

UC Berkeley

Indoor Environmental Quality (IEQ)

Title

Writing controls sequences for buildings: from HVAC industry enclave to hacker's weekend project

Permalink

<https://escholarship.org/uc/item/3671b82b>

Authors

Peffer, Therese
Pritoni, Marco
Fierro, Gabe
[et al.](#)

Publication Date

2016-09-08

Peer reviewed

Writing controls sequences for buildings: from HVAC industry enclave to hacker's weekend project

Therese Peffer, California Institute for Energy and Environment, UC Berkeley

Marco Pritoni, Western Cooling Efficiency Center, UC Davis

Gabe Fierro, Software Defined Buildings, UC Berkeley

Soazig Kaam, Joyce Kim, Paul Raftery, Center for the Built Environment, UC Berkeley

ABSTRACT

The advent of net zero buildings and increasingly stringent energy codes for new construction points us in the right direction towards reducing our carbon footprint, but what about the energy performance of the vast numbers of existing buildings? Utilizing advanced control sequences in existing buildings can be a cost-effective way of reducing energy use. Smaller commercial buildings (less than 50,000 square feet (sf)) typically have manually controlled systems (e.g., thermostats and light switches), and traditional controls for large commercial buildings—while typically digital (e.g., Building Automation Systems)—nevertheless use 20-30 year old technology. These systems were designed to be robust and perform simple tasks such as maintaining temperature and pressure setpoints. However, the controls cannot be easily reprogrammed to incorporate new control strategies. Since the controls are proprietary to each vendor, changing the control logic is expensive. A simple change in ventilation rate in each zone of a building may require hours of programming, rendering the task impractical.

In the past few years at UC Berkeley, computer scientists have been developing innovative software tools and platforms. The simple Monitoring and Actuation Profile (sMAP) software provides a consistent interface to data from various sources such as building sensors, networked devices, websites and other programs allowing people to query and use this data to write flexible and extensible control code. The community around this software has steadily grown and now includes new users with domain-specific knowledge (e.g., mechanical engineers and architects) but with limited background in computer science. This paper explores a few examples of how this new community used these tools to write advanced control sequences in real buildings and test innovative energy efficiency algorithms and components.

Background

Building Automation Systems (BASs) serve the majority of large (greater than 100,000 sf) commercial buildings in the U.S., and are estimated to save 5-15% of overall building energy (E-Source 2016). These existing buildings are difficult environments in which to implement innovative control strategies; despite standardized protocols like BACnet¹, it is difficult to programmatically inspect the installed systems and automatically apply advanced optimizations at scale and with low cost. Each manufacturer of BAS—Siemens, ALC, Johnson Controls, Schneider, Distech, and so on—uses proprietary software: thousands of lines of code, obscure

¹ Developing the BACnet protocol was an international effort led by the American Society for Heating Refrigeration and Air-Conditioning Equipment (ASHRAE) to improve interoperability between building control systems, started in the 1990s (<http://www.bacnetinternational.org/>).

acronyms specific to the BAS and the building, and hard-coded names. Control products are vertically integrated with software, hardware, and maintenance services provided by a single company or a limited number of industry partners. This market strategy effectively locks customers into single-supplier solutions, since it is difficult to add or replace components of an existing system with competitors' products. This oligopolistic market does not incentivize innovation and after 40 years of their first introduction, BASs have not significantly evolved.

Programming a BAS, commonly referred to as “writing control sequences,” is still a prerogative of specialized technicians who need to know the vendor-specific language, the configuration of the building, and all the details about the installation. Most of the programming time is spent in matching sensor names and writing long and primitive code that cannot be reused (e.g., it works with only one component in one building). Figure 1 shows one of the several pages needed to control the normal operation of an HVAC component. The code is clearly not reusable, hard to maintain (in case of any changes), and difficult to understand at first glance. Adding more advanced features (e.g., demand response) or improving sequences to comply with the latest energy code can be unwieldy, typically requiring hundreds of hours of work if applied to an entire building.

```

01000 ONPWRT(1040)
01010 DEFINE(X,"YOUNG.AHU1.")
01020 SAMPLE(300) "X%OAC" = "X%OAT"
01022 SAMPLE(300) "YOUNG.MAIN.OAT" = "YOUNG.AHU1.OAT"
01030 GOTO 1100
01040 SECDN7 = 8000
01100 LOCAL(SMADRM,COOLON,COLOFF,MADRCT,MADCTL,MATCTL,HUMOVR,CLVCTL,ECONDB,CLVOVR,OVR)
01110 LOCAL(SMTMIN,SMTMAX,DB1,DB2,DB3)
01200 SAMPLE(5) IF("X%SAF".EQ.PRFON) THEN "SMADRM" = "SMADRM" + 5 ELSE SET(0.0,"SMADRM")
01210 SAMPLE(5) IF("X%COL".EQ.ON) THEN "SCOOLON" = "SCOOLON" + 5 ELSE SET(0.0,"SCOOLON")
01220 SAMPLE(5) IF("X%COL".EQ.OFF) THEN "SCOLOFF" = "SCOLOFF" + 5 ELSE SET(0.0,"SCOLOFF")
01230 TABLE("X%OAO","X%MAM",0.0,20.0,100.0,50.0)
01298 C
01300 TABLE("X%HC0","X%HCV",0,0,100,10)
01310 TABLE("X%CC0","X%CCV",0,0,100,10)
01320 TABLE("X%MA0","X%MAD",0,2,100,10)
01330 TABLE("X%AO","X%AD",0,2,100,10)
01331 C
01333 C
01335 C
01335 C
01335 C
01340 IF("X%MAT".LT.30.0.OR."X%AT".LT.34.0) THEN ON("X%FREEZE")
01350 IF("X%FREEZE".EQ.#OPER) THEN RELEAS(#OPER,"X%FREEZE")
01390 C
01392 C
01394 C
01394 C
01396 C
01396 C
01398 C
01400 DBSWIT(0,"X%CO",33,35,"X%COL")
01410 IF("SCOOLON".GT.300) THEN ON("X%VLV")
01420 IF("SCOLOFF".GT.1800) THEN OFF("X%VLV")
01494 C
01496 C
01498 C
01498 C
01500 "SECONDB" = "X%RAT" + 1.0
01510 DBSWIT(1,"X%OAC","X%RAT","SECONDB","X%ECM")
01571 C
01573 C
01575 C
01577 C
01577 C
01579 C
01580 SAMPLE(300) DBSWIT(0,"X%RMT","X%DAY HTG STPT","X%HEAT.COOL")
01582 IF("X%HEAT.COOL".EQ.1.AND."X%MODE CO2".EQ.ON.AND."X%RMT".GT."X%DAY CLG STPT").OR.("X%HEAT.COOL".EQ.1.AND."X%MODE CO2".EQ.OFF.AND."X%RMT".GT."X%NGT CLG STPT") THEN "X%DB MODE" = 3.0
01584 IF("X%HEAT.COOL".EQ.0.AND."X%MODE CO2".EQ.ON.AND."X%RMT".GT."X%DAY HTG STPT").OR.("X%HEAT.COOL".EQ.0.AND."X%MODE CO2".EQ.OFF.AND."X%RMT".GT."X%NGT HTG STPT") THEN "X%DB MODE" = 1.0
01586 IF("X%HEAT.COOL".EQ.1.AND."X%MODE CO2".EQ.ON.AND."X%RMT".LE."X%DAY CLG STPT").AND."X%RMT".GE."X%DAY HTG STPT") THEN ON(SDB1) ELSE OFF(SDB1)
01588 IF("X%HEAT.COOL".EQ.1.AND."X%MODE CO2".EQ.OFF.AND."X%RMT".LE."X%NGT CLG STPT").AND."X%RMT".GE."X%NGT HTG STPT") THEN ON(SDB2) ELSE OFF(SDB2)
01590 MAX(SDB3,SDB1,SDB2)
01592 IF(SDB3.EQ.1) THEN "X%DB MODE" = 2.0
01594 IF("X%SP TYPE".EQ.0) THEN GOTO 1600 ELSE GOTO 1640
01595 C
01595 C
01596 C
01600 LOOP(128,"X%RMT","X%SAS","X%RSP","X%P16","X%I16",0,10,"X%ZSP",60.0,85.0,0)
01601 C
01602 IF("X%ZSP".LT."X%ZSPHI".AND."X%ZSP".GT."X%ZSPLOW") THEN "X%RSP" = "X%ZSP" ELSE GOTO 1620
01610 GOTO 1700
01620 IF("X%ZSP".GT."X%ZSPHI") THEN "X%RSP" = "X%ZSPHI"
01630 IF("X%ZSP".LT."X%ZSPLOW") THEN "X%RSP" = "X%ZSPLOW"
01632 GOTO 1700
01633 C
01635 C

```

Figure 1: Page 1 of 5 of a traditional control sequence to control the normal operation of an air-handling unit using traditional vendor-specific control code.

But what if we can learn from the smart phone “there’s an app for that”-world and apply that thinking to buildings? Smart-phones use a layered architecture, where different Apps are supported by an Operating System that interacts with the hardware. Updates are continually provided in the background, as necessary. Instead of a BAS, perhaps we should consider a Building Operating System (BOS). This BOS would:

- provide access to the hardware in a more abstract way (e.g., providing a query language to retrieve addresses of the sensors based on their function and position and not their unique name)
- provide access to other sources of information that are external to the building (e.g., price signals, weather forecast, calendars)
- use modern open-source, object-oriented programming languages, such as Python² or Javascript³ that are supported and developed by a larger community of programmers
- use open Application Programming Interfaces (API) to exchange data with other applications
- be organized in different layers, such as one for the interaction with the hardware, one that provides essential services such as data storage and security and one that allows programmers to develop innovative applications (Figure 2).

Can a software platform allow “hackers” to nimbly create new energy efficient algorithms in large commercial buildings with existing BAS? For the last few years at UC Berkeley, computer scientists have been developing such a platform. The first innovation was the Simple Monitoring and Actuation Profile (sMAP, pronounced ess-map) for aggregating and managing data sources (Dawson-Haggerty 2010). A core aspect of sMAP is the Hardware Presentation Layer, a layer of abstraction that establishes uniform read/write access to the set of devices and data sources in a building. With this open-source tool, any time-series data stream—whether online weather, third party sensors, hardware devices, or data from the BAS—can be labeled or tagged and stored in a fast, easily queried database. One can add new data streams by writing a simple “driver” interface. Thus sMAP enables interoperability among heterogeneous devices and systems.

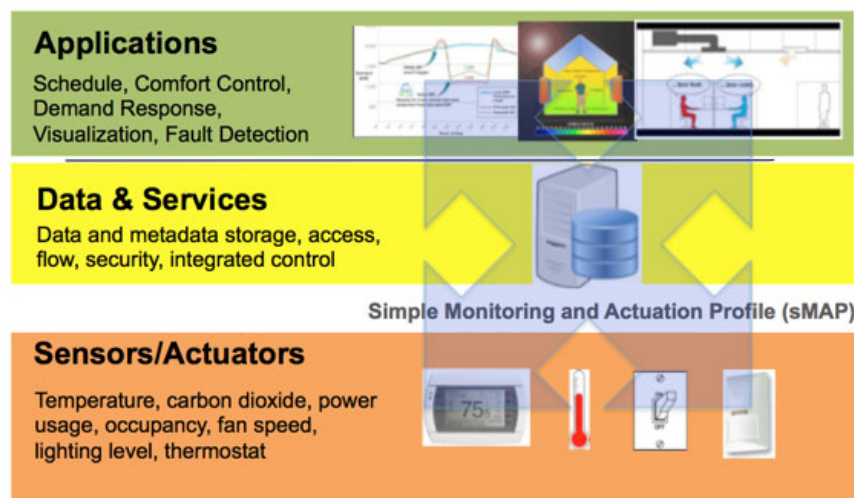


Figure 2: Open-Architecture of a Building Operating System (Blumstein 2014)

Several researchers developed a service platform called Building Operating System Services or BOSS (Dawson-Haggerty 2013) to enable using this rich data collection for

² <https://www.python.org/>

³ <https://www.javascript.com/>

applications. Previous papers have described the innovation and interoperability of this open layered software architecture enabled by sMAP (Blumstein 2014); see Figure 2. The bottom layer (Sensors/Actuators) holds all the data sources, whether devices or online data streams. An sMAP driver interfaces the data source with the middle Data & Services layer. This data management layer provides data and metadata⁴ storage, query management, and provides access to the data from the applications layer. Recently, one graduate student created software to automate the classification of metadata by learning from a facility manager's examples, and developed algorithms to quickly find rogue zones (zones with temperatures that never meet the setpoint) in any building (Bhattacharya 2015a, Bhattacharya 2015b, Pritoni 2015). The platform was further updated, in a third iteration, which implemented a new faster database (called BTrDB) (Andersen 2016), improved the actuation service, updated the archiver (now called Giles), and added a manager of the evolution of building configurations to create the eXtensible Building Operating System (XBOS) platform (Fierro 2015).

sMAP led to a number of applications that were implemented in Sutardja Dai Hall (SDH) at UC Berkeley and the Kress building in downtown Berkeley. One early application was demand response—reducing peak load in SDH by nearly 30% (Auslander, 2012). One researcher created an easy to use web-based lighting control that was a great improvement over the poor usability of the wall switches in SDH and also saved energy (Krioukov, 2011). Another project, Demand Controlled Ventilation, added carbon dioxide sensors to the conference rooms in SDH as well as calendar information to control the ventilation through the BAS; the implementation reduced energy consumption and improved air quality during meetings (Taneja 2013). The XBOS platform enabled a virtual BAS in a small commercial building, the Kress building, by creating a central interface for multiple networked thermostats, the building smart meter, and plug load control (Fierro, 2016).

This paper outlines several recent applications in buildings on two different campuses, and describes some of the successes and failures. In reviewing this work, we entertained several questions. How portable are the applications—how easy is it to truly write the code once and have the algorithm work in another building? Sutardja Dai Hall's Siemens Apogee BAS⁵ contains over 6000 data points; graduate students spent much time deciphering the metadata, mapping the systems' control points to the floor plans, and understanding the control sequences (Peffer 2012). Also, can non-computer scientists effectively use these tools to visualize, monitor, and/or control building systems to implement energy efficient strategies, effectively enable people who are domain experts to accomplish their research tasks? Certainly, sMAP use has expanded beyond UC Berkeley's Computer Science department: a survey from September 2014 found several users of sMAP: two other departments at UC Berkeley, Lawrence Berkeley National Laboratory (at least seven projects, not including the data management of the FLEXLAB⁶), UC Davis (3 projects), universities in Denmark, India, and China, and two Bay Area start up companies. Most of the projects dealt with data collection and management. We briefly describe a few recent applications, and then discuss some of the issues.

⁴ metadata is “data about the data”, information tags or labels about the sensor data, for instance the type of sensor, the role in the HVAC system, the zone it belongs to, and so on.

⁵ <http://w3.usa.siemens.com/buildingtechnologies/us/en/building-automation-and-energy-management/apogee/Pages/apogee.aspx>

⁶ <https://flexlab.lbl.gov/>

Research Methods

UC Berkeley Projects

The Center for the Built Environment (CBE) in the Architecture Department at UC Berkeley has used sMAP to monitor, develop and test several control algorithms in Sutardja Dai Hall (SDH) as part of a California Energy Commission funded project called Changing the Rules. sMAP serves an essential role of integrating the distinct advances: wider temperature setpoint ranges, improved supply air temperature and pressure resets, personalized control systems, human comfort modeling, and real-time occupant feedback.

Personal Comfort System Chairs. One research project observed and surveyed occupants regarding their use of Personal Comfort System (PCS) chairs, which are mesh office chairs outfitted with heating strips and fans that occupants can control to provide personal thermal comfort (Bauman 2015). Recently, the chairs were equipped with wireless 802.15.4 radio and Bluetooth, and sMAP was utilized to aggregate the data in real-time (Figure 3). Each chair has occupancy, temperature, and relative humidity sensors, and can also relay data about whether the seat and/or back is in heating or cooling mode. Border routers relay the data to the database via the Internet.

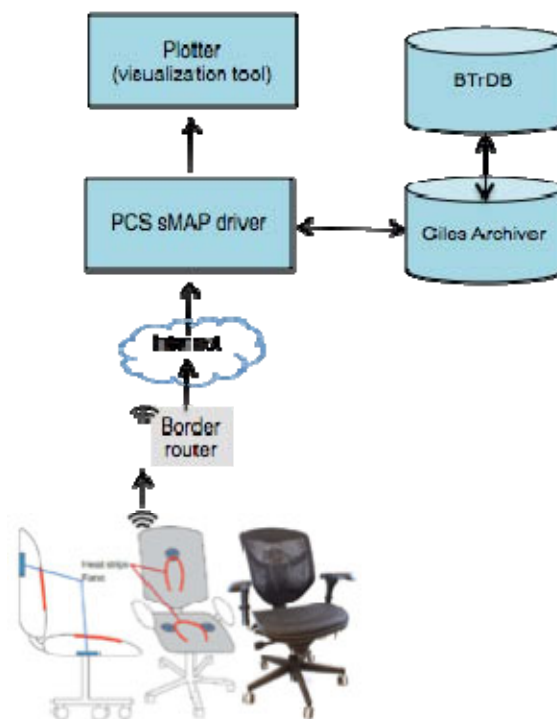


Figure 3: System architecture for the Personal Comfort System chairs with the latest sMAP-stack software tools (Giles and BTrDB).

Advanced HVAC Control Strategies. CBE implemented several control algorithms to improve energy performance of SDH. Air-Handling Units (AHU) with Variable Air Volume-boxes (VAV) are the most common HVAC types in large commercial buildings in the U.S. Typically the BAS does not coordinate the different components of these systems with resulting energy

waste due to excessive ventilation and simultaneous heating and cooling. The researchers implemented AHU reset strategies and Time Averaged Ventilation (TAV) (Soazig in preparation) for VAV boxes. These strategies reduce the amount of local reheat and ventilation in a building.

The first implemented control strategy was to reset supply air temperature and duct static pressure. The AHUs that serve the office spaces in SDH did not use either supply air temperature or static pressure resets as described in the most recent code requirements. Instead, it used fixed setpoints of 58°F and 1.4 in. w.c. respectively. Current best practice controls (required by California energy code, Title 24⁷) for both of these reset strategies modify the supply air temperature and pressure setpoints based on the requests from each zone in the building, which is significantly more efficient than using static setpoints. The most common strategy for implementing demand-based resets is “trim & respond.” Each zone generates a “request” when there is demand, such as for additional duct static pressure if the zone damper is nearly fully open while trying to meet the zone airflow setpoint. The requests are accumulated at the parent system, and the system responds by adjusting the setpoint when the number of requests exceeds a minimum threshold. Otherwise, the setpoint is gradually “trimmed” to minimize energy use. In the case of supply air temperature reset, the setpoint decreases when there are requests, and increases when there are none, within limits that vary based on outside air temperature. The researchers also developed and tested dynamic suppression of requests from rogue zones. This is an improvement over current best practice code requirements, and demonstrates the flexibility of using this approach for making changes to control sequences.

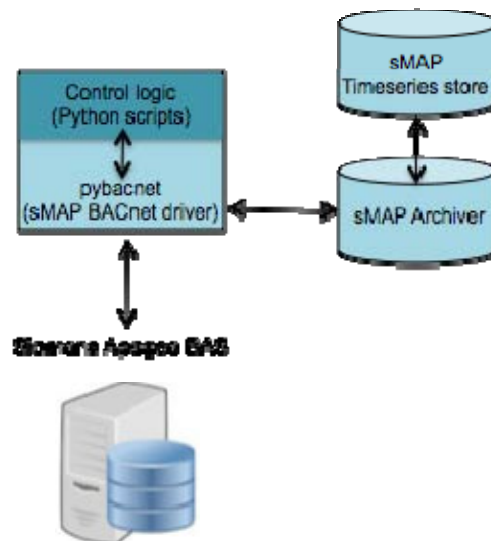


Figure 4: Implementation of control strategies in Sutardja Dai Hall at UC Berkeley using an sMAP BACnet driver called pybacnet.

The second control strategy was Time Averaged Ventilation (TAV). In typical building systems the minimum VAV box airflows are higher than the ventilation minimum required by current code (Title 24 and ASHRAE 62.1) due to a combination of perceived and real issues regarding VAV box control stability and measurement accuracy at low flows. Thus the majority of VAV boxes in buildings are supplying ventilation air at unnecessarily high rates, which

⁷ <http://www.energy.ca.gov/2012publications/CEC-400-2012-004/CEC-400-2012-004-CMF-REV2.pdf>, p 186.

wastes both fan and cooling energy and creates uncomfortably cool spaces, often during summer periods. To resolve this issue, CBE developed a Time Averaged Ventilation (TAV) strategy that can be applied to each zone in the building. It cycles the airflow from 0% (i.e., a fully closed damper) to a higher value, defined by the larger of either 30% of the design maximum airflow or the maximum airflow required to avoid stratification at the design heating condition. Time averaging is allowed by ASHRAE Standard 62.1-2013 and by Title 24 2013. When considering the average flow over a period, TAV controls to the ventilation minimum for each space as opposed to the controllable minimum (typically much higher), leading to energy savings.

Researchers used sMAP and the sMAP-BACnet driver⁸, called the pybacnet package, to interface their Python code to actuate systems via the BACnet protocol. The software runs on a miniature computer connected to the building’s Siemens Apogee BAS (Figure 4).

UC Davis Project

Researchers at UC Davis integrated the campus-wide thermal feedback application with sMAP to develop a participatory thermal feedback and control system (Pritoni 2016). A prototype of the system was deployed in a small office building in campus conditioned by rooftop units (small packaged HVAC systems). The thermal feedback application, at the top of Figure 5, provides real-time information to the control logic, through a driver. A second driver connects the control logic to the thermostats. Each data stream is saved in a time-series database, represented on the lower right. The whole software stack is implemented in the cloud using Amazon Web Services. The control engine can easily switch between control strategies to iteratively improve and debug the system and test the best alternative algorithm.

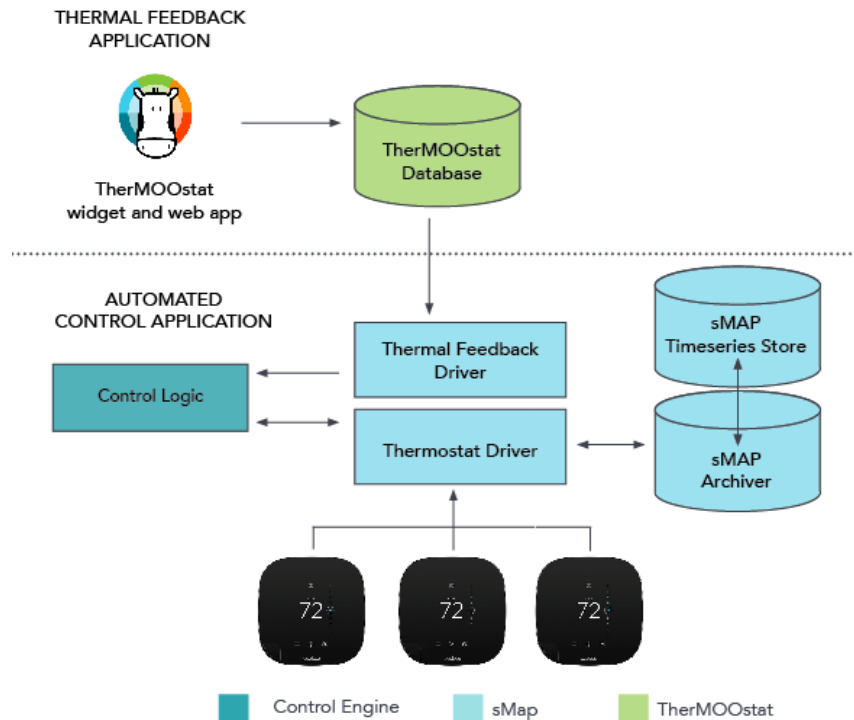


Figure 5: Implementation of a participatory thermal feedback and control system at UC Davis

⁸ <https://github.com/BuildingRobotics/pybacnet>

The researchers implemented and compared three control strategies:

- 1) Traditional schedule based on typical campus setpoints and hours.
- 2) Direct setpoint adjustment based on comfort feedback.
- 3) Drifting algorithm. The algorithm takes into account occupant feedback as (2.) above, but if no feedback is provided, it slowly adjusts the temperature setpoint towards the outdoor temperature. Further, where occupancy is not detected, the algorithm converts to energy-saving settings.

Results

UC Berkeley test results

Regarding CBE's study using wirelessly connected PCS chairs, the researchers learned about individuals' thermal preferences based on their survey feedback and control behavior for heating and cooling. sMAP enabled data reporting and access in real-time, providing great convenience and transparency to the researchers. Also, sMAP real-time data streams allowed survey email reminders to be automatically dispatched based on measured readings such as temperature or PCS usage to improve the predictive power of individuals' thermal comfort models and to manage HVAC setpoints that reflect the actual comfort preferences of occupants. The research team can now develop machine learning algorithms to produce actionable inputs to HVAC operation to provide data-driven comfort management in buildings.

CBE implemented various control strategies using the open source sMAP architecture and the pybacnet package, without using the proprietary BAS interface. The implementation was not painless. The various Python scripts to implement TAV in every VAV box in the building contain about 1700 lines. Researchers had to incorporate additional data streams from the BAS as well as add data streams from the external Discharge Air Temperature sensors added to the system. CBE researchers added additional metadata tags using the sMAP-query tool capabilities. In addition, the researchers added a script to catch errors and notify the operator by email.

The AHU reset strategies required approximately 600 lines of code and have been running constantly since October 2014, providing significant energy savings. The software is portable between buildings once the BACnet data is appropriately tagged, and thus, the scale for cost-effective retrofit of existing large commercial buildings is significant. CBE is now performing research to further refine the supply air temperature reset strategy above and beyond current best practice so that it operates at the most cost efficient setpoint (e.g., by accounting for availability of the airside economizer and the cost of zone reheat).

UC Davis test results

Integration of sMAP with a web-connected thermostat without a local API⁹ was another informative experience. On the one hand it allowed a full cloud implementation (that is, no local computer needed), with significant cost advantages if scaled up to the whole campus. On the other hand, the research team experienced several connection and data loss problems due to unavailability of the vendor's website or other cloud-to-cloud communication issues. sMAP was used here also to integrate two other software platforms in addition to the thermostats. One is a

⁹ It was possible to programmatically interact with the thermostat in the room only via the vendor website, thus if the Internet connection was lost the thermostats was impossible to control and reverted to the local schedule.

custom application developed at UC Davis to collect thermal feedback from building occupants, the other is OsiSoft PI¹⁰ used to gather energy meter data.

After the successful deployment of this prototype, UC Davis Facility Management is interested in applying the same concept to larger buildings and is currently trying to find a way of writing control code for the Siemens BAS without violating campus IT security rules.

Discussion

How easy is it to use these tools? One anecdote suggests that they are easier to use than the current state of practice! After monitoring the ventilation system of SDH, we asked the facilities manager to enact some permanent reductions in the minimum ventilation rate through the BAS. He asked if we could implement it using Python/sMAP instead, since for him to implement this in the BAS for each of the 130+ VAVs would take much longer than to write a couple lines of code. Testing innovative control strategies, such as the ones developed in UC Berkeley and UC Davis, requires writing code, implementing in the buildings, monitoring the result, and iterating multiple times. This process would be very difficult and time consuming trying to achieve this through the existing BAS.

sMAP/XBOS are promising approaches to provide simple, uniform abstractions of building components, sensors, and other data sources to application programmers in a way that makes it easier to automatically apply a range of techniques across a large set of buildings. The majority of large commercial buildings in the U.S. use Direct Digital Controls and are compatible (or can be upgraded to be compatible) with BACnet, but these buildings do not currently implement advanced controls. sMAP/XBOS could enable the quick modernization of these buildings to meet new energy code requirements and improve their energy performance.

However, several issues need to be addressed to be able to create applications that are truly portable between buildings. The first challenge occurs at the very beginning of the process, during the system setup. Even though progress has been made in automating metadata acquisition, a significant amount of time is still needed to map the legacy BAS points, their structure and relationships to be used in a new control application. Some of the problems are inconsistency in naming convention, human errors, and limitations in the underlying BAS software and hardware. A second category of problems is related to the lack of standardization in HVAC configurations. Two AHU-VAV buildings may have different types of sensors, number of fans, duct layout, and sensor positions since these systems are assembled on site and based on custom design. While some of these differences are driven by legitimately different requirements, they are sometimes due to arbitrary decisions taken by the manufacturer, designer, or controls engineer without any firm reason (e.g. should we include a mixing box temperature sensor in the AHU?). When metadata is acquired, it is difficult to capture these differences from the information stored in the BAS. Also, user interfaces that may display HVAC configuration are designed for humans and they are not precise enough to provide that information to a machine.

To write portable control sequences between buildings, programmers need to develop templates¹¹ that push the differences in HVAC configurations and sensor types to a “lower” or more abstract level. An HVAC template should provide the list of sensor and actuation points that are available for the specific configuration (e.g, one or more fans) and the possible

¹⁰ https://www.osisoft.com/software-support/what-is-pi/What_Is_PI.aspx

¹¹ e.g., using classes, in object-oriented programming

“methods¹²” that are accessible (e.g., set valve position). These templates should also capture the function of the sensors for control purposes, in addition to the type and unit of measurement. For example, the temperature sensor that measures the supply air temperature at the AHU can be used in different control sequences. If there are two fans and two sensors, and the two are averaged for control purposes, the method “read_SAT¹³” should provide the average value using a lower-level function. By organizing the code in this way, the control strategy can be easier to read and understand, showing the high-level behavior, while other pieces of the code hidden to the user manage the translation to the specific system. Creating such templates requires a clear understanding of HVAC configurations and controls sequences requirements. Researchers involved in these projects are currently defining these abstractions after the first iteration of software development focused on the functionalities of the control code, rather than the form. Future work should also look at adding a GUI for building operators that are not skilled programmers.

These projects also highlighted that the distributed nature of HVAC systems needs to be considered when developing control code. For example, thermostats and packaged HVAC units have on-board controllers that directly actuate the equipment. Sometimes only their setpoints can be changed using software tools. In these cases, the global behavior of the HVAC system strongly depends on the particular hardware implementation of these controllers. This is an important limitation to take into account, given the goal of dealing with existing buildings and BAS without massive hardware replacements.

While “hacking” BASs seems initially cost-effective, we are concerned about the business model of open-source, open architecture. These tools have to get beyond the hands of computer scientists to controls engineers, but who will develop and maintain them? Certainly the monitoring capability of sMAP is commercially available (Trendr by Building Robotics¹⁴), but tools to enable automating metadata, matching the building control points, and enabling actuation are all currently academic constructs, with no plans for commercialization.

We note that most of the co-authors are not computer scientists, but are mechanical engineers who understand the control sequences and eagerly learned Python in the last few years in order to test and implement innovative control strategies in buildings. Over the past few years, we have recognized the difficulties the different disciplines have in communicating with each other, but the value of these tools encourages us to persevere. In our interactions, the computer scientists were concerned about the structure and efficiency of the code (for example, distinguishing apps from services from drivers), whereas the mechanical engineers were less concerned about the quality and design of the code and more concerned about the function. We plan to continue to learn from and with each other.

Conclusion

The objective of this paper was two-fold: 1) to determine if the sMAP stack helped researchers to “hack” the traditional BAS in existing buildings, implement innovative energy efficient measures, and interface with new IoT devices¹⁵, and 2) whether these tools are limited to computer scientists or if mechanical engineers and architects can learn to hack too. At UC

¹² methods are software functions in object-oriented programming

¹³ SAT=supply air temperature

¹⁴ <https://trendr.buildingrobotics.com/>

¹⁵ IoT=Internet of Things

Berkeley, researchers used tools based on sMAP and XBOS to conduct testing of Personal Comfort System chairs; these tools enabled convenient real-time monitoring and helped the researchers automate sending email notices for occupants to take the survey at key times. Researchers also used an sMAP-BACnet driver, pybacnet, to implement new control strategies: Supply Air Temperature reset, Duct Static Pressure reset, and Time Averaged Ventilation. sMAP and pybacnet successfully enabled adding new sensors to the database and resulted in reduced ventilation and reheat energy. At UC Davis, researchers successfully used sMAP to interface networked thermostats with a web-based thermal voting App to change temperature setpoints based on occupant comfort votes. While not natural hackers, researchers managed to use these tools and platforms to successfully and nimbly enact innovative energy efficient algorithms and components in buildings.

Acknowledgements

Research conducted by the Center of the Built Environment as part of the Changing the Rules project is funded by the California Energy Commission. Development of sMAP was partially funded by the National Science Foundation under grants CPS-0932209 and CPS-0931843 and the California Energy Commission. A precursor of XBOS (OpenBAS) was funded by the Department of Energy. The project in Davis was partially funded by a Seed Grant from CITRIS, the Center for Information Technology Research in the Interest of Society. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, California Energy Commission, or the Department of Energy.

References

- Andersen, Michael P, and David E. Culler. 2016. "BTrDB: Optimizing Storage System Design for Timeseries Processing." 14th USENIX Conference on File and Storage Technologies (FAST 16), Santa Clara, CA, Feb 2016.
- 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/sutardjaidai/DIADRFinalReport.pdf>.
- Bauman, F., H. Zhang, E. Arens, P. Raftery, C. Karmann, J. Feng, Y. Zhai, D. Dickerhoff, S. Schiavon, and X. Zhou. 2015. Advanced Integrated Systems Technology Development: Personal Comfort Systems and Radiant Slab Systems. California Energy Commission. www.escholarship.org/uc/item/88p8v7zb.
- Bhattacharya, Arka, Dezhi Hong, David Culler, Jorge Ortiz, Kamin Whitehouse, and Eugene Wu. 2015a. "Automated Metadata Construction to Support Portable Building Applications " BuildSys 2015, Seoul, Korea.
- Bhattacharya, Arka, Joern Ploennigs, and David Culler. 2015b. "Analyzing Metadata Schemas for Buildings : The Good, The Bad and The Ugly " BuildSys 2015, Seoul, Korea.

- Dawson-Haggerty, Stephen, Xiaofan Jiang, Gilman Tolle, Jorge Ortiz, and David Culler. 2010. "sMAP — a Simple Measurement and Actuation Profile for Physical Information." Proceedings of the Eighth ACM Conference on Embedded Networked Sensor Systems (SenSys '10), November 2010.
- Dawson-Haggerty, Stephen, Andrew Krioukov, Jay Taneja, and David Culler. 2013. "BOSS: Building Operating System Services." Proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI '13), April 2013.
- ESource. 2016. Building Automation Systems, Building Energy Advisor. Available at: https://bizenergyadvisor.com/BEA1/PA/PA_BuildingAutomationSystems/PA-36 Retrieved March 11 2016.
- Fierro, Gabriel, and David E Culler. 2015. XBOS: An Extensible Building Operating System. EECS Department, University of California, Berkeley. Available at <http://www.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-197.html>
- Kaam, Soazig, Paul Raftery. In preparation. Time-Average Ventilation for optimized control of variable-air-volume systems: demonstration in an institutional building. in press. <http://www.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-197.html>
- Krioukov, Andrew, Stephen Dawson-Haggerty, Linda Lee, Omar Rehmane, and David Culler. 2011. "A Living Laboratory Study in Personalized Automated Lighting Controls." BuildSys'11, Seattle, WA, November 1, 2011.
- Krioukov, A, G Fierro, N Kitaev, and D Culler. 2012. "Building application stack (BAS)." BuildSys '12, Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings, New York, NY.
- Peffer, T.E., D. M. Auslander, Domenico Caramagno, David Culler, Tyler Jones, Andrew Krioukov, Michael Sankur, Jay Taneja, Jason Trager, S. Kiliccote, Rongxin Yin, Yan Lu, and Prasad Mukka. 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*.
- Pritoni M., Arka Bhattacharya, David Culler, Mark Modera. (2015) Short Paper: A Method for Discovering Functional Relationships Between Air Handling Units and Variable-Air-Volume Boxes From Sensor Data. Conference on Embedded Systems for Energy-Efficient Built Environments (BuildSys), Seoul, South Korea.
- Pritoni M., Salmon K., Sanguinetti A., Morejohn J., Modera M. (2016) Thermal Feedback for Improved Efficiency in University Buildings. under review for Energy and Buildings, Special Issue on "Energy Management for Institutional Buildings".
- Taneja, Jay, Andrew Krioukov, Stephen Dawson-Haggerty, and David Culler. 2013. "Enabling Advanced Environmental Conditioning with a Building Application Stack." 4th International Green Computing Conference (IGCC '13), June 2013.