BuildingSherlock: Fault Management Framework for HVAC Systems in Commercial Buildings

Hidetoshi Teraoka†, Bharathan Balaji†, Rizhen Zhang†, Anthony Nwokafor†,
Balakrishnan Narayanaswamy†, Yuvraj Agarwal†‡

†University of California, San Diego ‡Carnegie Mellon University
†{hteraoka, bbalaji, shiemi, anwokafor, muralib}@cs.ucsd.edu,
‡yuvraj.agarwal@cs.cmu.edu

ABSTRACT

HVAC systems constitute $\sim 40\%$ of energy in commercial buildings, and faults in HVAC account for 5% to 20% of its energy consumption. Typically, HVAC in modern buildings are managed using Building Managing Systems (BMS), and fault detection is one of the essential services provided by BMS to keep HVAC operational. We study the fault management practices followed in real commercial buildings, and find that current techniques used for fault detection fail to detect large number of efficiency related faults. Based on our findings, we developed BuildingSherlock (BDSherlock), which is a web service based fault management framework that exposes building information to automatically detect faults using useful algorithms. We deploy BDSherlock in a 145000 sqft building at UC San Diego, successfully detected 87 faults using data driven analysis.

1. INTRODUCTION

Modern buildings are complex entities, consisting of numerous subsystems including lighting, electrical loads, fire safety, security and Heating, Ventilation and Air Conditioning (HVAC). To control the energy footprint of a building, faults that occur in these systems need to be managed effectively. Typically, building managers rely on occupants to communicate any faults in the building. Typically, only the HVAC system is managed systematically due to its complexity and visibility to occupants. Even the relatively well managed HVAC systems, waste 5% to 20% of their energy due to faults [20]. As the complexity of building systems increases with automation, efficiency directives and the integration of renewable energy, management of faults from all these sub-systems in an organized manner will become even more critical in the near future.

Building faults occur due to various reasons – failure of equipment, sensor failure, mis-configuration by operator and misuse by occupants. We focus on faults in the HVAC system as it is the most complex component in buildings today, accounting for $\sim 40\%$ of building energy [6]. Typically, buildings use a Building Management System (BMS) for monitoring and control of HVAC, supported by sensors distributed across the building. To detect faults, most BMSes provide a fault analysis and reporting framework that analyzes available sensor data [1, 4]. In traditional BMSes, fault analysis is designed to help the facility managers to keep the HVAC operational, with efficiency as a much lower priority. Major faults such as fans not functioning, or inadequate cooling are reported by the BMS, but faults such as drop in efficiency of fans or a leaking valve are not reported. Over time, these minor faults accumulate, leading to significant energy wastage and equipment deterioration [16].

Commissioning of HVAC systems is encouraged as a standard practice to resolve the shortcomings of BMS based maintenance. The commissioning process considers the problem from a system perspective, and ensures best practices are followed in all phases of HVAC implementation - design, installation, configuration, operation and maintenance [16]. Commissioning can provide significant energy savings (16%), in addition to providing a number of additional benefits such as better indoor air quality with a return on investment of 1 to 4 years [16]. However, comprehensive commissioning is seldom done in practice as it requires extensive manual labour and has a high upfront cost. Automation techniques have been proposed to increase adoption of commissioning, and several solutions have been demonstrated both by industry and academia for different aspects of commissioning [3, 5, 13].

Although the Facilities Management (FM) in our university judiciously applies monitoring based commissioning on buildings which are energy inefficient, and has adopted several of the available automation solutions, it remains an ongoing challenge to manage faults in HVAC systems and keep them efficient. We study the fault management solutions proposed in literature and those provided by commercial products and investigate the fault management practices followed by four enterprise campuses in California, with an in-depth focus on the facilities at UC San Diego (Section 2). We find that modern solutions available for fault management have a number of limitations with the key observations as follows: (1) Although fault detection is automated, its coverage is not comprehensive, (2) The algorithms used for fault detection are limited by the framework, and more sophisticated algorithms suggested in literature are hard to implement, and (3) the information provided to the building managers about a fault is limited, which leads to significant manual time investment in diagnosing and fixing a fault.

To address the limitations in the current fault detection systems, we have designed BuildingSherlock (BDSherlock), an extensible, future proof fault management framework for buildings (Section 3). The goals of BDSherlock are to solve the problems faced in management of current HVAC systems, be extensible to other types of systems, and scalable to large and sophisticated deployments. We propose an open framework that would encourage collaboration between the stakeholders in building systems - facility managers, equipment vendors, service providers and building occupants. BDSherlock supports detection of complex faults, reports these faults to the building manager in an organized manner, and assists with swift diagnosis and fixing of faults. We advocate that fault management be a priority from the time building systems are designed, standardized data formats be used to encourage free sharing of information, automated analysis be deployed to detect faults comprehensively, and that these faults come with adequate contextual information for manual analysis.

We have designed BDSherlock to provide the essential ingredients to satisfy these set of objectives. BDSherlock builds on top of BuildingDepot [7], an open source web service based framework for building data storage. We have designed BDSherlock with two key principles in mind - (1) provide extensive HVAC information and allow reporting of faults through an open API so that sophisticated algorithms can be developed for fault detection, (2) provide relevant information to the facilities personnel to review these faults and act on them quickly. As part of BDSherlock, we have implemented a fault dashboard for facility managers where building information from different sources is organized in a single interface that provides a prioritized list of faults reported and tools for analyzing these faults. Finally, we have integrated several pieces of useful contextual information in BDSherlock including estimation of energy wastage due to a fault, and direct feedback from the occupants of the building to aid building managers in rapid exploration and fault remediation.

We implemented BDSherlock and deployed it in the Computer Science and Engineering building at UC San Diego, a 145000 sqft 5 floor building with 237 Variable Air Volume (VAV) units and one Air Handler Unit (AHU). We implemented algorithms to detect faults both at the VAV and the AHU level, and present our summary of findings here. Some of the faults we found were unique, and have not been found in commissioning studies conducted by the Facilities Management (FM) in other buildings on our campus. We captured a total of 87 faults.

2. CURRENT PRACTICES

To investigate current practices followed for fault management in modern buildings we interviewed facility personnel at four campuses - University of San Diego, San Diego County (SD County), University of California at Berkeley (UC Berkeley) and University of California at San Diego (UCSD). The personnel we interacted with have different positions within the facilities management - director of construction and maintenance, facility managers, facility energy managers, maintenance operators and building managers. A range of vendors were used in the buildings across these campuses, and providing us useful insight into the fault management services provided by Siemens [21], Johnson Controls [1], and Niagara AX [4]. As UCSD is our home campus, we were able to study the deployed Johnson Controls systems in depth across different buildings. We also followed the monitoring based commissioning process in a seven floor building undertaken by our facilities management group, and the implementation of continuous commissioning in two large buildings at UCSD. Furthermore, we present the background literature and related work in this section.

2.1 HVAC Background

HVAC systems in commercial buildings typically consists of a centralized component with heavy equipment and numerous terminal units distributed across the building. This component consists of chillers and boilers for cooling and heating water, heat exchangers and cooling towers to exchange cooled/heated water with air supplied to individual rooms, valves and dampers for modulation of water and air flow respectively, numerous other parts such as air filters, air compressor for pneumatic control and sensors distributed across the system. The conditioned air and water from the centralized units are distributed to the rest of the building using pipes and ducts to terminal units, which control the amount of air and its exact temperature let into a room. Terminal units have their own valves, dampers, fans and more sensors. Each of these parts fail over time due to regular wear and tear, improper use by occupants or incorrect configuration by maintenance staff. Examples of faults include a dirty filter, broken damper, leaking valve, fouled heat exchanger, improper schedule of HVAC, or placement of refrigerator in front of a thermostat. Typically, these faults can be detected automatically using sensor data due to rise in temperature in a room, excessive energy consumption and other symptoms. When a maintenance operator receives a notification of a fault, he/she analyses the sensor data to diagnose the underlying cause, or inspects the equipment manually. After a fault is confirmed manually, the operator fixes it if possible, or hires contractors to fix the fault.

2.2 Building Management Systems

Building Management Systems (BMS) in use today were designed to manage all aspects of HVAC system including control of equipment using configuration parameters, scheduling of operation, access control and the management of the underlying sensor network. Fault management was just another component typically focusing on detecting only faults that affect critical functionality and cause significant occupant discomfort. The data points in the BMS include sensor measurement, equipment status and control parameters such as setpoints, gain of PID control loop and schedule of operation. All the three BMSes we investigated provide the facility to report a fault when a sensor reading does not match its setpoint, or if equipment status is inconsistent with its control parameters [1, 23]. For example, the BMS raises an alarm when the temperature of discharge air exceeds a specified limit, or if the status of a fan reads Off when it is commanded to be On.

2.2.1 Sensors and Data Storage

The BMS fault detection system was not designed to capture the faults that indicatd a drop in equipment efficiency or deterioration of air quality. Hence, the sensors that can provide valuable information to uncover these faults are often not installed. For example, discharge air temperature sensors at the terminal units can be used to detect faults such as valve leakage or broken damper, but are not installed in most buildings at UCSD or UC Berkeley.

Most BMSes do provide data trending functionality, but trending has to be initiated manually for each sensor in both Johnson Controls and Siemens systems. Personnel who install and maintain the data logging facility avoid trending all sensors to reduce data deluge, assuming that it would take up too much disk space. Upon inquiry, we found that the size of sensor data logs across all the buildings at UCSD is less than 20 GB, which is quite trivial. Data collection is a key feature that needs to be enabled by default in BMS for analysis of performance and detection of faults. Data storage facility is now being provided by default in newer BMS and fault frameworks such as Niagara AX and SkySpark [3].

2.2.2 Fault Detection

In all the three BMSes we examined, the fault rules analyze data from a single sensor, and faults which occur across different sensors or due to lack of sensors are not captured by the BMS. In the Metasys system installed at UCSD, the rules fail to detect energy faults such as leaking valves or a fouled heat exchanger, and can only be detected by manual analysis of sensor data. The threshold for these rule based fault detection are set based on prior experience or using ASHRAE standards. We observed that several false positives occur due to minor sensor drifts or due to temporary spikes in buildings at UCSD, and it was difficult to change the threshold for the corresponding rule to reduce these false alarms.

Some of the modern HVAC components, such as the Computer Room Air Conditioning (CRAC) unit in our building testbed, come with automated fault detection facility built into the firmware. These equipment report faults such as high temperature, or inadequate water flow as a data point in the BMS. Such fault reporting is effective in determining faults with high accuracy, and can be used by all complex equipment to ease fault detection. We discuss this option further in Section 2.5.

2.2.3 Information Management

BMSes have been well designed to provide sensor data and equipment control parameters to the user. However, associated building information is often not integrated into the BMS, making it hard to use. At UCSD, facility personnel have different sources of information for sensor data, building power meter readings, building architectural plans and occupant complaint information. Such diverse set of sources make it hard for operators to analyze data, and considerable amount of time is spent on trying to figure out the underlying meaning behind the information provided. The naming convention, even for sensors installed by the same vendor, is not consistent across buildings, making it harder for operators to analyze data. For example, if the temperature reading in a room is high, the operator has to look up the architectural plans to figure out what type of room it is, and if it needs immediate attention. Such problems are addressed by newer versions of BMSes which provide graphical information connecting all these information. Vendor agnostic frameworks such as Niagara AX are providing support for standardizing naming system such as Haystack [2], and integrate information from sensors, power meters and building plans. We saw this implemented in practice across eight buildings maintained by SD County. Occupant feedback, however, remains separate from the BMS in all these systems.

2.2.4 User Interface

The user interface provided by the BMS also makes it hard to resolve faults quickly. In the Metasys system used at UC San Diego, an alarm is raised each time a sensor violates its threshold. This leads to a deluge of alarms for the same type of fault, and does not necessarily add information. The system is also designed so that there is a pop-up for every alarm raised, making the user experience further annoying. The alarms are provided a priority, and are listed separately so that operators can figure out which faults need to be addressed first. However, the faults are assigned static priorities, with higher priority for more critical faults. The operator maintains a separate mapping from types of faults to priority levels. The combination of these factors along with the limitations mentioned earlier make it hard for the operators to diagnose and fix even trivial faults. A fluid and intuitive user interface would help increase the productivity of operators significantly.

2.3 Human Element

Fault management system interacts with all the stakeholders in a building directly or indirectly. Faults remain undetected, and sometimes even caused by humans because of lack of information and related socio-economic issues.

2.3.1 Occupant Issues

Occupants of a commercial building often have little understanding of how HVAC system works, and do not know which of their actions will cause faults or energy wastage. In most cases the interface between occupants and buildings takes the form of a thermostat which measures temperature, and provides a dial to adjust its setpoint. Occupants at UCSD are often unaware of the HVAC schedule, and sometimes cannot even access the thermostat if the HVAC zone consists of multiple rooms.

Usage of space heaters during winter, which wastes cooling power and increases plug load power, is a well known issue caused by occupants [9]. Further, the thermostats get blocked by furniture, or worse, by a heat generating appliance such as a refrigerator which gives false feedback to terminal units, and leads to excessive cooling. Fume hoods are used in labs due to exhaust requirements, and consume significant energy to maintain the airflow. Fume hoods left open even after their required usage period is another problem which causes considerable wastage. Occupants sometimes change the room usage model, for example, by converting an office space to a machine room. As the HVAC system is designed for a different usage, the high cooling demand from such a room leads to overcooling of adjacent offices and wastes energy.

Occupants also report faults to the building managers when they notice them. The facilities management at UC Berkeley, SD County and UCSD rely on these reports to detect faults at the terminal units as they do not have the manpower to analyze all the terminal units in buildings. Typically, a work order is generated for these faults, and the issue is tracked until it is fixed. However, the occupant feedback system is maintained separately from BMS information reducing its effectiveness.

2.3.2 Understaffed

Many of the problems mentioned above can be overcome if sufficient manpower is available to maintain the HVAC system. Of the four campuses we talked to, understaffing was a common complaint. There are some buildings at both UC San Diego and UC Berkeley for which there are no dedicated building level managers, and no one has examined the building HVAC for several years. When the maintenance operators feel overwhelmed by the number of alarms they need to address, they first focus on critical faults and complaints given by occupants. Thus, faults that are further down the list often get ignored for extended periods of time. Further, operators often make temporary fixes which resolves the issue, but it leads to energy wastage or allows faults to resurface again later.

2.3.3 Vendor Lock-in

Typically, entire HVAC system is setup by one vendor, from equipment to software installation. Standardized building network protocols such as BACnet and LonTalk are being adopted in the industry to support interoperability. We saw a practical example when BACnet compatible thermostats (University of San Diego, UCSD) and fume hoods (UCSD) from different vendors were installed in practice. But BACnet usage by each vendor is different, and they depend on proprietary tags for a variety of operations, causing confusion during such integration efforts. Such vendor lock-in causes further problems when a fault occurs, and manual troubleshooting is required. Vendors do not publish detailed datasheets for their products, as is common in the electronics industry. Instead, they rely on vendor supplied service contracts to fix faults. Furthermore, while training is offered it comes at an additional cost by the vendor. Lack of publicly available technical information on the equipment makes troubleshooting difficult.

2.4 Commissioning

Commissioning of HVAC systems was introduced to mitigate the shortcomings of BMS based maintenance. Commissioning process involves a thorough investigation of the HVAC system to detect faults, upgarde inefficient equipment, use of efficient control mechanisms, and benchmarking of performance with measured energy data. Mills et al. [16] studied the commissioning across 224 builings, and report *median* energy savings of 16% with return on investment of 1 to 4 years, and numerous other benefits such as better indoor air quality and increased longevity of equipment. LEED certification system and California Energy Commission recommends commissioning as a standard process for buildings.

Commissioning requires significant manual labour and time investment, and hence, commissioning in practice focuses on certain aspects of building such as comfort, energy efficiency or security [16]. In the efficiency focused commissioning in UCSD buildings, the commissioning process targets the centralized portion of the HVAC as it contains all the heavy equipment, and improvement in their efficiency yields the highest return on investment. We followed the commissioning of the seven floor Calit2 builing at UCSD. The energy consumption of the building was measured for three months for benchmarking. This was followed by pre-functional testing, where the commissioning contractors manually inspect the equipment, test for miscalibrated sensors and leaking/broken valves or dampers. The faults found were fixed and equipment upgrades suggested were also made at this time. The contractor then proceeds with functional testing, where the data points from BMS are analyzed to test the HVAC in various modes of operation. A second round of fixing of faults occurs, mostly by tuning the programming system. The energy consumption of the HVAC is again measured for three months and compared against the baseline consumption to report energy savings results.

Significant energy savings are typically obtained as a result of such commissioning processes, but several aspects of the process can be improved to reduce manual labour and increase energy savings. As the process is focused on the central units of HVAC, the faults associated with terminal units are neglected. Although real measured data is used to isolate faults and provide its impact on energy savings, the analysis is done manually. Such manual analysis increases the time to analyze faults, the entire fault analysis phase took about six months in the building we observed on our campus.

Continuous commissioning is recommended for buildings which are already commissioned to monitor the building performance, and further tune its performance if possible. Continuous commissioning was applied to two recently commissioned buildings at UCSD - Giesel Library and Pacific Hall. A fault framework was provided by the service contractors to analyze the datapoints in these two buildings, and detect any faults that were left undetected. Analysis of data revealed that several faults that were fixed had resurfaced due to impairment of recently installed transducers and a valve, providing evidence that continous commissioning is valuable to ensure efficient operation of HVAC. The fault framework provided a dashboard to the building managers to visualize datapoints, and allowed them to benchmark performance across time easily. Data storage was also provided as a service to analyse sensor data. However, the analysis to detect any faults and evaluate their impact was still manual, as the automated detection methods used were rudimentary and led to false positives. The dashboard also didn't provide contextual information about the building itself, but only focused on sensor data.

2.5 Automation Solutions

Automated commissioning has been proposed in literature to decrease upfront cost, manual labour and increase adoption of commissioning practices [25]. There are three parts to automated commissioning - information management, functionality testing and fault detection with diagnosis. Prior work has focused on information management [15] and automated fault detection [18], and relatively few works have focused on automated functionality testing [13]. To the best of our knowledge, no work in literature focuses on supporting all three at the same time.

Information management focuses on integration of data from different sources, standard naming conventions, long term data storage, and visualizations to analyzing these data easily [19]. None of these focus on fault management in particular, and only provide support for limited fault detection. Niagara AX is a commercial product which has been designed with scalable information management, and implements several solutions proposed in literature, resolving many of the limitations of current BMSes. Niagara AX is vendor agnostic, allows integration of information across various sources, provides data storage services, and extendable functionality with Java based plug-in support. With the help of plug-ins support has been added to provide personalization, standard naming convention using Haystack [2], and researchers have integrated a separate fault management plugin as well [5]. However, the support for fault analysis and associated alarms are simplistic, and relies on rules which monitor a single sensor similar to other BMSes [23].

One of the solutions to automate fault detection is to integrate detection into the firmware of complex equipment, and the faults be reported as other data points so that existing BMSes would be able to easily report these faults [5]. Embedding the detection mechanism in the firmware allows usage of contextual information that is not available at the BMS level, integrate domain knowledge and complex algorithms within the equipment itself. However, the fault management system cannot rely on just these reports to detect all types of faults in the HVAC. If the fault were to occur in the embedded firmware itself, then the entire mechanism can fail. It is harder to change the firmware once programmed, and the onus is on the vendor to get all the parameters of fault detection correct at design time itself. This can be hard for complex equipment such as variable speed drives, which can fail due to many reasons [11]. Further, system level faults caused by misconfiguration or at the point of interaction between equipment would be missed out.

Several commercial fault management frameworks are available, and we examine SkySpark as a typical example [22]. SkySpark supports integration across data sources with compatibility with various standards - gbXML, oBIX, and external information such as utility data. SkySpark supports fault analysis with their own custom designed Axon language, that allows users to write sophisticated rules using available library functions and detects faults by executing these rules in their engine. Although this broadens the type of faults that can be detected considerably, the framework restricts the complexity of algorithms as they need to be executed within the Spark framework. Further, data analysts cannot use popular programming tools - Matlab, R, Python, along with their vast collection of available libraries. The APIs exposed are restricted for visualization, and does not include fault management. The framework we propose in this paper, BDSherlock, builds on top these automation solutions, and mitigates their limitations by advocating an open framework and flexibility in implementation.

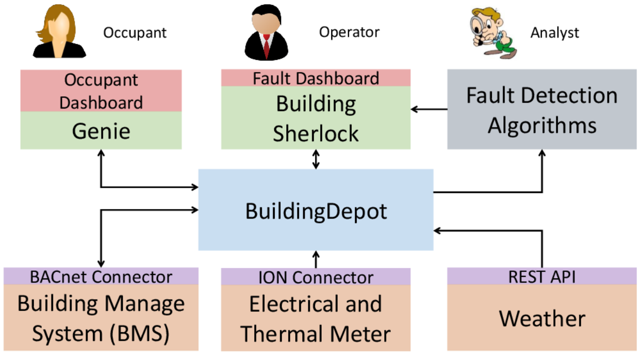

Figure 1: System Architecture of BuildingSherlock

3. DESIGN AND IMPLEMENTATION

A fault management framework is primarily geared towards reducing the effort to detect, diagnose and fix faults. Thus, the main objectives of a fault management framework are to reduce the time spent by an operator to analyze and fix faults, to provide assistance to the operator while she/he is investigating a fault, to increase their productivity, and finally, to reduce the knowledge required before the operator can investigate a fault. Apart from these usability requirements, the framework should be extensible to other kinds of systems, and scalable to manage a large number of faults and its associated data. We have designed BDSherlock to satisfy these objectives by emphasizing building information management, flexibility of algorithm implementation and providing contextual information about faults to facility personnel.

3.1 Software Architecture

BDSherlock has been designed as a web service, as it provides flexibility, scalability, and ease of development. Figure 1 depicts the software architecture of BDSherlock. We have built BDSherlock on top of BuildingDepot (BDDepot), an open source building data storage framework. BDDepot exposes RESTful APIs for storage and retrieval of sensor data, associated metadata, and provides access control [7]. Data from different sources are collected in BDDepot and it acts as our information management service.

BDSherlock is itself a web service, exposing a RESTful APIs for fault management. The APIs allow algorithm developers to register and report faults, as well as provide building information for development of user interfaces. Fault algorithms are third party applications which access data from BDDepot, analyze the data for faults, and report detected faults to BDSherlock. Genie is another web service which serves the needs of the occupants of the building by providing information about the current status of the HVAC, and allows them to send feedback [8].

Each webservice in the architecture - BDDepot, BDSherlock, Genie - is implemented using the nginx web server, uWSGI and Python Flask framework. BDDepot uses Cassandra databases to store sensor data, with redundancy and backup to increase reliability and performance. BDDepot also uses MySQL to store sensor metadata and other information such as authentication data. The details of BDDepot implementation is provided in our prior work [24]. BDSherlock uses MySQL database and Genie uses redis key value database to store local information.

3.2 Support for FDD Algorithms

BDSherlock aims to provide as much information to third party developers as possible to enable implementation of sophisticated fault detection techniques [14], and increase their accuracy and efficacy. We not only seek to provide long term sensor data, but include contextual information such as the type of HVAC system, the area covered by each zone, the usage model of a room, the model and make of a variable speed drive, and the expected efficiency of a cooling coil. We make such data accessible via BDDepot. We also provide authentication and access control to data using BDDepot services so that only trusted developers are provided access to data.

BDSherlock introduces a library of *fault types* to encourage usage of standard naming convention when reporting faults. Each fault type is associated with specific faults associated with parts of equipment or sensor. Examples include 'damper stuck', 'temperature high', and 'valve leak'. Each algorithm registers with BDSherlock before it can report faults with relevant information such as email for contact, type of faults reported, and data used for analysis. BDSherlock tracks the faults reported by each algorithm to provide insightful information to the facility personnel such as historical success of an algorithm, list of faults reported, etc.

3.3 Fault Reporting

The faults reported to BDSherlock should allow the facilities personnel to make a decision on which fault should be prioritized first, and also help with analysis of the fault itself. To prioritize faults, metrics such as return on investment and impact on indoor air quality provide the most relevant information. For analysis of faults, data used for detection of faults, and contextual information such as the equipment and location of fault will be helpful. Currently, we support reporting of equipment, its location, the severity of fault in percentage, the energy impact, relevant sensor data and a high level summary of the fault. We plan to include cost to fix faults, and hence determine return on investment to fix an energy related fault, a comfort index, the duration of the fault and possible causes of the fault.

All the faults reported by algorithms are grouped together if they indicate the same type of fault, equipment and location. Thus, the user can analyze reports by multiple algorithms for a single fault if needed, and it also reduces data deluge due to multiple reports by an algorithm. The history of faults reported for a particular fault is also provided. When a particular fault is marked as "fixed" by a user, the corresponding algorithms are notified via subscription mechanism, and are expected to start over their analysis for this fault from the time of fix.

3.4 Occupant Feedback

Occupants are provided limited information about HVAC in modern buildings. With the help of our Genie web service, we enable active occupant participation in keeping their environment comfortable, and report faults when they occur. Genie provides a web dashboard which reports the current status of the HVAC, the estimated power consumption of the HVAC zone, and historical sensor data. It also allows occupants limited control over their HVAC settings - change of temperature setpoint within a limited range, change of HVAC schedule and to report any feedback they may have regarding HVAC. We have over 140 registered occupants in Genie for our building testbed, and we routinely receive feedback on the thermal environment in different zones as well as reports of HVAC not functioning properly. We provide these feedback received in the

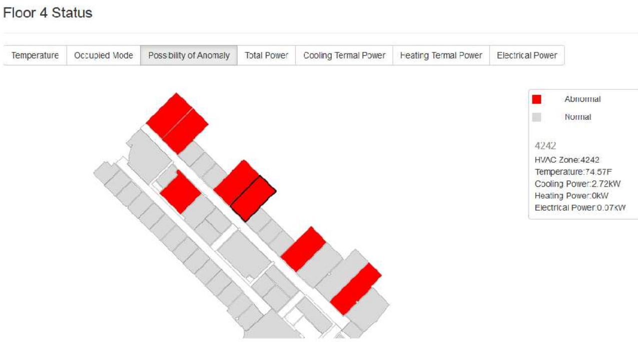

Figure 2: Floorplan view in BDSherlock Faults Dashboard.

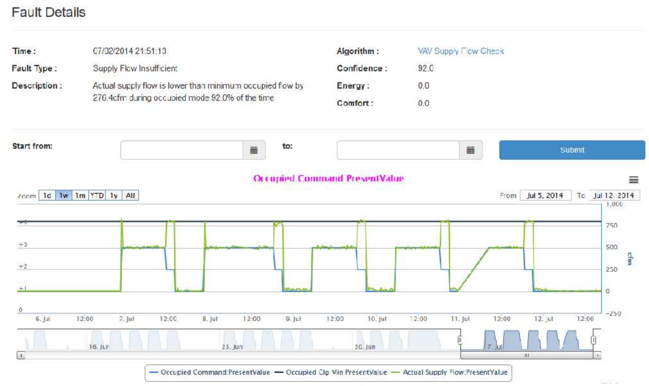

Figure 3: Fault report view in BDSherlock Faults Dashboard.

faults dashboard of BDSherlock. Further details on Genie has been described in prior work [8].

3.5 User Interface Design

The faults dashboard was designed to make it easy for building managers to check if there are faults. If there are multiple faults, the manager should be able to prioritize the faults, and act on the most important ones first. The front page of the UI shows an overview of the building, providing building power consumption information, and faults found in each part of the system - cooling system, heating system, computer room air conditioner (CRAC) unit, or VAV boxes located in each of the floors.

The user can click on each of these systems, and are provided with a visualization of the system. The visualization is a symbol diagram connecting different parts of the heating and cooling system, with symbols representing fans, pumps, cooling coils, sensors and how they are connected. The page shows live data being collected from BDDepot, and it is refreshed every minute. Each of the values can be clicked to get a trend of the value for the past one month. Any fault associated with the system is also provided in this view. For each of the floors, a floorplan view of the building is provided, with color graphs for different parameters - temperature, energy consumption, faults reported. Figure 2 shows the floorplan for one of the floors in our building testbed. Each of the zone in the floor is highlighted when moused over, and the important sensor values gets displayed in a sidebar. When the user clicks on this zone, a system diagram of the VAV box is presented with relevant sensor information on airflow. These diagrams are similar to Figures 4 and 5.

There is a faults tab which shows the complete list of faults being reported. It can be sorted by the system, subsystem, type of fault, confidence of fault reported, energy wastage, or time of reporting. The faults can also be searched based on system, subsystem, and fault type. Clicking on each fault provides the details of the fault as reported by the algorithm. Figure 3 shows a snapshot of this page for one of the faults reported. The details about the fault include the user facing information provided by the algorithm - summary of the fault, confidence level, type of fault, and it automatically plots the relevant sensors with the past one month's data.

4. DEPLOYMENT

4.1 HVAC System

Our building testbed is the Computer Science and Engineering building at UC San Diego, with an area of 145,000 sqft over five floors. The HVAC system of the building is maintained by the Facilities Management (FM) of our university, and is managed using Metasys BMS provided by Johnson Controls. The hot water and the cold water required for the HVAC is provided by a central co-generation plant to all the buildings on campus to save energy, and hence, the building HVAC system does not have a separate boiler or cooling tower in the building. The HVAC systems contains one Air Handler Unit (AHU) and 237 Variable Air Volume (VAV) boxes for zone level control. Figures 4 and 5 show the details of the AHU and a typical VAV box respectively. There is a separate Computer Room Air Conditioning (CRAC) unit which maintains the thermal environment in the server room.

The AHU uses cold water from the central plant to cool the supply air via cooling coils. The hot water from central plant is passed through heat exchangers and used for heated domestic water and hot water required for heating coils in the VAV boxes. The amount of hot and cold water used is regulated using valves, and the flow of water is maintained using variable speed pumps. The cold air generated from the cooling coils is supplied to the VAV boxes spread across the building using duct network, and the flow of air is maintained using supply and return fans. An economizer (not shown) modulates the appropriate amount of outside air to provide fresh air, and to save energy by partially cooling the return air when possible.

The VAV terminal boxes receive the cold supply air and hot water from the AHU. A damper is used to modulate the amount of air let into the zone, and a heating coil modulated by a valve is used to reheat the air when required. A temperature sensor installed in the thermostat located in the zone provides control feedback for maintaining the thermal environment. Some of the bigger zones have separate fans to maintain airflow, and zones such as kitchens and restrooms have dedicated exhaust fans to increase return air flow. Each zone can be maintained in three modes of operation - *occupied, standby, unoccupied*, and the zone is maintained within a temperature band of 4 °F, 8 °F and 12 °F respectively. The HVAC is maintained in a static schedule, with *occupied* mode from 6am to 6pm on weekdays, *standby* mode from 6pm to 10pm and *unoccupied* mode on nights and weekends.

There are about 4700 sensors located throughout the setup to monitor the condition of HVAC. However, the coverage of sensors is not comprehensive for fault analysis. As part of this project, we installed the mixed air temperature sensor in the air mixer to detect faults related to the economizer. The control of the HVAC system is a combination of both electronic and pneumatic signals. For the pneumatically controlled actuators, the control signal is converted

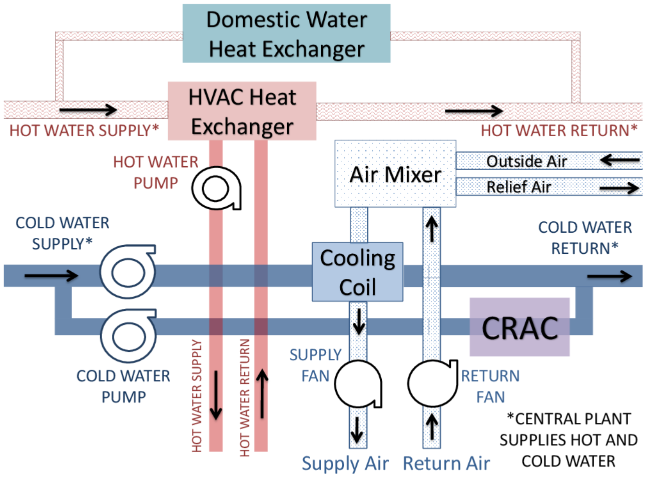

Figure 4: Centralized part of HVAC system showing details of air handler unit - cooling coils, heat exchanger, water pumps, supply/return fans, domestic water and CRAC units [8].

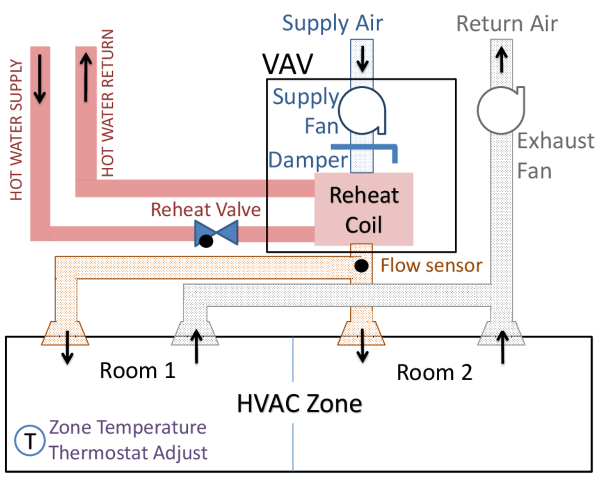

Figure 5: VAV with reheat system used for controlling the temperature of discharge air in each HVAC zone [8].

from the electronic signal sent by the BMS using electronics to pneumatic transducers.

4.2 Data Collection

We communicate with the BMS sensors and actuators (Section 4.1) using BACnet protocol. As BACnet by itself does not support authentication, our campus maintains a separate virtual network (VLAN) for BACnet related communication for security. We install our machine in this VLAN, and it acts as the connector between the BMS and BDDepot. For each writable point in BACnet, there is an associated priority table, and we are assigned a fixed priority by the Facilities Management for writing to these points. We leverage BDDepot for services such as event notifications, access control and resolving control conflicts to manage the HVAC system.

The power meter data at UCSD is maintained on a separate network, using a Schneider Electric proprietary protocol, called ION. We continuously poll the meter data using a separate connector. Our building testbed has specially instrumented meters, measuring the breakdown of electrical power from the HVAC, lighting, elevators and machine room. Further, we have sensors measuring the temperature and flow of hot and cold water used, allowing us to account for thermal power consumed by HVAC. Thus, we can accurately measure the savings obtained from fixing faults in our testbed. We also collect data from the local weather stations using APIs provided by Weather Underground. Genie, our occupant dashboard, has been in operation for over a year, and we have been collecting occupants thermal feedback and complaints over this period. We provide this information to both the algorithms and the fault dashboard.

To provide as much contextual information as possible to both the algorithms and HVAC operator, we studied the architectural plans of our building, and extracted useful information for analysis. In the Metasys BMS, only the sensor data and its associated metadata like units and operation mode were provided, and location data such as the room could be located by scrutinizing the names or certain proprietary BACnet data properties. For each VAV box, we manually extracted which rooms belong to a particular zone, in which of these rooms was the thermostat located, the area covered and the usage model (office, kitchen, restroom, etc.) of these rooms. We also extracted design specifications for each VAV box, corrected misnamed sensors, informed FM about missing sensors and zones. For instance, we found that no datapoints were provided for a large conference room in the building. Further, we created a graphical representation of the floor plans and the system diagrams of the VAV and AHU for our faults dashboard.

We have been collecting data from our testbed building since August 2013, and will release this dataset as part of this paper to encourage public sharing of information, and providing researchers with real data to apply innovative algorithms and detect faults. Note that we will exclude or anonymize any personally identifiable information.

4.3 Prefunctional Test

Before the available data can be analyzed for faults, we performed a sanity check using pre-functional testing. It is a standard procedure in the commissioning process, in which an operator manually verifies calibration of sensors, visually inspects HVAC equipment and fixes any faults noticed during this observation [16]. An experienced Johnson Controls contractor performed the pre-functional test for our building HVAC. Note that these tests are limited to the centralized part of the HVAC as it would take too much time to manually inspect each VAV box. This is the standard practice for pre-functional testing for any building on our campus, as it gives most energy savings for the time spent.

During the pre-functional test, the contractor found that some of the sensors as well as the hot water thermal meter were miscalibrated, and these were fixed. He also noticed that Electronic to Pneumatic (EP) transducers were broken for cooling coil valves, which essentially kept the valve in a fixed position leading to significant water leakage. This is a good example example of how a minor fault left undetected which costs less than 100 dollars to fix led to a massive energy penalty. Another observation was that the economizer damper was broken, letting 100% outside air into the air mixer irrespective of outdoor weather. This leads to substantial wastage of energy, but it has still not been fixed as it would take >$20,000 to replace it, and FM did not have the funds for making the fix. The control loop related to the economizer was also not working, and the contractor fixed this programming error as well. The discharge air temperature for the supply air was fixed at 55 °F as per design, and dependent on the cooling demand of the top three energy intensive zones in the building. The contractor changed it to be dynamic, with the setpoint determined by the average demand of all the zones. This changed the average supply air temperature to 65 °F. He made a similar change to the hot water supply temperature setpoint, reducing the average setpoint value from 180 °F to 120 °F. Both of these changes led to significant savings, and made the HVAC energy consumption more proportional to the demand.

4.4 Algorithms

4.4.1 Rules

Rule based fault detection is the most prevalent and well known way to detect faults in HVAC. Rules check a particular metric against a predefined threshold, and a fault is flagged when the threshold is violated. These type of rules are already implemented in modern BMSes, and the rules implemented is dependent on some standard rules specified by certification bodies and the domain knowledge of the operator. We implement similar rules, but overcome the fallacies of the traditional implementation using BDSherlock. We describe how we implement a generic rule based algorithm here.

To detect a particular fault, the algorithm first registers with BDSherlock and specifies the type of faults it will detect along with the list of sensor data it will use for its analysis. For the analysis itself, the algorithm reads the relevant data from BDDepot for a specified timerange, say a month, and stores the data locally for fast analysis. Depending on the logic for the rule and the threshold specified, the algorithm checks for violations of threshold across the duration, and reports the fault if the violation occurred for a significant period of time. The number of violations is coded as the *severity* of the fault, giving the frequently occurring fault a higher *severity*. While reporting the fault, the algorithm provides the specific location of the fault, such as the zone and the particular sensor which is defective. It also provides a human readable summary indicating under what situation the fault occurs, how much is the violation, and how severe the fault is. An example summary reported by an algorithm is "Actual supply flow is lower than minimum occupied flow by 276.4 cfm during occupied mode 92.0% of the time". The algorithm also provides any energy impact the fault has, so that operators can prioritize to fix faults which give most savings. Note that we deliberately use *conservative thresholds* to reduce false positives from these rules, as the FM was only interested in those faults which caused a large energy cost for extended period of time. Thus, the results we present do not have any errors in them, and the building HVAC likely has many more minor faults not reported by us.

As the algorithm looks for violations for significant periods of time, temporary spikes in the data are not flagged as faults. The duration across which the fault is analyzed depends on the type of fault. Faults which need to be fixed immediately, such as supply fans not operational, are checked for shorter duration, like 15 minutes, and, faults which are non-critical, and needs to provide high accuracy data, such as excess airflow is analyzed across a week. We implement rules based on those suggested in literature and our own observations. The complete list of rules implemented, along with the faults found and their energy impact is presented in Section 5.

4.4.2 Machine Learning

Many data driven algorithms have been proposed in the literature to automatically detect faults in building HVAC systems - see [14] for a detailed review. In parallel work [17], we implement and test a number of these methods. In particular we implemented subspace methods (exemplified by Principle Component Analysis or PCA), correlation based methods (based on the intuition behind Strip, Bind and Search [12]) and developed a new algorithm, Model, Cluster and Compare (MCC). These methods form exemplars of the range of possible FDD methods that our system makes feasible in practice. We do not include the faults detected using these methods in our results, as we focus on faults which cause significant energy wastage and reduce false alarms. A detailed analysis of these anomalies has been presented in our parallel work [17].

Rule	**System**	**# Instances**
Supply Flow Excess	VAV	6
Temperature Setpoint	VAV	27
Insufficient Flow	VAV	10
Thermostat Adjust	VAV	33
Insufficient Cooling	VAV	8
High Temperature	VAV	1
Economizer Damper Stuck	AHU	1
Total		87

Table 1: Summary of HVAC faults detected

5. EVALUATION

The goal of our project is to detect all the faults that may exist in our building HVAC using the data available. We implemented the rules for detection based on standard checks listed in literature [10], as well as our own rules based on anomalies detected from data driven analysis [17]. We present the list of rules implemented, the faults we found and their energy impact on the HVAC system in this section, and our results are summarized in Table 1.

5.1 VAV Faults

At the VAV controller level, we detect faults due to misconfiguration, insufficient/excessive airflow, cooling or heating, and any airflow leakage. We could not detect heating valve leakage due to lack of discharge air temperature sensors.

We were surprised to find a large number of configuration errors in the VAV controllers. In 27 zones, the VAV controllers did not change the temperature setpoints with changes in occupancy mode. This fault kept the HVAC operational in these zones even during nights and weekends, an extravagant waste of energy. We found a similar configuration anomaly in 6 zones, where the airflow setpoint was set to be high in unoccupied mode even when the zone did not require cooling. Note that there were certain zones which was configured in this manner by design, such as hallways, kitchens and restrooms, and we excluded these rooms by data available from design specifications. A particularly wasteful example of a conference room is shown in Figure 6, where the airflow setpoint is high enough to cool the room to heating temperature setpoint, which then led to heating coil usage to increase the temperature of the supplied air. Thus, simultaneous heating and cooling occurred in an unoccupied zone due to a configuration error. We estimate that the anomaly in this zone alone led to energy wastage of 6.4 MWh/yr. The Johnson Controls contractor we were working with surmised that these errors were probably caused due to misunderstanding at the time of initial building commissioning. As these type of faults are not expected to be present, and not mentioned even in standardized rule sets recommended by California Energy Commission [5, 10], they would have gone unrecognized even during a retro-commissioning process.

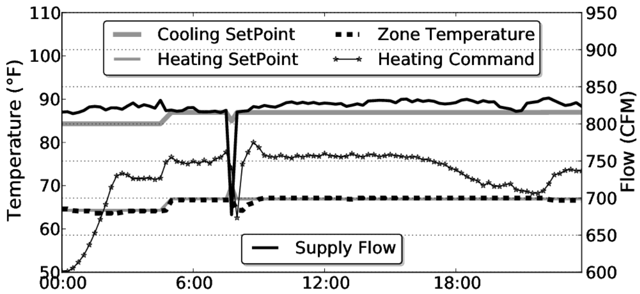

Figure 6: Supply flow excess in a conference room

Such faults are also not detected in any of the fault frameworks we have examined.

Another uncommon fault we found were associated with thermostat adjust available to occupants. By default, the thermostat allows occupants to change their temperature settings by $\pm1\,^{\circ}$F. If this change was found to be inadequate to keep a comfortable temperature within the zone, the maintenance operators increase their range upon a request from occupants. These changes in thermostat adjusts effectively shifted the temperature band maintained by HVAC in different occupancy modes (Section 4.1). Due to a flaw in the initial configuration of thermostats, these changes to thermostat adjusts remained in effect in unoccupied mode. For zones with large change in adjust, the temperature band shifted enough to require cooling or heating during unoccupied mode. We found 33 zones with adjusts greater than $\pm3\,^{\circ}$F. Unfortunately, now it is hard to change the programming to disable these adjusts in unoccupied mode using the BMS, and we had to leverage BDDepot to implement the fix ourselves. We skip the energy waste analysis for this fault, as it hard to estimate the energy consumed by these zones without such a shift in temperature band.

We also found several faults that would have been detected with a standard set of rules. The temperature in one of the kitchens was found to be particularly high despite maximum cooling by the VAV box. Upon inspection, we found a water cooler was placed right in front of the thermostat, which led to incorrect measurement. This is an example of a fault where the occupant is unaware of the implications of their actions on the HVAC system. We also found 8 zones to have insufficient airflow during the occupied mode, and our basement computer labs to require excessive amount of cooling. These faults occurred due to the recent change in static pressure settings during the Prefunctional Test (Section 4.3). Although this setting leads to energy savings, insufficient airflow can be a health hazard, and insufficient cooling will make occupants uncomfortable.

We are currently investigating other faults that may exist in the VAV controller such as short cycling, airflow leakage, misconfiguration of PID loop. We have also informed the FM of our findings, and in the process of fixing the faults reported.

5.2 AHU Faults

We have implemented the standard set of rules recommended in literature to detect faults at the AHU level [10]. These include checking economizer functionality, and leakage in heating/cooling coil valves. As there is no heating done at the AHU level in our building HVAC (Section 4.1), we do not apply any rules associated with heating. We confirmed with the help of mixed air temperature sensor installed at our request that the economizer dampers were indeed broken, and were 100% open all the time. The fault was detected by comparing the estimated mixed air temperature with the mixture of outside air and return air, and the actual measured mixed air temperature.

We found several sensors to be miscalibrated even after the prefunctional testing as they were not critical to the operation of HVAC - the return water temperature sensors in cooling coils, the airflow sensors in the supply and return fans (their readings are beyond maximum capacity of HVAC), and chilled water supply temperature sensor in the CRAC unit. We will further examine the efficiency of the variable speed drive fans and pumps once these sensors are calibrated by FM.

6. LIMITATIONS AND FUTURE WORK

We have implemented and deployed BDSherlock successfully in a large commercial building, and detected large number of faults which lead to wastage and discomfort. A number of improvements can be made on the framework proposed to further automate and ease analysis of faults in HVAC systems.

Functional testing is a useful method to detect many faults in HVAC system, in which an actuator is commanded to specific configurations to detect faults. For example, a valve is commanded to be shut off, the measurements are checked to see the impact of any leakage present. This method currently done manually, and can compensate for lack of sensors if automated. BDSherlock framework can be easily extended to support automated functional testing, as it is written on top of BDDepot, which supports authentical, access control, and controlling of actuators. However, careful policies and mechanisms need to be developed to avoid damage to HVAC equipment by third party applications.

Another extension BDSherlock which would maintenance operators is to integrate the fault management framework with the rest of the BMS functionality. Further, more contextual data about a particular fault would help in both analysis and prioritizing faults. Integration of energy simulation frameworks such as eQuest or EnergyPlus would help accurately predict the energy wastage due to a fault. A database of cost associated with fixing each fault would help facility managers take decisions based on return on investment to fix a fault. Further, a comfort index associated with faults can help maintenance operators gauge the impact of a fault on occupants of the building.

We have not focused on many of the transient faults that occur in HVAC systems. These transient faults also cause significant discomfort and damage equipment, but is overlooked by facilities as it focuses on more critical faults. Automation control techniques need to be developed that would detect and compensate for these faults dynamically. Thus, the facilities personnel can focus on the faults which cause long term problems.

7. CONCLUSION

We have studied the fault management practices followed across four enterprise campuses, with an in-depth focus on the facilities in UC San Diego. We show that modern practices fail to capture many faults prevalent in the HVAC system due to limitations in the management framework, and approach towards fault detection. Based on our findings, we propose a list of requirements for fault management systems. Our suggestions include integration of information sources, long term data storage, standardized naming conventions, support for wide variety of fault detection algorithms, tools for analysis of faults, and reporting of contextual information with fault detection.

We designed and implemented BuildingSherlock (BDSherlock), a web service based fault management framework which exposes RESTful APIs for reporting of faults by third party algorithms. We have deployed BDSherlock in the Computer Science building at UC San Diego, and successfully detected 87 number of faults in the HVAC system.

8. REFERENCES

[1] Johnson Controls Metasys. `http://www.johnsoncontrols.com/metasys`.

[2] Project Haystack. `http://project-haystack.org/`.

[3] SkySpark by SkyFoundry. `http://www.skyfoundry.com/skyspark/`.

[4] Tridium Niagara AX. `http://www.niagaraax.com/`.

[5] Advanced automated hvac fault detection and diagnosis commercialization program. *Energy Research and Development Division Final Project Report*, 2008.

[6] Buildings Energy Databook(DoE). *Energy Efficiency & Renewable Energy Department*, 2011.

[7] Y. Agarwal, R. Gupta, D. Komaki, and T. Weng. Buildingdepot: an extensible and distributed architecture for building data storage, access and sharing. In *BuildSys*. ACM, 2012.

[8] B. Balaji, H. Teraoka, R. Gupta, and Y. Agarwal. ZonePAC: Zonal Power Estimation and Control via HVAC Metering and Occupant Feedback. In *BuildSys*. ACM, 2013.

[9] A. Beltran, V. L. Erickson, and A. E. Cerpa. ThermoSense: Occupancy Thermal Based Sensing for HVAC Control. In *BuildSys*. ACM, 2013.

[10] C. E. Commission et al. Nonresidential compliance manual, 2005.

[11] F. Filippetti, G. Franceschini, C. Tassoni, and P. Vas. Recent developments of induction motor drives fault diagnosis using ai techniques. *Industrial Electronics, IEEE Transactions on*, 2000.

[12] R. Fontugne, J. Ortiz, N. Tremblay, P. Borgnat, P. Flandrin, K. Fukuda, D. Culler, and H. Esaki. Strip, bind, and search: a method for identifying abnormal energy consumption in buildings. In *IPSN*. ACM, 2013.

[13] P. Haves and M. Kim. Model-Based Automated Functional Testing - Methodology and Application to Air-Handling Units. 2005.

[14] S. Katipamula and M. R. Brambley. Review article: methods for fault detection, diagnostics, and prognostics for building systemsŮa review, part i. *HVAC&R Research*, 2005.

[15] L. Luskay. Identifying building design information necessary for commissioning and proper system operation (rp-1032). *Transactions-ASHRAE*, 2003.

[16] E. Mills. Building commissioning: A golden opportunity for reducing energy costs and greenhouse-gas emissions. *Lawrence Berkeley National Laboratory*, 2010.

[17] B. Narayanaswamy, B. Balaji, R. Gupta, and Y. Agarwal. Data Driven Investigation of Faults in HVAC Systems with Model, Cluster and Compare (MCC). In *Submission to BuildSys*. ACM, 2014.

[18] L. K. Norford, J. A. Wright, R. A. Buswell, D. Luo, C. J. Klaassen, and A. Suby. Demonstration of fault detection and diagnosis methods for air-handling units. *HVAC&R Research*, 2002.

[19] M. A. Piette, S. K. Kinney, and P. Haves. Analysis of an information monitoring and diagnostic system to improve building operations. *Energy and Buildings*, 2001.

[20] K. Roth, D. Westphalen, M. Feng, P. Llana, and L. Quartararo. Energy impact of commercial building controls and performance diagnostics: market characterization, energy impact of building faults and energy savings potential. *Prepared by TAIX LLC for the US Department of Energy. November. 412pp (Table 2–1)*, 2005.

[21] Siemens. Apogee building automation. 2008.

[22] SkyFoundry. Skyspark analytic rules: Combining a comprehensive library of analytic functions with full programmability. Feb 2014.

[23] Tridium. Niagara ax user guide. May 2007.

[24] T. Weng, A. Nwokafor, and Y. Agarwal. BuildingDepot 2.0: An Integrated Management System for Building Analysis and Control. In *BuildSys*. ACM, 2013.

[25] F. Xiao and S. Wang. Progress and methodologies of lifecycle commissioning of HVAC systems to enhance building sustainability. *Renewable and Sustainable Energy Reviews*, 2009.