International High Performance Buildings Conference

School of Mechanical Engineering

2014

A Novel Human Machine Interface for Advanced Building Controls and Diagnostics

Ritesh Khire
United Technologies Research Center, United States of America, khirer@utrc.utc.com

Francesco Leonardi
United Technologies Research Center, United States of America, leonarf@utrc.utc.com

Paul Quimby
United Technologies Research Center, United States of America, quimbypw@utrc.utc.com

Soumik Sarkar
United Technologies Research Center, United States of America, sarkars@utrc.utc.com

Khire, Ritesh; Leonardi, Francesco; Quimby, Paul; and Sarkar, Soumik, "A Novel Human Machine Interface for Advanced Building Controls and Diagnostics" (2014). *International High Performance Buildings Conference.* Paper 144.
http://docs.lib.purdue.edu/ihpbc/144

A Novel Human Machine Interface
for Advanced Building Controls and Diagnostics

Paul W. Quimby[1]*, Ritesh Khire[2], Francesco Leonardi[3], Soumik Sarkar[1]

[1]UTRC, Decision Support & Machine Intelligence Group,
East Hartford, CT, USA
quimbypw@utrc.utc.com, sarkars@utrc.utc.com

[2]UTRC, System Dynamics and Optimization Group,
East Hartford, CT, USA
khirer@utrc.utc.com

[3]UTRC, Embedded Systems Group,
East Hartford, CT, USA
leonarf@utrc.utc.com

* Corresponding Author

ABSTRACT

A new generation of Human Machine Interfaces (HMI) for building automation systems is needed to allow facility managers to leverage the potential of advanced controls and diagnostics. In this paper we describe a design process and the end product, a novel HMI prototype and the system that supports it. The system is an integration of advanced algorithms, an underlying software architecture, building equipment, and the human operators that use it.

Recent developments in building controls and diagnostics techniques promise to improve occupants comfort while minimizing energy consumption. Advanced diagnostics algorithms can detect equipment failures and anomalous behaviors, while also estimating the energy and comfort impact of faults disrupting normal operation. New sophisticated control schemes can regulate a building based on past and future conditions rather than a static model. These control schemes can also automatically adapt to equipment failures, thereby maintaining the highest comfort given the available resources. There are several hurdles that must be overcome to effectively deploy these technologies. The perceived algorithmic difficulty of these approaches and the absence of proper tools to leverage them create a gap between what we know is computationally possible and operators in the field. One of the biggest problems is that current Building Management Systems (BMS) are not designed to natively support these advanced capabilities even if they were commercially deployable today.

As a part of the Department of Energy (DoE) sponsored Energy Efficient Building Hub (EEB Hub), a team led by UTRC prototyped a new HMI that natively supports a variety of advance features. Within the EEB Hub, several academic and industrial teams are experimenting with new technologies to reduce the energy footprint of buildings. In collaboration with these teams, UTRC integrated novel diagnostic and control techniques with building automation infrastructure to better understand the possibilities of a new HMI for building applications.

1. INTRODUCTION

There is substantial interest and funding by many sources towards the development of energy efficiency measures for commercial buildings using scalable automated diagnostics and intelligent supervisory control. In this context, several studies have shown that energy savings potential of such technologies can be greater than 20% of whole-building energy usage Sarkar *et al.* (2013), Schein *et al.* (2006). However, the industry has been slow to offer these advanced information-rich features primarily because of high implementation costs associated with providing site-specific solutions relative to the savings potential. New state-of-the-art automated building diagnostics systems can provide advanced warning of an impending fault in the building system along with substantial information regarding the root-cause of the problem Katipamula *et al.* (2005a and 2005b). However, unless facility managers or technicians can absorb the information and act upon the issue, this capability will not translate into utility in the form of comfort or energy savings. Supervisory control modules can provide set point recommendations for a better trade-off between energy usage and comfort. However, it may not be suitable to apply these recommendations automatically without proper human approval.

In order to prototype this system, a middleware platform for controls and diagnostics applications was implemented. Then a User-Centered Design (UCD) process was conducted to design an energy management decision support system with automated control and diagnostics algorithms. The design process involved defining market requirements through interaction with as many stakeholders as possible. Based on the requirements gathered, an interface prototype was wireframed and skinned to enable further feedback with users. The final design was then constructed as a web application and deployed on two buildings.

This paper is organized in five sections. First, the research goals of this project and relevant challenges are described. Then a brief background on the user-centered design process is presented followed by the results of the design process. The implementation and system integration process is summarized in section four and five. The implications of this work and future directions are discussed.

2. BACKGROUND

With the advent of advanced sensing, actuation and computation, buildings are becoming increasingly automated. Although an intelligent building operating system makes precise decisions in an attempt to find the optimal trade-off between energy usage and comfort, it causes an increasing number of interactions with the different human elements in the ecosystem. In this context, facility managers are high-level supervisors of the building operating system. They are responsible for interpreting high-level data from the analytics system using sensor measurements and acting upon the information available. However, to perform these tasks efficiently, building operators need decision support systems that far surpass the capabilities of those available today. In addition to large commercial buildings, data is becoming available even to average consumers such as residential customer and employees in an office. The decision support systems required for each of these uses will be different. This paper focuses on the user interfaces to help the facility managers navigate through decisions made by automated diagnostics and supervisory control systems in commercial buildings. Some key issues are the intelligence provided by the system and the scalability of the system.

In this context, *intelligent* operations means that automated diagnostics systems monitor building health through sensors and provide degradation information to a supervisory control system that optimizes building operations in response to time-varying inputs such as weather, occupancy, or utility rates. The maintenance and repair of building subsystems can also be prioritized based on fault impact such as energy waste, comfort, or equipment life and service costs estimated by the intelligent analytics system. *Fault detection and diagnostics* is a mechanism by which the data collected by the building sensors can be converted into actionable information regarding the health of building subsystems. The information then can flow into a decision support interface that enables the building facility managers to seamlessly interact and explore the diagnostics information and controls possibilities. While completely automated solutions have many advantages, due to deviations from a normal occupancy schedule or rare events, automated algorithms are too brittle to trust without human oversight. Intelligent supervisory controls can also act as assistive algorithms that suggest energy-saving possibilities by choosing optimal combinations of different set points.

Scalability implies that the complexity of implementing and using intelligent features does not change significantly with the building size. This requires that the implementation time and cost does not grow unreasonably with the size of one building or number of buildings. Similarly, the information provided to operators by the intelligent features must be handled in a reasonably way even for a large building. Unless automated recommendations are provided in a scalable manner, designed to reduce complexity, a facility manager can easily become overloaded while managing a large commercial building or group of buildings. Similarly, the wealth of diagnostic information that can be shown to a user can easily exceed manageable levels. One key scalability requirement that emerged from the user-centered design process involves the alarms generated by traditional BMS systems. An *alarm* is traditionally defined as a condition which is abnormal and will result in a user notification if it is detected. Unfortunately, since one malfunctioning piece of equipment can result in many alarms across the whole building, this is not a scalable solution. New diagnostic techniques can pinpoint root causes, dubbed *faults*. Using fault diagnostics instead of alarms in the HMI was one key innovation this novel HMI leverages. In addition, due to the under-sensed nature of the control systems involved, sufficient information for a fault detection system may not be available using passive diagnostics. Therefore the HMI must also be capable of initiating active diagnostics to aid the fault diagnostics.

3. METHODOLOGY

Many approaches can be used to generate a design for a user-facing product. The approach taken in this effort is categorized as *user-centered design* (UCD). User-centered design has many meanings according to different design experts, but the central theme remains a focus on understanding the user and involving the user in the creation of a product. The summary in this section describes UCD in terms of three phases. For further reading, we recommend *The Design of Everyday Things*, which popularized the concept and presents its basic principles.

3.1 Understanding the User

The first phase of UCD concentrates on understanding the user. Depending on the availability of the intended users of the product, it may or may not be possible to interact with the exact type of user who will use the product. For instance, it is common in advanced aerospace settings to be unable to find real users, simply because there are no such users in existence. In other cases, the number of individuals who will use the interface is simply too small to practically obtain their expertise. Regardless of whether or not it is possible to interact with the intended end-user, experts who are more closely familiar with the use of the product than the interface designers are consulted at nearly every phase of this process. These individuals are known as Subject Matter Experts (SMEs). Documenting the user can be done in several ways, most commonly by developing a fictitious persona who represents the average user. In many cases, multiple personae are created to represent different significant portions of the user population.

Once these personae exist, cognitive walkthroughs are conducted, in which interface design experts attempt to document how the SMEs do their job. This allows a designer to roleplay the actions and decisions made by the SMEs without having to always have SMEs present. The outcomes of walkthroughs are commonly scenarios that motivate the user of the product, and use-cases detailing how a product might be used in a given situation.

The final product from the first phase of UCD is a set of requirements, sometimes divided into two categories: information requirements and functional requirements. Information requirements represent the information necessary to drive decisions made by the user. Functional requirements represent the actions that the user must be capable of effecting using the interface. A better design will satisfy these requirements in a more user-friendly manner than a weaker design.

3.2 Design and Implementation

The second phase of UCD is a creative effort. The designer takes the requirements generated from the first phase and attempts to satisfy them using best practices and novel inventions. This is where non-traditional solutions to existing problems, “crazy ideas” can be tried out in addition to industry standard ideas. Frequently this phase lasts for many cycles, as ideas are thought up, explored, and rejected for one reason or another.

The outcome of this phase is commonly a set of documents or prototypes that indicate how an interface might present information and allow the user to engage with it. Paper prototypes are frequently used as a low-cost way to rapidly explore ideas. After down-selecting from a range of ideas on paper, interactive low-fidelity prototypes are often generated. Software prototypes eliminate the need for a designer to explain how the interface works as is common with the paper prototypes, because the digital system accomplishes the transitions between screens and enables some level of interactivity.

In this research, a fully-functional prototype was generated with the help of a graphic designer, who supplied a look-and-feel (colors, fonts, icons, etc.).

3.3 Feedback and Evaluation

The third phase of the UCD process evaluates the prototypes and ideas generated in the second phase, which attempted to meet the requirements from the first phase. Depending on the fidelity of the prototype generated in the previous step and the nature of the product, it may or may not be possible to give users an actual product to use in their daily work. If it is not possible to allow users to fully use the prototype, then designs on paper can be presented and the designer can demonstrate how the design would work. If an actual product exists, then the real product can be used instead of a presentation. The goal of this step is to understand how successful the design is. This is made substantially more complicated because users often reject non-traditional ideas, preferring to work with what they already have rather than risk a new idea.

Useful feedback from users includes comments pertaining to how they might use the new functionalities of the product, what elements of the product might cause new problems, what parts of the interface are intuitive or counterintuitive, and whether or not the user would prefer the new product to existing products and why.

In this research, several rounds of feedback were obtained from users, by showing them wireframe prototypes, revised designs with a proposed look-and-feel, and then the actual product once it was fully functional.

When rigorous or scientific standards are applied, a form of social science experiment, commonly categorized as *human subjects experimentation* can be conducted. The goal of a human subjects experiment is to evaluate the behavior of people, often in conjunction with a technology, by asking SMEs (or volunteers, commonly undergraduate students, when SMEs are unavailable) to use the product under identical circumstances, so as to test a hypothesis. Many metrics, some industry standards, and some from human factors, psychology, ergonomic, or cognitive science literature can be used to assess the properties of the human-product system based on how users succeed or fail to use it. For instance, one might decide to assess a decision-support system by the time it takes users to make a decision and the correctness of the decision compared to a predetermined, desired answer. Both qualitative and quantitative metrics can be used to gather subject feedback.

When designers with a strong commitment to user experience conduct UCD, they do not stop with one cycle of these three phases. The process can go on forever, with the results of phase III feeding a revised understanding of the user's requirements in phase I of another cycle. Quite commonly, revising a product will change the relationship the user has with other people. This in turn changes procedures, which in turn suggests new methodologies for accomplishing tasks. Unfortunately, limited funding and diminishing return on investment often cause companies to conclude after one cycle.

4. HUMAN-MACHINE INTERFACE DESIGN

The interface design from the final round of this research effort is presented in four screenshots. The four screens are the major screens a user interacts with during normal operation of the interface: a summary screen, a list of diagnosed faults, the details for a given fault, and a screen for adjusting the control settings for a building zone.

4.1 Overall Status Tab: (Figure 1 on page 6)

This tab provides summary information for several types of users. It aims to present the most useful “big picture” information at a glance to both technicians and facilities managers. Very prominent at the top are the summary statistics for the advanced diagnostic algorithms. The number of faults, total estimated energy impact, and monetary consequences of the faults are presented to give users an idea of how many things are currently wrong with the building and how severe those issues are. The rest of this tab is a customizable widget display. Realizing that different users will care more about different pieces of information from the building, the design supports user-configurable small interface components. For instance, a breakdown of energy usage in the building, a cumulative display of energy usage over the last thirty days, and an analysis of the building's energy usage given past historical data from days with matching outside air temperatures.

4.2 Diagnostic Tab: (Figure 2 on page 6)

This tab presents the faults identified by the advanced diagnostics algorithms. This page is a substantial departure from conventional BMS interfaces that present alarms. This tab presents root causes, not a list of symptoms. This addresses a variety of user requirements motivated by information overload complaints. Instead of seeing hundreds of alarms, all of those related symptoms are condensed down into a single fault, such as a failed or degraded component. Major features of this page include the ability to sort faults by a variety of metrics, including age, whether or not they are currently evident to building occupants, and estimated energy impact of those faults. These variables allow a user to prioritize resources to fix the most severe problems according to whichever metric they care most about. These faults can be silenced once fixed, or ignored, similar to silencing an incoming phone call if the operator does not wish to address the fault at this time.

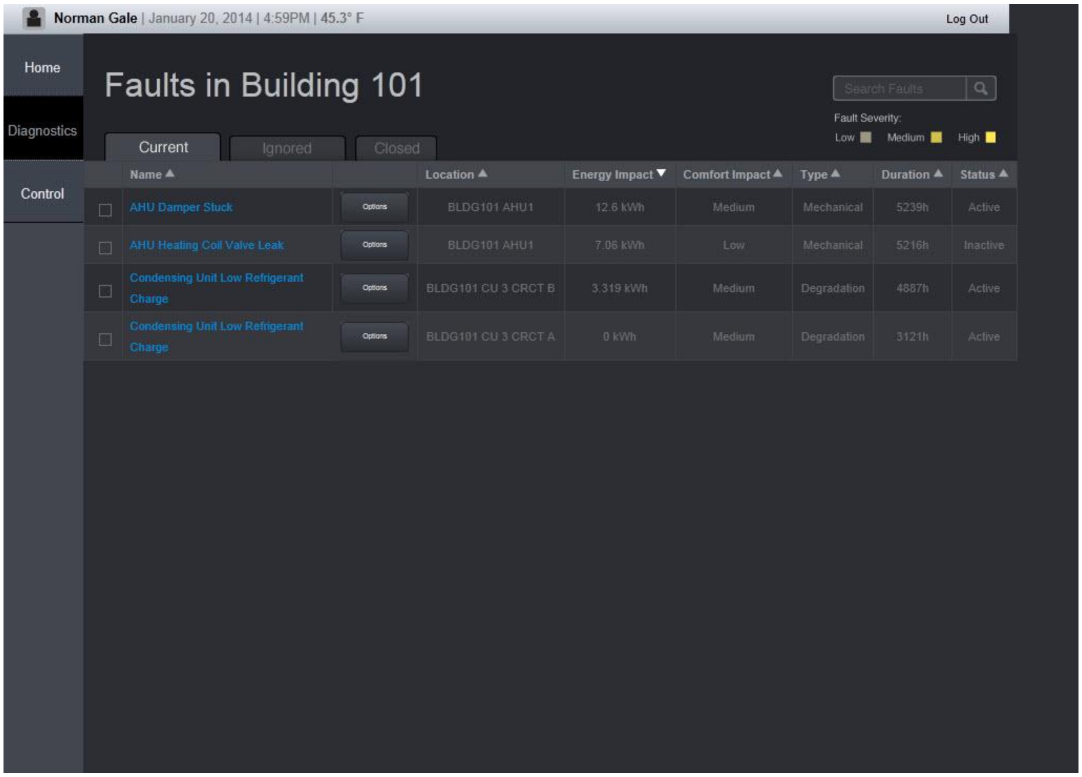
4.3 Fault Detail Page: (Figure 3 on page 7)

This page presents the information output for the advanced diagnostics algorithms. Some of this information is directly logged by sensors and available through the middleware described in Section 5. Other fields are calculated by third-party analytics and made available through the middleware. The major design aim of this page was to find a useful distillation of the massive quantity of potentially calculable statistics that would support our operators. Information such as location and type of fault is designed to support facility managers dispatching relevant technicians to the correct location. Explanation text provided by the diagnostic algorithm can communicate the reasoning behind the fault determination, which helps contribute to the algorithm's transparency. If active diagnostics are plausible for this fault, they can be initiated from this screen and the results are fed back into the algorithmic system.

4.4 Zone Control and Energy Evaluation: (Figure 4 on page 7)

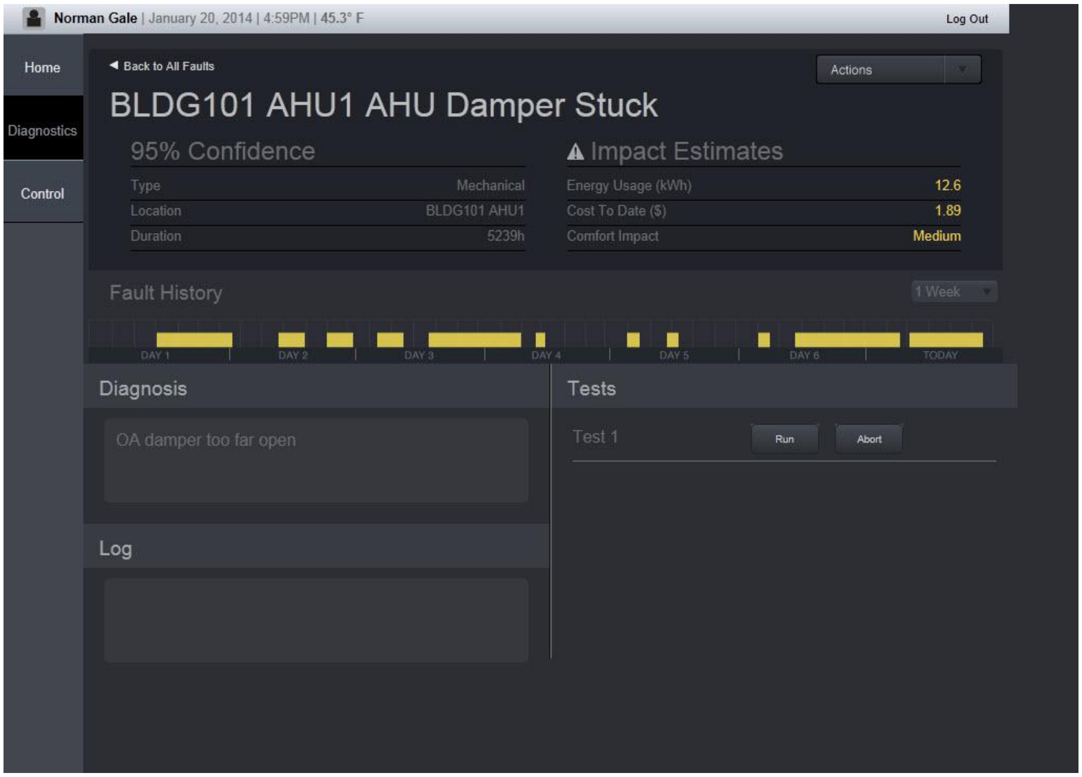
This tab presents the user with the ability to change zone set points and understand the consequences of doing so. As previously discussed, one significant area for improvement in energy efficiency is through better choices of set points. The definition of "better" is unique to each building, though in general maximizing comfort and minimizing expense are always positive. Since different users will have different priorities, for instance whether cost is more or less important than comfort, rather than attempt to automate the entire system, an exploratory system was designed. In this interface, users can try out a different window for low and high set points and receive an energy estimate based on historical data. If the user likes the result of a proposed pair of set points, the user can save the settings to the real building, or keep exploring more. This interface required a novel invention, an extremely fast estimation algorithm, which can provide an answer to the user in fractions of a second, rather than hours of simulation.

Figure 1: Overall Status Tab



	Name		Location	Energy Impact	Comfort Impact	Type	Duration	Status
☐	AHU Damper Stuck	Options	BLDG101 AHU1	12.6 kWh	Medium	Mechanical	5239h	Active
☐	AHU Heating Coil Valve Leak	Options	BLDG101 AHU1	7.06 kWh	Low	Mechanical	5216h	Inactive
☐	Condensing Unit Low Refrigerant Charge	Options	BLDG101 CU 3 CRCT B	3.319 kWh	Medium	Degradation	4887h	Active
☐	Condensing Unit Low Refrigerant Charge	Options	BLDG101 CU 3 CRCT A	0 kWh	Medium	Degradation	3121h	Active

Figure 2: Diagnostic Fault Tab



BLDG101 AHU1 AHU Damper Stuck

95% Confidence

Type	Mechanical
Location	BLDG101 AHU1
Duration	5239h

Impact Estimates

Energy Usage (kWh)	12.6
Cost To Date ($)	1.89
Comfort Impact	Medium

Fault History

Diagnosis

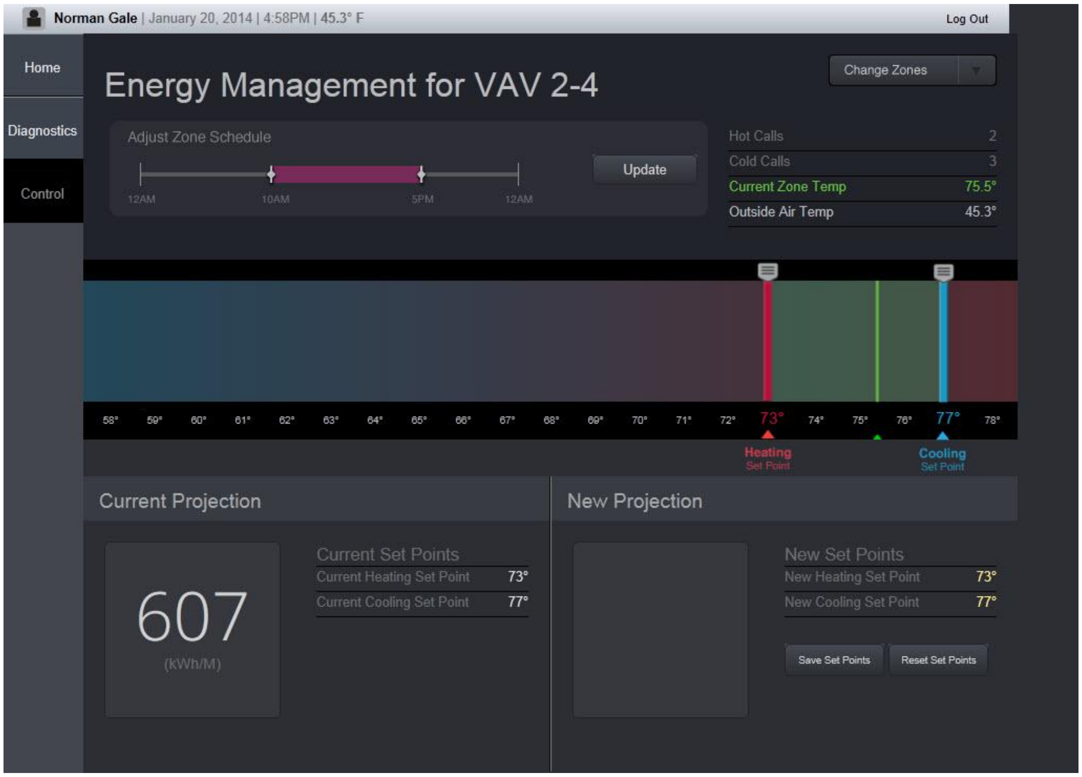
OA damper too far open

Tests

Test 1

Log

Figure 3: Fault Detail Page



Energy Management for VAV 2-4

Hot Calls	2
Cold Calls	3
Current Zone Temp	75.5°
Outside Air Temp	45.3°

Current Projection

607
(kWh/M)

Current Set Points

Current Heating Set Point	73°
Current Cooling Set Point	77°

New Projection

New Set Points

New Heating Set Point	73°
New Cooling Set Point	77°

Figure 4: Energy Management and Zone Control Tab

5. SYSTEM INTEGRATION

In order to implement this HMI as a functional system, an information architecture was implemented and deployed in several buildings to support testing. At each test site, a building management system, a handful of data-loggers, advanced control and diagnostics applications, and the HMI were deployed. An additional challenge was the fact that control and diagnostics application are created and operated remotely by different teams, limiting the flexibility of the system. Faced with this distributed design, the BMS and data-loggers were decoupled from the rest of the system by introducing a middleware layer. Every application accesses those systems through a common interface. A set of RESTful web services opens this interface to remote applications over HTTP on the internet or a local intranet.

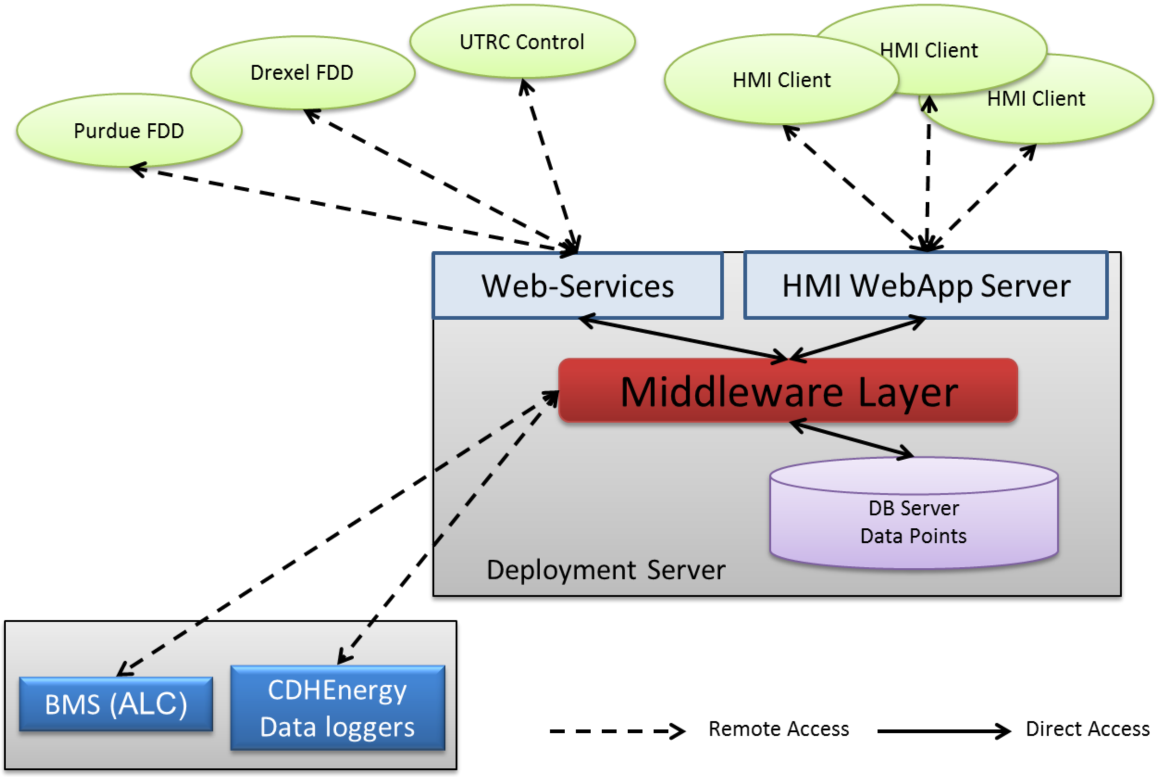


Figure 5: Simplified software architecture in a test building. The deployment server hosts the middleware layer, data web services, the HMI web service, and a database to store fault data computed by diagnostics applications. Remote applications get data through the middleware layer.

5.2 Service-Oriented Middleware Architecture

In the demonstration buildings a deployment server running a Linux operating system was hosted. Apache Tomcat acts as a web server and connects to a local PostgreSQL database server. Tomcat hosts the HMI web application and the web services. A database is used to keep track of the faults detected by the diagnostics algorithms running remotely. To simplify the integration effort, the teams involved in fault detection agreed on a fault data format that is general enough to capture the output of all the diagnostics algorithms. During normal operation a remote diagnostics application can retrieve both current and historical values of the data points in the building through the exposed web services. With the same interface it can also write the output of its analysis back into the database. Thanks to this architecture multiple diagnostics applications can interact asynchronously and simultaneously on the data with little or no coordination.

5.3 HMI Web Application

The HMI web application consists primarily of three components. The first component is an in-memory data cache containing all the dynamic content that can be visualized and produced by the client. This data cache is exposed externally through a RESTful interface. The second component is responsible of serving the static content of the web application, the html, CSS, and JavaScript code, which makes the website function in a browser. The third component is a background thread that implements the communication between the web application and the middleware. It periodically refreshes the data cache of the web application by pulling all the required data points either from the database, in the case of fault data, or from the buildings in the case of BMS data and data loggers. This same thread also computes other types of data that need to be visualized on the HMI and that are not directly available. The presence of the background thread and the data cache helps reduce the latency experienced by an operator when loading and interacting with the HMI. By limiting the number of read and write operations involving the database, the data cache increases the scalability of the number of HMI clients that can be served simultaneously.

This web application is built on open-source software libraries. In particular, Oracle Jersey is used to create a RESTful web service, and Google AngularJS is the templating engine used to produce the interactive components of the HMI.

5. CONCLUSION

This work presents a solution to some of the challenges presented by the next generation of building management technologies. The motivation for this research was to design a user interface that would enable a variety of new analytic technologies to be effectively deployed in a building. The user evaluation phase of the research produced a set of requirements that were satisfied by a novel design. That design was implemented as a website for evaluation and testing. A novel middleware was created to serve the data needs of this interface and analytic services. The resulting system was deployed on two buildings, Building 101 at the Philadelphia Navy Yard and at UTRC's high performance testbed building located at its East Hartford campus, and a variety of feedback sessions were held to gauge user responses.

Future work on this project, which is currently being conducted, includes a human subjects experiment designed to evaluate the quantitative performance improvements made possible by this interface. This work will provide a rigorous statistical basis upon which to judge the efficacy of this design in comparison to traditional BMS interfaces. Additionally, an instrumented long-term deployment at a building site with active use by facilities staff would provide a rigorous qualitative test for user acceptance and additional user feedback. User responses have thus far been positive and suggest we have succeeded at the initial goals. Due to the extremely small sample size of the available facilities staff on this project, and to protect their anonymity, specific comments are not presented in this publication.

The middleware implementation used in this work has proven itself extremely useful and capable of solving a wide variety of problems. Multiple buildings with different infrastructure, different sensors, and different facilities staff were integrated into a cohesive system interacting with the HMI.

REFERENCES

Katipamula, S., Brambley, M. 2005a. Methods for fault detection, diagnostics and prognostics for building systems-A review, Part I, *IJHVAC&R Research*, vol.11, no. 1: p. 3-25.

Katipamula, S., Brambley M., 2005b. Methods for fault detection, diagnostics and prognostics for building systems-A review, Part II, *IJHVAC&R Research*, vol.12, no. 2: p. 169-187.

Norman, D., 2013, *The Design of Everyday Things: Revised and Expanded Edition*, Basic Books, New York, 368 p.

Sarkar, S., Lovett, T., Bertuccelli, L., Vrabie, D., Krucinski, M., Mijanovic, S., 2013, From Smart Homes to Green Cities: Role of Intelligent Diagnostics and Control in Energy-Efficient Buildings, *Dynamic Systems & Control Magazine*, Special Issue December 2013.

Schein, J., Bushby, S., Castro, N., and House, J., 2006, A Rule-Based Fault Detection Method for Air Handling Units, *Energy and Buildings*, vol. 38, no. 12: p. 1485-1492.

ACKNOWLEDGEMENTS

We would like to express our gratitude the other teams who contributed to the success of this activity. In particular Prof. James Braun of Purdue University and Prof. Jin Wen of Drexel University, who contributed to the diagnostics components of this research. Prof. MaryAnn Sorensen Allacci, Prof. Clint Andrews, and Jennifer Senick of Rutgers University were instrumental in arranging the SME interactions with facilities staff and gathering user feedback.

This material is based upon work supported by the Energy Efficient Buildings Hub (EEB Hub), an energy innovation hub sponsored by the U.S. Department of Energy under Award Number DE-EE0004261.