Title
Enabling Advanced Environmental Conditioning with a Building Application Stack.

Permalink
https://escholarship.org/uc/item/5368q9mp

Authors
Taneja, Jay
Krioukov, Andrew
Dawson-Haggerty, Stephen
et al.

Publication Date
2013-06-27

Peer reviewed
Enabling Advanced Environmental Conditioning with a Building Application Stack

Jay Taneja, Andrew Krioukov, Stephen Dawson-Haggerty, and David Culler
Computer Science Division
University of California, Berkeley
Berkeley, California 94720
Email: {taneja,krioukov,stevedh,culler}@cs.berkeley.edu

Abstract—There is enormous potential for building-focused applications to improve operation and sustainability, both for classical uses like modeling or fault detection as well as innovative ones like occupant-driven control or grid-aware energy management. We show that a building application stack – that addresses shortcomings of existing antiquated architectures by democratizing sensor data, constructing a framework for reliable and fault-tolerant operation of concurrent applications, and establishing an application programming interface to promote portability throughout the building stock – enables development of advanced applications. We observe the growing importance of applications that integrate sensors and actuators from the building infrastructure with those from “add-on” networks, and show how this design pattern is further empowered by the architecture. To prove the efficacy of the approach, we implement two advanced environmental conditioning applications on a large, commercial building that was not designed for either of them: a demand-controlled ventilation (DCV) system for balancing air quality considerations and energy use in conference and classroom settings and a demand-controlled filtration (DCF) system for conserving recirculating fan energy in an intermittently occupied cleanroom setting. The DCV application is able to reduce air quality threshold violations by over 95% and concurrently reduce ventilation energy consumption by over 80%, while the DCF application can reduce recirculating fan power consumption by half with no repercussions on air quality when the room is occupied. Further, the portability of these applications highlights the potential of the architecture to enable widespread and rapid application development throughout the building stock.

I. INTRODUCTION

Buildings, where we spend over 90% of our time [1] and 72% of our electricity in the U.S. [2], are a prime opportunity for information technology to improve sustainability. However, the building sector is slow to innovate, with design lifetimes counted in the decades and limited budgets for improvements. Though changes in building codes exert some pressure on new buildings to incorporate technologies that improve comfort and energy efficiency, little is generally done to improve existing buildings and their control systems. The recent emergence of reliable wireless sensor networks as the next tier of the Internet can enable advanced sensing and the associated control to augment existing building control systems. Potentially, with an increased ability to monitor and control, advances in building environmental conditioning can permeate buildings that otherwise must wait until their systems can be retrofitted or, in the worst case, until the building is torn down and replaced.

However, the challenges in integrating additional sensors to building control systems today are myriad: the systems deployed are a cornucopia of aged technologies speaking a wide array of protocols; the control systems that govern building operation are vertically-integrated, barely programmable, and not extensible; and the custom design of buildings and building control systems by a range of different parties results in a potpourri of naming schemes. Recent efforts aim to address these shortcomings. First, in work to democratize the physical data streams generated by buildings and other sources by providing a unified, RESTful interface [3], the architecture for accessing physical data in future control systems is becoming clearer. Second, a proposal for “building operating system services” (BOSS) [4] details a programmable platform for constructing fault-tolerant applications on top of the physical building infrastructure. Last, a “building application stack” (BAS) [5] provides an application programming interface and runtime for applications that are portable among buildings, enabling a “write once, run anywhere” paradigm for building application developers. Together, these advances enable the ability to deploy applications over the physical space in buildings, something previously largely intractable.

In this paper, we study the types of applications that can be created using a building application stack. We classify the emerging applications space, recognizing that nearly all applications incorporate data from a combination of sensors within traditional building management systems as well as “add-on” networks, often delivered over a wireless sensor network or an Internet data feed. These networks are generally deployed exclusively to support one application, and each application has recreated the functionality needed to fuse the new data source. We show how the BOSS/BAS-based architecture can encourage sharing of both hardware (sensor deployments) and software (application code), greatly enhancing the building application development process. To prove the merit of the building application stack approach, we present two advanced environmental conditioning applications that extend the application stack via both software and hardware additions to the existing building control fabric: demand-controlled ventilation (DCV) and demand-controlled filtration (DCF). The DCV application employs a CO₂ sensor to proportionally provision outside air ventilation to rooms that are intermittently densely occupied (e.g., conference rooms or classrooms). The DCF application uses motion sensor data in a cleanroom setting to detect periods of inactivity when recirculation fans can be
turned down or off to save energy.

Though both of these applications are well-studied and deployed in a number of buildings, they generally appear only in settings where the building and its control systems were designed with that specific purpose in mind; the “application” was built into an integrated building management system from a third-party vendor, and its input and output data remain in a stovepipe architecture of building control. In these cases, modifying the application is cumbersome, and extending it to other spaces is impossible. In the BOSS/BAS architecture, a simple additional sensor tier coupled with an easily-programmed control system enables the substantial benefits of the applications, both in environmental conditioning as well as energy efficiency, without requiring costly integration into the existing building management system.

II. EXISTING BUILDING SYSTEMS

Most modern commercial buildings contain extensive infrastructure systems to ensure occupant health, safety, and comfort. This includes providing heating, ventilation, and air conditioning (HVAC), as well as, lighting, security, and fire safety services. These systems are often networked and can be centrally managed through operator interfaces, but are frequently provided by different vendors and have little interoperability or extensibility beyond the scope of the original system design.

A typical commercial building HVAC system is shown in Figure 1. Fresh air is brought in from the outside to satisfy health requirements and is mixed with return air from the building. This mixed air is cooled by passing over cold water coils and blown through ducts throughout the building. In each thermal zone, typically encompassing 1 to 3 private offices or 4 to 8 cubicles, the air passes through a variable air volume (VAV) box that dynamically controls airflow and may use a reheat coil to meet temperature and ventilation requirements. From there, air enters the occupied space through diffusers. After circulating, air is sucked back through a return air plenum where a portion is exhausted and the remaining portion is recirculated.

A typical HVAC system for a large office building contains thousands of sensors and actuators measuring air and water temperatures, airflow, humidity, and duct pressures throughout the building, supporting several underlying loops. Actuators range from simple on/off relays to variable speed fans and pumps, water valves, and dampers. A modern HVAC system with digital controls contains embedded controllers, also known as programmable logic controllers (PLCs), throughout the building that are used to collect data from these sensors and run the logic to actuate active components.

The logic running on embedded controllers is custom-written for each building. Historically these devices were programmed with ladder logic [6]; today, a range of graphical and text-based programming languages are used. For example, Siemens systems use the Powers Process Control Language [7], a BASIC-like interpreted language, while Automated Logic systems use a graphical tool that consists of “microblocks,” simple functions and logical blocks that can be wired together [8]. All of these systems lack meaningful high-level abstractions, easy communication with data sources outside the building, and an environment that allows rapid upgrades. Instead, today’s building applications are hard-coded in low-level programming languages, requiring an engineer to visit the building for even minor changes.

Most HVAC vendors follow a stovepipe design with proprietary sensors, actuators, controllers, programming languages, and management software, making upgradability and interoperability a major challenge. Several standards have been established to address these problems. BACnet [9], standardized in 1995, is the most widely adopted controls standard. It defines a common protocol for communicating with controllers, or in some cases with gateways that translate internal proprietary protocols. BACnet specifies physical, data link, network, and application layers. At the application layer, BACnet exposes a set of devices and points each with certain properties that can be read or written. For example, a common action is a read of the “PRESENT_VALUE” property on a given point. Unfortunately, there is no standardization of point names or values: a variable air volume box can be represented by tens of points with unrelated names and may have widely differing functionality from one vendor to another. BACnet also does not specify a standard way to reprogram building controllers; instead, writes to BACnet points may override the inputs or outputs of the programmed control logic.

Overall, this legacy architecture poses a number of challenges: ease of programmability, extensibility to support new applications and new hardware or online data sources, and portability of applications. Recent work on a building operating system [4] and a building application programming interface [5] begin to address these issues.

III. BUILDING APPLICATIONS

To enable building applications, we build on previous work introducing a building operating system and API. BAS [5] is an application programming interface and runtime that enables writing portable code by providing methods to explicitly and implicitly handle differences in building designs. A key insight of BAS is the use of fuzzy, relativistic queries to allow authors to express their high-level intent in a way that is inherently portable, e.g., “turn off the lights for top floor cubicles near windows,” as well as supporting programmatic exploration of a building’s specific components, allowing applications to explicitly handle building differences. Thus application
developers can alternate between macro- and micro-level views of the building (e.g., “lights on the top floor” vs. “Light Relay 1023”) to express both general intentions and specific actions. This frees developers from understanding the often idiosyncratic design of each building, allowing development to focus on the interaction between physical spaces and occupants rather than the specifics of equipment.

BOSS [4] proposes a new architecture for building control systems that, in addition to operating the machinery, provides for robust, portable application development and supports many simultaneously running applications. BOSS consists of a collection of services comprising a distributed operating system that solves several problems that have hindered earlier systems from scaling. Faults are addressed by implementing a transactional system for updating the state of multiple physical devices and reasoning about what will happen during a failure. Historical and real-time data are treated uniformly in a time series service that allows applications to make identical use of both past and present data in a scalable way.

The combination of these software layers allows multiple building applications to be easily deployed on existing buildings and enables building control systems to be dynamically supplemented with new data sources, both physical and virtual. Developers can use these systems to rapidly design, install, test, and deploy building applications.

Many building applications have been developed prior to the emergence of this architecture; Table I categorizes applications from the literature by function. Most deployed building applications combine sensing and actuation capabilities of the building management system with those of “add-on” sensors. These add-on sensors augment existing building sensing infrastructure by either gathering data streams from the Internet or adding additional sensing hardware and often a wireless network for retrieving data. Generally, hardware is used only by the particular application for which it was deployed, and data generated by these applications remain isolated from other applications. By recognizing that there is significant overlap in the types of sensors used by the array of applications, both from the BMS as well as in add-on networks, there is enormous potential to reuse hardware for multiple applications within the same building. Further, development of these applications often involved custom handling of varied data streams from building and add-on sensors. This represents significant redundant software effort. Instead, the BOSS/BAS architecture promotes reuse of hardware, enabling application development to become progressively more time- and resource-efficient and letting developers focus on the application rather than the infrastructure. Further, this architecture provides additional security and reliability benefits to enable multiple applications to coexist gracefully on a shared infrastructure. As the set of applications matures, common services – such as occupancy detection or localization – can easily be incorporated into the architecture and made available for rapid development of new applications.

### IV. Ventilation Applications

We present two concrete applications that make use of a building application stack: demand-controlled ventilation (DCV) and demand-controlled filtration (DCF). We demonstrate each application in our test building, a 7-story, 140,000 ft² facility on the UC Berkeley campus that has two “buildings-within-a-building”: the first is primarily office, classroom, and cubicle space for undergraduate and graduate students, staff, and faculty, and the second is a multi-floor industrial-grade chip fabrication laboratory for research. Since these two “buildings” have significantly different conditioning requirements, the test building allows for a wide range of applications to run on the same physical infrastructure. Though these two applications employ different sensors and operate in very different settings, they share the same basic architecture, each bringing together the building control system and an add-on wireless sensor network to implement a leading-edge environmental control application in a space that was not designed for it, as seen in Figure 2. Both applications leverage a wireless sensor network testbed that is deployed throughout the building; this network provides a communication backhaul for sensor data to reach the data historian in the BOSS/BAS architecture. Though we acknowledge that not every building

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
<th>Building Sensors Used</th>
<th>Add-on Sensors Used</th>
<th>Actuators Used</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individual Energy Accounting</td>
<td>Provide personal feedback of energy use</td>
<td>Power, Light</td>
<td>Plug, Proximity</td>
<td>Relay</td>
<td>[10][11][12][13][14]</td>
</tr>
<tr>
<td>HVAC Control</td>
<td>Condition indoor environment based on occupancy</td>
<td>Power, Temp, Network</td>
<td>Door, Motion, Camera, CO₂</td>
<td>VAV</td>
<td>[15][16][17][18][19][20][21]</td>
</tr>
<tr>
<td>HVAC Control</td>
<td>Condition indoor environment based on occupancy</td>
<td>Power, Light, Network</td>
<td>Door, Motion, Camera</td>
<td>Relay</td>
<td>[22][23][24][25][26][27]</td>
</tr>
<tr>
<td>Personalized Control</td>
<td>Unleash the indoor environment based on occupancy</td>
<td>Power, Light, Network</td>
<td>Occupant Input</td>
<td>VAV</td>
<td>[30][31][32]</td>
</tr>
<tr>
<td>Shared Room Management</td>
<td>Schedule use of common space</td>
<td>Power</td>
<td>VAD, temp, Light</td>
<td>N/A</td>
<td>[33][34]</td>
</tr>
<tr>
<td>Dashboarding</td>
<td>Provide visualization of resource use</td>
<td>Power, Water</td>
<td>Weather</td>
<td>N/A</td>
<td>[35][36]</td>
</tr>
<tr>
<td>Plug Load Management</td>
<td>Monitor/control plug-connected devices</td>
<td>N/A</td>
<td>Power, Water</td>
<td>Relay</td>
<td>[27][28][29][30][41][42]</td>
</tr>
<tr>
<td>Baseline/Forecasting/Modeling</td>
<td>Model building performance</td>
<td>Power, Light, Temp</td>
<td>Weather</td>
<td>HVAC</td>
<td>[43][44][45][46][47][48]</td>
</tr>
<tr>
<td>Daylighting</td>
<td>Daylight-aware lighting and shading</td>
<td>Light</td>
<td>Light, Weather</td>
<td>Relay</td>
<td>[49][50]</td>
</tr>
<tr>
<td>Water Management</td>
<td>Monitor/control water use</td>
<td>Water</td>
<td>Water Flow, Temp</td>
<td>N/A</td>
<td>[51]</td>
</tr>
<tr>
<td>Fault Detection</td>
<td>Find anomalies in building performance</td>
<td>Power, Light, Temp</td>
<td>Power</td>
<td>N/A</td>
<td>[52][53][54]</td>
</tr>
<tr>
<td>Building System HVAC Control/Optimization</td>
<td>Model and manage HVAC loop</td>
<td>Power, Temp</td>
<td>Weather, Temp</td>
<td>HVAC</td>
<td>[55][56][57][58][59]</td>
</tr>
<tr>
<td>Localization</td>
<td>Identify location of occupants</td>
<td>N/A</td>
<td>Magnetic, Range, Radio</td>
<td>N/A</td>
<td>[60][61][62][63]</td>
</tr>
<tr>
<td>Grid-Aware</td>
<td>Modulate power consumption based on electricity data</td>
<td>Power, Temp</td>
<td>Weather, Network, Price, Power</td>
<td>HVAC, Relay</td>
<td>[64][65]</td>
</tr>
</tbody>
</table>
has this capability, we believe that the particular choice of physical and link layer protocols used for sensor data is not critical to the applications.

A. Demand-Controlled Ventilation Application

The challenge for a modern building operator in selecting ventilation rates and schedules is to achieve energy-efficient operation while ensuring that building denizens receive ample fresh air. Traditionally, these decisions have been made at the commissioning stage, with airflow levels selected to ensure adequate ventilation such that air quality and human bioeffluent levels remain at a comfortable level during full occupancy. This airflow level is called the equilibrium level, and its selection is governed by a variety of factors, including maximum occupancy, usage pattern, air volume, and adherence to state and national building standards.

In commercial buildings in California, the relevant standards governing indoor ventilation levels are the California Code of Regulations (CCR) Title 24 [66] at the state level and the American Society for Heating, Refrigerating, and Air-Conditioning Engineers (ASHRAE) standard 62.1 [67] at the federal level. These standards dictate that mechanically-ventilated spaces must always receive at least 0.15 cubic feet per minute (cfm) for each square foot of area ventilated. Further, ventilation should be delivered based on occupancy, at a rate of 15 cfm per occupant of the space. In the absence of occupancy sensors, the default is to ventilate according to maximum occupancy. To detect whether the space is occupied, the codes define two possible means: occupant sensors, which generally detect motion and provide only a binary signal, or CO₂ sensors, which reflect the concentration of CO₂ in the indoor air. Use of these sensors to determine ventilation rates is called demand-controlled ventilation (DCV).

Though DCV has conceptually been around for decades, it is not widely in use. Now, emerging technologies, changes in building standards, and more awareness of occupant comfort are increasing deployment of DCV systems. Sensors and the information technology needed to retrieve the data they produce are becoming cost-effective and ubiquitous. Recognizing this and the energy savings potential from reducing excess indoor ventilation, the Title 24 standard has been modified. The next iteration of the standard requires that DCV be used in dense settings – i.e., any room larger than 150 square feet with 40 square feet or less per expected occupant. Indoor air quality standards also dictate that indoor CO₂ concentration should remain below 600 parts per million (ppm) beyond outside air concentration (in the absence of a sensor measurement, this is assumed to be 400 ppm). Figure 3, adapted from the ASHRAE 62.1 standard, encapsulates the challenge for building managers: despite dynamic conditions created by the movement of people and a changing environment, maintain a CO₂ concentration in all spaces between 900 and 1175 ppm. Newer research indicates that even this target region may be too high; human decision making performance can show significant reductions at even 1000 ppm CO₂, as compared to 600 ppm [68]. Given this, for the duration of our study, we chose a maximum concentration threshold of 800 ppm CO₂. With the changes in building standards and the increasing focus of building operators on occupant comfort, DCV systems are now common in most new commercial buildings.

The application of DCV studied here is focused on existing buildings, where the sensors and associated controls for DCV are not initially installed. For these types of buildings, meeting the improved standards for air quality is not possible without deep modifications. Though this hardware and software could be installed by the third-party building management system vendor, we advocate an incremental approach, where a wireless sensor network is used to provide occupancy data to support the application. In deploying this network, we compare three ventilation control system configurations: first, a baseline system that reflects the state of the ventilation controls after the building commissioning process; second, an extreme efficiency system that aims to reduce ventilation by as much as safely possible; and third, a demand-controlled ventilation system that modulates ventilation based on occupancy sensor data. To evaluate the performance of each control system, we deployed 8 wireless nodes with CO₂ and passive infrared (PIR) motion sensors throughout a single floor of office space in our testbed building on campus, as shown in Figure 4. The floor is primarily an open office “cubicle” area, which is surrounded by enclosed offices and conference rooms, and is roughly 10,000 ft² total. To capture the variations in behavior among these spaces, six of the sensors were deployed in the open office space, one in an office, and one in a conference room. The sensors were all positioned near room air return vents and at a height of between three and six feet, according to the guidelines laid out in previous work [69]. The wireless sensors use a 6LoWPAN-based IPv6 networking stack with the building network acting as edge routers, as in previous deployments [40], [41]: data were reported from each sensor every 15 seconds via sMAP to the BOSS/BAS historian.

Baseline Controller. For the baseline study, we use the conference room, whose ventilation is managed by a variable-air-volume (VAV) unit configured with minimum and maxi-

---

1K30 CO₂ Engine - http://www.senseair.se/products/oem-modules/k30/
mum airflow settings and a temperature setpoint. The room is 214 ft², an estimated maximum occupancy of 12 people, and a default minimum airflow setting of 220 cfm. Given that standards advise that each room occupant should receive 15 cfm, the baseline airflow settings expect that 15 people are in the room continuously, even though the room is likely only occupied for a handful of hours per day, primarily on week- days. This anecdote represents the all-too-common problem of building ventilation controls: overprovisioning of minimum airflow settings causes chronic overventilation. Further, since supply air is generally cooler than inside air, the end result is a room that is constantly supplied with cold or reheated air. Even though temperature is supposed to be maintained in the room by a PID controller with a temperature deadband, the controller seldom needs to cool the room beyond what the minimum airflow already does. Thus, the airflow in the room generally equals the minimum airflow setting, and in fact hot water is often used to reheat the air, thus needlessly wasting both fan energy and gas for heating.

Figure 5 shows the interaction of the ventilation system of the conference room, the minimum and maximum airflow settings, and the CO₂ concentration in the room over a week. The average airflow over the week is 222.2 cfm. Despite this “always occupied” configuration, the CO₂ concentration in the room still crosses the maximum threshold multiple times within this week, for a total of just over 6 hours spent above the threshold (3.6% of the total time). Further, it is estimated that the room is occupied less than 10% of the hours of the week. This presents an opportunity for substantial energy savings by only operating the ventilation system during occupied hours.

**Extreme Efficiency Controller.** The second ventilation controller compared aims to reduce the ventilation by as much as possible while still maintaining sufficient air quality. This effort, which combines a precomputed time-of-day-based occupancy model, outside air damper control sequence, and significant reductions in default airflow levels, represents an extreme efficiency ventilation strategy. This controller uses aggressive airflow cutbacks of up to 70% of the default value.

Figure 6 shows the CO₂ levels collected throughout our deployment for three typical weekdays using the extreme efficiency controller. In each area, the CO₂ concentration remains near the concentration of outside air (generally, 400 ppm) during the nighttime hours. As occupants arrive late in the morning, the concentration in all of the areas begins to increase. During the occupied hours, the conference room has intermittent spikes due to meetings, while the other spaces reflect some but minimal occupancy extending into the evening, in line with typical graduate student schedules.

The conference room approaches the maximum threshold of 800 ppm, crossing it four times during the three days. Neither the enclosed office, which seldom has more than 1 or 2 occupants, nor the open office area, which is shared among tens of occupants but is far less dense, approaches the maximum threshold. In fact, over the month-long duration of our deployment in this area, the open office area never violated the 800 ppm maximum concentration. This is primarily because this area is not densely occupied, has far more baseline ventilation due to multiple VAV units with overlapping zones, and comprises a much larger volume of air, all resulting in lower, generally acceptable CO₂ concentration.
Fig. 7. Operation of a conference room ventilation system with an extreme efficiency control strategy.

The concentration in the enclosed areas, the office and conference room, generally remains below the threshold, but rises quickly during meetings of multiple people. The breath of occupants permeates the space, generated at roughly 0.5 L/min and with a concentration of near 5% CO₂ (50000 ppm) for each occupant. As this air mixes with existing room air, without any response from the ventilation system, the room air quality quickly degrades. The slow diffusion of heat from the occupants, relative to the much faster diffusion of CO₂, does not elicit a fast response from the temperature controller; even a large meeting in a small enclosed space can take tens of minutes to generate enough heat for the room to exit the temperature deadband and cause the control system to increase ventilation to cool the room. Generally, by this time, the CO₂ concentration in the room has violated the 800 ppm threshold.

An extreme efficiency approach appears to maintain acceptable CO₂ concentrations for lightly-occupied and mostly open spaces. However, enclosed spaces with highly variable occupancy, such as the conference room, may show unacceptable air quality levels under this type of configuration. Figure 7 shows a week of operation in the same conference room as the baseline controller, but instead running an extreme efficiency ventilation system. The average airflow for the room over the week is 79.8 cfm. Though the total airflow to the room, and with it the energy spent on ventilating the room, has significantly decreased (over 64%), we see even more violations of the maximum CO₂ threshold than we did in the baseline scenario. In fact, there is at least one violation every day, and the total time spent over the maximum threshold is nearly 11 hours (6.5% of the total time). Further, the room is not comfortable from a temperature perspective either. Each time the airflow deviates from a value below 100 cfm represents the temperature controller responding to a violation of the deadband, meaning that the temperature in the room is 74 degrees Fahrenheit or above; this happens multiple times throughout the week, likely whenever a large meeting is held.

To enhance our understanding of the air quality implications of meetings, we leverage the department room reservation system, deployed independently of this study. In this system, occupants use bConnected, a service in the campus suite that provides a Google Calendar for each of the conference rooms. By fetching calendar entries, it is possible to discern scheduled meetings – when people were expected to be in the room. However, the room is not necessarily occupied when a meeting is scheduled; further, the room may be occupied for unscheduled meetings. Also, meeting size cannot be determined from these data. Looking over a two-week period, we sought to uncover the frequency of unscheduled meetings. During this period, the ventilation system used the extreme efficiency method, and a meeting was determined to be taking place if the threshold CO₂ concentration was surpassed. Over the two-week period, there were 28 total meetings, or violations of the CO₂ threshold. Of these, 15 were during scheduled meeting times (out of 26 total scheduled meeting times in the period), and 13 were unscheduled meetings. Half of scheduled meetings did not have enough occupancy to create high CO₂ concentrations even with reduced airflow and a third of all high CO₂ concentration events were during unscheduled meetings. Given this, we believe that these calendar entries can provide useful information about when people are likely going to be in the room, but do not cover all gatherings in the room and thus are not sufficient for providing software-only ventilation control that meets our CO₂ concentration goals.

Demand-Controlled Ventilation Controller. Enclosed, intermittently occupied spaces, where air quality may be below the acceptable standard, are the target setting for the DCV controller. Variable usage patterns can still be exploited for significant energy savings, but ample ventilation can be provided when occupants are present. In our campus building, conference rooms, classrooms, and large presentation rooms typically have these occupancy patterns.

For implementing a DCV system, indoor air quality standards permit using two different sensor technologies: binary occupancy sensors (most commonly, motion sensors) and CO₂ sensors. Since our sensor platform has both, we can compare their performance in the same room; Figure 8 shows motion triggers and the CO₂ concentration in the same conference room over a day. On initial observation, the strengths of each sensor are clear: the CO₂ sensor provides a continuous measure that may provide an estimate of room occupancy, while the motion sensor is able to immediately detect when any occupants enter and all occupants leave the room. However, the highly-discretized occupancy pattern provided by the motion sensor provides minimal guidance about actual air quality in the room; though it might be possible to infer this information from the frequency of motion, the sensor simply has very little potential to differentiate a small meeting that does not generate enough CO₂ to approach the maximum threshold from a large meeting that does. Additionally, the potential for a motion sensor to be obstructed may limit its utility. Given these reasons, we designed a demand-controlled ventilation system that employs CO₂ sensors.

Our system uses a moving average of CO₂ readings over the previous two minutes; since the CO₂ sensors take a reading every 15 seconds, this averages over enough samples to dampen the effect of outliers, but also allows the controller to be agile to somewhat fast changes in the concentration, such as at the beginning of a large meeting. To describe the controller rules, Table II introduces some terminology, as well as relevant values for the conference room under study.

Table III shows the rules used by the controller to set minimum and maximum airflow values under all conditions. Using the calendar entries, the controller ventilates during scheduled meetings by selecting a minimum airflow value that reflects the
Fig. 8. Response of a CO₂ sensor and motion sensor in the same room.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value for Conf. Room</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a_{\text{fmin}})</td>
<td>Min airflow (\text{Title 24)}: 0.15 * ft^2)</td>
<td>32.1 cfm</td>
</tr>
<tr>
<td>(a_{\text{fmax}})</td>
<td>Default minimum airflow</td>
<td>220 cfm</td>
</tr>
<tr>
<td>(CO_2)</td>
<td>Default maximum airflow</td>
<td>600 cfm</td>
</tr>
<tr>
<td>(CO_2_{\text{max}})</td>
<td>Baseline (CO_2) value</td>
<td>425 ppm</td>
</tr>
<tr>
<td>(CO_2_{\text{hyst}})</td>
<td>Maximum allowable (CO_2)</td>
<td>750 ppm</td>
</tr>
<tr>
<td>(\text{Threshold to reduce airflow})</td>
<td></td>
<td>700 ppm</td>
</tr>
</tbody>
</table>

**TABLE II.** TERMINOLOGY USED FOR DCV CONTROLLER. VALUES ARE PROVIDED FOR THE CONFERENCE ROOM USED IN THIS STUDY.

The concentration of \(CO_2\) in the room between a baseline value for the room \((CO_2_b)\) and a safe maximum \((CO_2_{\text{max}})\), which is slightly below the maximum threshold to ensure that the system can respond quickly enough to prevent violations of the maximum threshold. Additionally, the controller ventilates briefly before and after scheduled meetings to ensure fresh air for occupants when they arrive and in case the meeting runs over its allotted time. In non-meeting times, as long as the \(CO_2\) concentration is not approaching the maximum threshold, the minimum airflow remains at the absolute minimum \(a_{\text{fmin}}\), which is based only on square footage and assumes zero occupancy. During these times, the maximum is increased to reflect occupancy, though the deadband in the temperature control system generally dictates that air volume will match the minimum airflow setting. In unscheduled meeting times when the \(CO_2\) concentration does approach the maximum threshold, the system responds by providing full airflow \(a_{\text{fmax}}\) in order to prevent a violation. To reduce cycling between \(a_{\text{fmin}}\) and \(a_{\text{fmax}}\) around the threshold, a measure of hysteresis is added by not reducing airflow until the \(CO_2\) concentration falls below a level less than \(CO_2_{\text{max}}\), called \(CO_2_{\text{hyst}}\); in our system, the hysteresis level is 700 ppm.

We have implemented the DCV application using BAS; the pseudocode is included in Figure 9. To do this, we extended the architecture to include two new object tags: \#CO2 and \#CAL, representing \(CO_2\) sensors and Google calendar entries, respectively. These objects are associated with areas in the spacial domain so that the fuzzy query interface of BAS could be used to locate the relevant physical object and its associated data. Further, we have defined drivers for each of these objects, including methods to get current sensor and calendar data. Additionally, we have extended the VAV driver to include methods for retrieving default airflow settings. As the diversity of add-on networks increases, we expect that additional object tags and drivers will be created; as a result, building equipment graphs will become richer, supporting further applications using the same infrastructure. Use of the BOSS/BAS architecture enables the application code to be free of references to specific equipment in a particular building, and thus allows us as developers to focus on application requirements rather than interfaces with building equipment.

<table>
<thead>
<tr>
<th>Situation</th>
<th>Minimum Setting</th>
<th>Maximum Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>5m before to 5m after scheduled meeting</td>
<td>([a_{\text{fmin}}, a_{\text{fmax}}] \times CO_2) in ([CO_2_b, CO_2_{\text{max}}])</td>
<td>(a_{\text{fmax}})</td>
</tr>
<tr>
<td>Other times</td>
<td>(CO_2 &lt; CO_2_{\text{max}})</td>
<td>(a_{\text{fmin}})</td>
</tr>
<tr>
<td>Other times</td>
<td>(CO_2 \geq CO_2_{\text{max}})</td>
<td>(a_{\text{fmax}}) until (CO_2) falls below (CO_2_{\text{hyst}})</td>
</tr>
</tbody>
</table>

**TABLE III.** RULES USED FOR DCV CONTROLLER.

![Image](https://via.placeholder.com/150)

A week of performance data for the DCV controller is provided in Figure 10. For the grand majority of hours, the airflow closely mimics \(a_{\text{fmin}}\) as meetings are not being held and \(CO_2\) concentration does not approach the maximum threshold. During scheduled meetings with significant occupancy, the reactivity of the system maintains the \(CO_2\) concentration at a safe level. During unscheduled meetings, the system responds to maintain \(CO_2\) concentration near the maximum threshold. The system may benefit from lessening the hysteresis threshold to reduce the cycling of the VAV damper.

Table IV compares results of the three ventilation strategies. Mean ventilation power can be calculated by using a model of supply fan power derived from measurement data, as shown in Figure 11. At each airflow level, we calculate the power required to provide an incremental \(cfm\) of airflow; we use this ratio and the total airflow required by that room to calculate its instantaneous ventilation power.

**TABLE IV.** RESULTS OF A WEEK OF OPERATION OF THREE DIFFERENT VENTILATION STRATEGIES. COUNTS OF SCHEDULED MEETINGS ARE OBTAINED FROM THE DEPARTMENT CALENDAR; A MEASURE OF EVENTS OVER THE 800 ppm THRESHOLD IS ALSO PROVIDED.
while only using 15% and 44% of the power of those systems. Further, small changes to the DCV rules such as reducing the maximum allowable $CO_2$ value or adding a derivative term could further improve violation performance with minimal effect on power consumption. In the absence of far-improved localization systems that can provide instantaneous and accurate occupancy estimates, we believe the strength of these results highlights the importance of incorporating $CO_2$ sensors into ventilation systems in dense settings with variable occupancy such as conference rooms.

As we deploy this system throughout our testbed building, we see similar performance in other rooms. As of this writing, there are $CO_2$ sensors deployed in 7 of the 10 conference and class room settings in the building, with our DCV system running on 6 of them, saving roughly 2.7 kW continuously out of approximately 3.9 kW used for the ventilation systems in these rooms (69% savings), showing that a small number of sensors and limited application code can augment an existing building to both save power as well as improve air quality.

**B. Demand-Controlled Filtration**

The second application is primarily concerned with the other setting found in our test building: a chip fabrication facility. Our testbed building has over 15000 $ft^2$ of Class 100 and Class 1000 cleanroom. In these settings, maintaining low particle counts of impurities is critical. As such, besides using VAV systems for injecting fresh air into the space, recirculating air handler units (RAHUs) are used to continuously push air through particle filters. In our building, there are 25 RAHU units, each consuming 2 kW, with a total aggregate airflow of around 215000 cfm; this dwarfs the airflow in our 10 conference and class rooms, which is a total of about 5000 cfm. The potential to curtail RAHU operation when the cleanrooms are not in use could save large amounts of energy; this is called demand-controlled filtration (DCF) [71].

Fig. 12. $CO_2$ (blue) and motion sensor (red) values in one bay of a cleanroom over two weeks. Identifying periods of inactivity will allow energy savings by turning down the rates of recirculating air handler units (RAHUs).

Though this application has much in common with the DCF application, the sensor required for DCF is different; Figure 12 shows two weeks of operation of a combination $CO_2$ and binary motion sensor we installed in one bay of the cleanroom. During the two weeks, the $CO_2$ concentration varies by at most 200 ppm and generally much less despite significant variance in occupancy. This is partly caused by the large air volume in mostly open spaces, but also by the continuous and frequent recirculation of air in the space, preventing the $CO_2$ sensors from providing enough indication of cleanroom occupancy. Instead, basic motion sensors are better in this instance. There are substantial periods of no motion in this bay; energy can be saved by turning down or off the relevant RAHUs during inactive periods. In this cleanroom, four RAHU units cover a single bay, so it is possible to save anywhere from 4-8 kW, depending on the conservativeness of the RAHU strategy. The implementation of DCF using BOSS/BAS allows us to scale this application throughout the cleanroom, allowing further savings with almost zero additional development effort.

Another parameter that determines possible energy savings is how aggressively the controller actuates the fan. We construct a simple controller that waits an inactivity period after any motion event before modulating fan speed. The selection of this parameter presents a tradeoff between the frequency of fan actuations and the total time with the fan spent at a lower speed or off, as seen in Figure 13. By more aggressively setting the inactivity parameter, the fan actuates more often, creating additional wear and tear on the equipment, but saving more energy. It may be possible to learn researcher behavior to construct predictive models to further improve performance. In summary, in this DCF application, a small add-on network has the potential to unearth substantial power savings.
Another potential application for DCV and DCF systems is as a supply-following load [72]. In this scenario, the rate of ventilation would be modulated to make the energy consumption of the supply fan better match the availability of electricity from the grid. This becomes more valuable as non-dispatchable renewable sources such as solar and wind comprise a larger proportion of generation on the electricity grid. It is important to note, though, that the slack, or capacity to change, in the load is limited in one direction; that is, in nearly all situations, the system is running as efficiently as possible, and energy consumption cannot be reduced any further. However, at the same time, these systems can nearly always increase consumption to better match a surplus of grid electricity. This potential to sink extra electricity could be used in combination with other loads with different characteristics to provide supply-following capacity.

A key aspect of the two applications profiled in this work but also of many of the applications referenced in Section III is the substantial benefit of additional hardware. In both the DCV and DCF applications, adding a small number of sensors enables an enormous change in operation of the control system and results in significant energy savings. One critical opportunity in using a platform that enables rapid development of building applications is to identify and exploit the instances where an incremental addition of a piece of hardware or software enables a non-incremental benefit in performance.

VI. CONCLUSIONS

The key to energy-efficient buildings in the future is delivering applications, such as DCV and DCF, that are customized to the needs of a site, and yet can also naturally evolve as technology improves and the site is reconfigured. Existing systems are ill-suited for this model of continuous change because reconfiguring them requires significant manual effort unique to each site. As a result, buildings’ performance is widely known to continuously degrade following any commissioning or recommissioning effort. Using BAS and BOSS, we are able to install applications onto existing building infrastructure, integrating a network of embedded sensors with calendar data from the Internet and the existing control system in a way that can easily be modified to take advantage of new sources of occupancy data like class or meeting schedules, network activity monitors, and other sources as they become available. By improving ventilation and filtration control, we simultaneously enhanced air quality and achieved significant energy savings – by over 80% and 50%, respectively – in a widely deployable way; furthermore, the pattern represented by these example applications embodies a broad class of building applications, hinting at the wide scope a programmable platform for buildings might ultimately encompass.

ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation under grants CPS-0932209 (LoCal), CPS-0931843 (ActionWebs), and CPS-1239552 (DB). We also thank Albert Goto for deployment support.

REFERENCES
