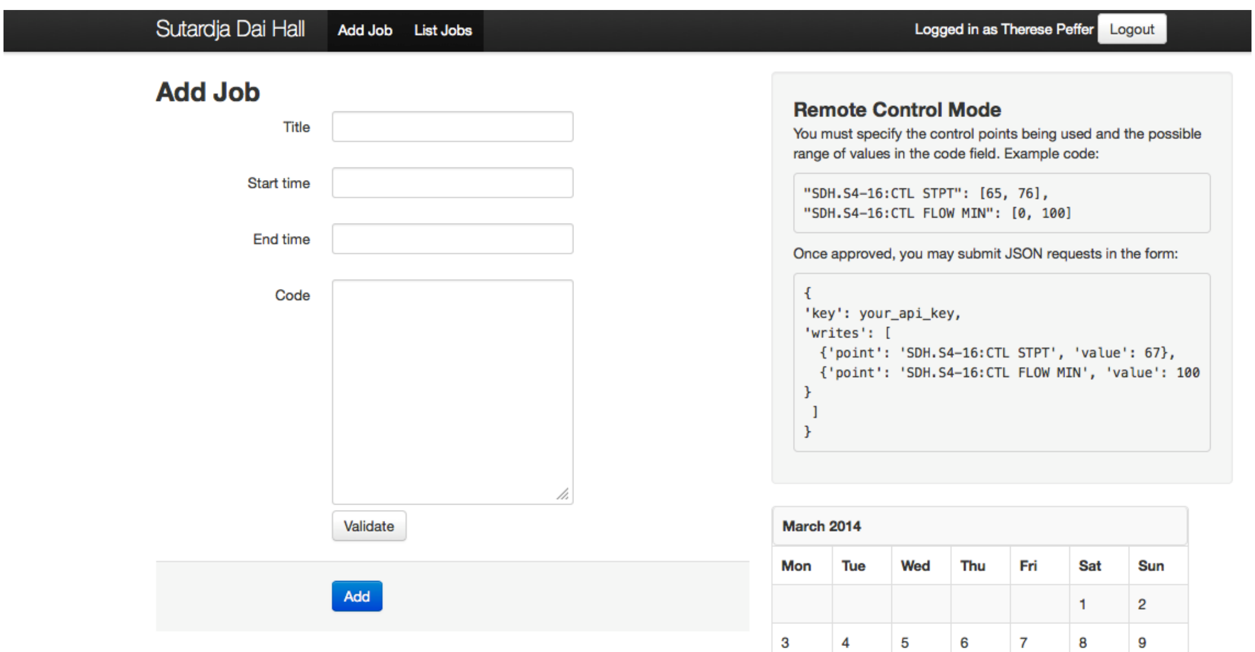

Figure 5. Website for uploading HVAC or lighting control tests.

Graduate student researchers developed and tested several applications using BOSS. Krioukov developed a browser-based lighting control interface (application/user interface at the top layer) that allowed occupants to easily control the lighting (bottom layer) to the level desired without having to use the cryptic wall switches. Users could also see real time lighting energy use on their floor. This lighting application saved 50% energy overall, mostly at night (Krioukov *et al*. 2011). He also developed a web-based user interface where occupants could request a blast of warm or cool air from the closest VAV supply. This app was very popular and helpful to improve comfort especially during demand response events. This application has turned into a commercial product, Comfy (http://buildingrobotics.com/).

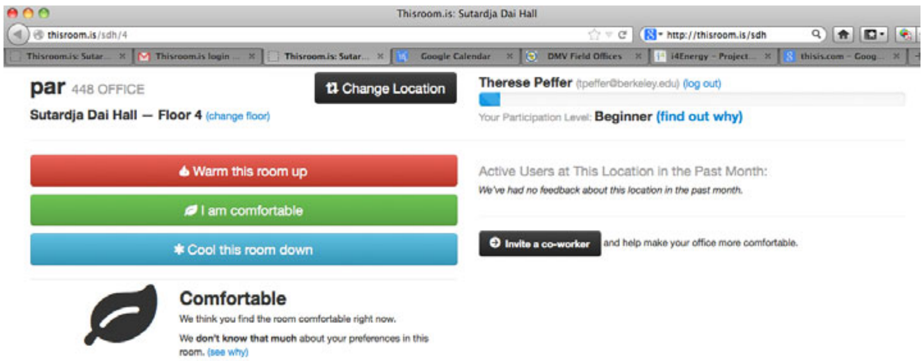

Figure 6. Web-based occupant thermal comfort control interface.

Taneja implemented a demand-controlled ventilation (DCV) application (top layer of BOSS) for several conference rooms in Spring 2012. He added carbon dioxide sensors to the sMAP TimeSeries database via a wireless 6lowPAN network (bottom layer). He also added the data from the building calendar to the database for scheduled meetings in the conference rooms; this also was a simple input at the bottom sensor layer. He wrote and implemented an application that controlled the ventilation rate of each conference room to preemptively blast air into the room before meetings and also during meetings based on carbon dioxide concentrations. The DCV application was able to reduce air quality threshold violations by over 95% and concurrently reduce energy consumption by over 80% (Taneja *et al*. 2013).

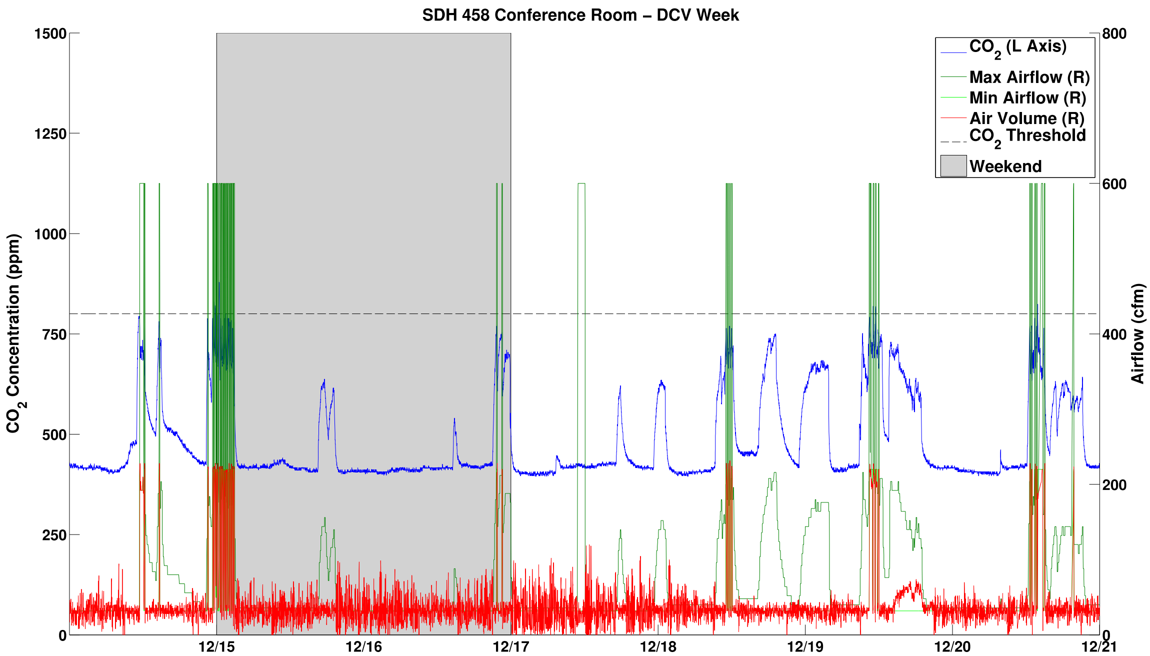

Figure 7. Operation of a conference room ventilation system with a demand-controlled ventilation strategy. *Source*: Taneja, 2013.

In September 2012, Krioukov implemented a dynamic minimum ventilation application over the entire office portion of the building using BOSS. This application reduced the minimum ventilation rate by 70-85% for short periods by estimating occupancy by time of day and calculating actual outdoor air intake from the economizer via the sMAP TimeSeries database to provide the required fresh air per person (per ASHRAE standards). He also used carbon dioxide sensors to ensure that the ASHRAE calculated minimum ventilation complied with acceptable carbon dioxide values.

Although we have not conducted a detailed analysis of the annual energy savings from the energy efficiency measures, we have calculated some approximate numbers. The load on the air handling units supplying the office portion of the building dropped by 20 kW from mid-2011 to end of 2012, mostly due to the dynamic minimum ventilation scheme. The load on the absorption chiller and associated pumps decreased by about 30 kW, probably primarily due to the VFDs installed on the pumps, but some effect from the implementation of the temperature deadband (i.e., control points at 70-74F versus a single control point of 70F), increase of the supply air temperature to 58F from 56F, and the reduced ventilation rate. This amounts to approximately $44,000 savings annually.

The BOSS system was instrumental in conducting Demand Response tests for a Department of Energy funded research project (Auslander *et al*. 2013, Peffer *et al*. 2012). Through the top layer, we implemented several demand response controls: raised the cooling setpoint, reduced minimum ventilation rate, raised the supply air temperature, and dimmed and turned off lighting throughout the office portion of the building. sMAP allowed us to trouble-shoot the control strategies in real-time. For example, the lighting commands often did not "go through" the first time and had to be resubmitted. We were able to catch this immediately and correct. An analysis of the zone temperatures indicated that many zones did not drift very much during the DR events; we iteratively reduced the minimum ventilation rate with each successive test. We also were able to detect zones on the 7th floor that grew warm rather quickly, and adapted the control strategy accordingly.

We achieved 14-25% average peak load reduction during our DR events using the absorption chiller, which uses steam for cooling and thus consumes far less electricity than the centrifugal chiller; we have estimated that the savings would have been higher using the centrifugal chiller. sMAP allowed us to evaluate the DR events by components, shown in Figure 8 below.

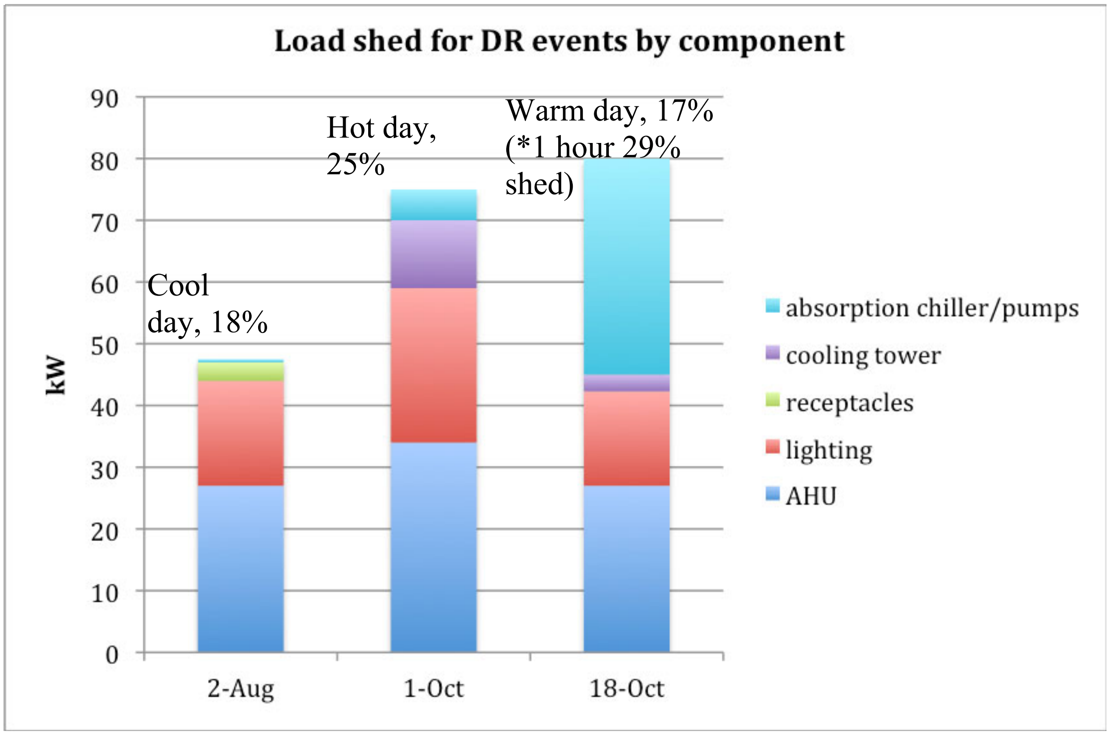

Figure 8. Summary of load shed for DR event days. *Source*: Auslander et al 2013.

Discussion

The simple layered architecture of BOSS provided the modularity needed to support a great deal of innovation and ease of testing by the researchers. In fact, when we wanted to implement a reduced minimum ventilation rate permanently, the facilities manager asked if we could implement it through BOSS. Otherwise he would have to manually enter hundreds of numbers (e.g., one for each VAV), each in a different window, through his BAS interface. This architecture does require a thorough understanding how a particular BAS works. For example, when we implemented the temperature deadband through a software application, we quickly learned that the command to change the heat/cool mode for each VAV must come before the desired setpoint (e.g., COOL, to 74°F). If these commands were sent in the reverse order, the system would heat or cool to the setpoint based on the current temperature in the zone. On one occasion, researchers not understanding this detail heated the building to 78F in the summer instead of allowing the temperature to drift to and cooling above 78F.

As a software solution, an open architecture BAS is not a zero cost solution—there are development and maintenance costs. But it does represent a low-cost means of experimentation as well as spurring innovation. Arguably, brilliant graduate students are cheaper than industry-paid PhDs, so our personnel investment was fairly low compared to the years of expected energy savings. We continue to develop Sutardja Dai Hall as a living laboratory; BOSS has become the foundation for future researchers to experiment towards improved performance.

While the prototype described in this paper was implemented on a campus building, we expect the platform would succeed in commercial and/or residential settings as well. In fact, a 2013-2014 DOE funded project has three teams (UC Berkeley, Carnegie Mellon University, and Virginia Tech) developing open-source, open architecture Building Automation Systems for small-to-medium sized commercial buildings (more information at openBASworkshop.org). The UC Berkeley solution builds upon BOSS. Virginia Tech uses the Pacific Northwest National Laboratory-developed Volttron platform (built on sMAP) and CMU is expanding Sensor Andrew work.

In a vertically integrated system, each company uses its own sensors and actuators, data management, and applications—usually proprietary. In horizontal architecture, a company might still have a proprietary application or sensor, but it would be able to be easily added to the system and interact with components from other companies. This distinguishes open-source software (publicly available, and thus non-proprietary) from open architecture software.

The Internet of Things has direct application for the current state of building controls. Third-party developers are essential for the innovation and specialization in the applications needed to operate the diverse equipment in buildings and respond to constantly changing requirements. These developers can easily add their own tools to the top layer, which could include specialized user interfaces, new control algorithms, and fault detection algorithms. Third parties can also add new devices for sensing and actuating to the bottom physical layer, enabled by the hardware presentation layer, sMAP.

We anticipate a reluctance of vendors to embrace this architecture due to a perception of losing market share. However, the more–forward looking companies are the ones who will quickly adapt to the rapidly changing market. Already more and more devices have software components, and many companies are releasing the Application Programming Interface (API) to allow third parties to interface with the device. We expect a more open software-architecture would lead to more standards in the way building automation systems are designed (e.g., a simple way of developing a temperature deadband or integrating demand response etc).

Conclusion

The layered and open software architecture of BOSS allowed researchers to easily and nimbly develop and test control applications and add sensors/data without disturbing the existing building automation system. Not only were we able to diagnose problems with HVAC equipment, but also improve day-to-day energy performance as well as reduce peak periods. We feel this approach leads to greater innovation in building controls at low cost compared to traditional proprietary stovepipe or vertically integrated control systems. A horizontal architecture allows multiple third-party vendors to provide applications—specialized user interfaces, optimization, fault detection—at the top layer or add devices—sensors, calendar data, hardware—at the bottom layer. Like the Internet, an open software-architecture fosters interoperability and can create energy-efficient software-defined buildings.

Acknowledgment

The authors drew extensively from the work of UC Berkeley EECS Chair and Professor David Culler and his brilliant graduate students in the LoCal and Software Defined Buildings research groups, namely Stephen Dawson-Haggerty, Andrew Krioukov, Jorge Ortiz, and Jay Taneja, as well as others. This work was supported by the National Science Foundation and the Department of Energy.

References

Auslander, David, Culler, David, Wright, Paul, Lu, Yan, Piette, Mary Ann, Peffer, Therese. 2013. Distributed Intelligent Automated Demand Response Final Report, Department of Energy, DE-EE0003847, from http://i4energy.org/downloads/projects/sutardja-dai/DIADRFinalReport.pdf.

Blumstein, Carl. 2011. Energy Efficiency, Information Technology, and the Electricity System. *Proc. 2011 Conference on Energy Efficiency*, European Council for an Energy-Efficient Economy, Paris, France.

Dawson-Haggerty, Stephen, Jiang, Xiaofan, Tolle, Gilman, Ortiz, Jorge, and Culler, David. 2010. sMAP — a Simple Measurement and Actuation Profile for Physical Information. *in* Proceedings of the Eighth ACM Conference on Embedded Networked Sensor Systems (SenSys '10), November 2010.

Dawson-Haggerty, Stephen, Krioukov, Andrew, Taneja, Jay, Karandikar, Sagar, Fierro, Gabe, Kitaev, Nikita, and Culler, David. 2013. BOSS: Building Operating System Services. In Proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI '13).

Krioukov, Andrew, Stephen Dawson-Haggerty, Linda Lee, Omar Rehmane, and David Culler. 2011. A Living Laboratory Study in Personalized Automated Lighting Controls. In *BuildSys'11*. Seattle, WA: ACM.

Krioukov, Andrew, Fierro, Gabe, Kitay, Nikita, Culler, David. 2012. Building Application Stack. In *4th ACM Workshop On Embedded Sensing Systems For Energy-Efficiency In Buildings*, Toronto, Canada, Nov 2012.

Peffer, T. E., Auslander, D. M., Caramagno, D., Culler, D., Jones, T., Krioukov, A., et al. 2012. Deep Demand Response: The Case Study of the CITRIS Building at the University of California-Berkeley. Proceedings of the 2012 ACEEE Summer Study on Energy Efficiency in Buildings.

Taneja, Jay, Krioukov, Andrew, Dawson-Haggerty, Stephen, and Culler, David. 2013. Enabling Advanced Environmental Conditioning with a Building Application Stack. In Proceedings of the 4th International Green Computing Conference (IGCC '13).