Demo Abstract – BuildingSherlock: Fault Management Framework for HVAC Systems

Rizhen Zhang[†], Bharathan Balaji[†], Yan Zhang[†],
Balakrishnan Narayanaswamy[†], Yuvraj Agarwal[†‡]

[†]University of California, San Diego [‡]Carnegie Mellon University
[†]{shiemi, bbalaji, yaz074, 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. The current techniques used for fault detection fail to detect large number of efficiency related faults. 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 demonstrate faults using data driven analysis.

1. INTRODUCTION

Faults in modern buildings are common and account for 5% - 20% of their energy consumption [4]. Todays faults management systems only capture a subset of faults, and detect high level symptoms without insight to its underlying cause. Commissioning of buildings are used to address these limitations [3], but it is a labour intensive, expensive and time consuming process. Hence, commissioning is only applied to building systems which are energy intensive or critical for operation. As the complexity of building systems increases with automation and efficiency directives, management of faults will become a critical component to keep buildings operational and energy efficient.

The Facilities Management (FM) in our university judiciously applies monitoring based commissioning on buildings which are energy inefficient, and has adopted several automation solutions, but it remains a challenge to manage faults in buildings. Modern solutions available for fault management have several limitations: (1) Coverage of the detection system is not comprehensive, (2) The algorithms used for fault detection are limited by the framework, and sophisticated algorithms suggested in literature are hard to implement, and (3) the information provided to the building managers about a fault is limited, which increases the time to diagnose and fix a fault.

To address the limitations in the current systems, we have designed BuildingSherlock (BDSherlock), an extensible, future proof fault management framework for buildings. We propose an open framework that encourages collaboration among the stakeholders - 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. BDSherlock builds on top of BuildingDepot [1], an open source web service 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 implemented, (2) provide relevant information to the building managers to review these faults and act on them quickly.

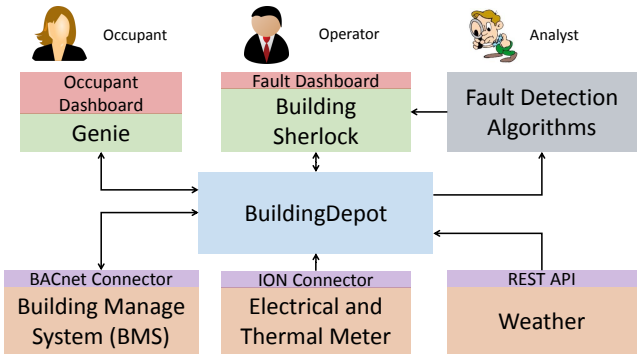

Figure 1: System Architecture of BuildingSherlock

As part of BDSherlock, we have implemented a fault dashboard for building managers where building information from different sources is organized in a single interface that provides a prioritized list of faults and tools for analyzing these faults. In our demonstration, we will present the fault dashboard for a 145,000 sq ft building at UC San Diego.

2. DESIGN AND IMPLEMENTATION

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. BuildingDepot (BDDepot), our building data storage framework, exposes RESTful APIs for storage and retrieval of sensor data, associated metadata, and provides access control [1].

The APIs provided by BDSherlock allow algorithm developers to register and report faults, as well as provide building information for development of UI. Fault algorithms are third party applications which access data from BDDepot, analyze the data for faults, and report detected faults to BDSherlock. Genie is a web service which serves the occupants by providing information about the current status of the HVAC, and allowing them to send feedback [2].

2.1 Support for FDD Algorithms

BDSherlock provides historical sensor data, includes contextual information such as the type of HVAC system, different equipment

installed, and how they connect with each other. We make these accessible via BDDepot by studying the architectural diagrams of our testbed building. We provide authentication and access control to data using BDDepot services so that only trusted developers get access to data.

BDSherlock introduces *fault types* to encourage use of standard naming convention when reporting faults. 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.

2.2 Fault Reporting

The faults reported to BDSherlock helps the building manager to prioritize faults, and analyze each fault. For prioritization, we report equipment type, its location, the severity of fault, relevant sensor data and a summary of the fault.

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, and it also reduces data deluge due to multiple reports by an algorithm. The history of faults reported for a particular fault is provided. When a particular fault is marked as "fixed" by a user, the corresponding algorithms are notified via subscription mechanism, and start over the analysis for this fault from the time of fix.

2.3 User Interface Design

The faults dashboard was designed to make it easy for building managers to check if faults exist. 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, or VAV boxes located in each zone.

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 plot of the value for the past one month. For each of the floors, a floorplan view of the building is provided, with color graphs for different parameters - temperature, energy consumption, faults reported.

There is a faults tab which shows the list of faults reported. Faults can be sorted by parameters, and searched based on system, and fault type. Clicking on each fault provides the details of the fault as reported by the algorithm. Our demonstration will include this website, with live data streaming in from our building testbed, and analysis of faults detected thus far.

Genie is our web service which connects the occupants of the building with the HVAC system. It supplements the thermostat information with temperature measurements and status of HVAC. The user can also change the temperature settings of their zone, activate the HVAC on weekends/nights and send thermal feedback or complaints to the building manager. User feedback is shown in BDSherlock to assist building managers with their analysis of faults. We will include the Genie as part of our demonstration.

3. CONCLUSION

Modern practices fail to capture many faults prevalent in the HVAC system due to limitations in the management framework, and approach towards fault detection. We propose a fault management framework which includes 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 a five floor, 145000 sqft building in UC San Diego, and successfully detected 87 faults in the HVAC system. Our demonstration includes the BDSherlock web application along with the web interface developed for the occupants of the building.

4. REFERENCES

[1] 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.

[2] 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.

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

[4] 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.