An Expert System for Building Energy Management through the Web of Things

Daniel Ibaseta[1[0000−0001−6069−6781]], Julio Molleda[2[0000−0002−2949−1195]], Martín Álvarez[1[0000−0002−1750−0955]], and Fidel Díez[1[0000−0003−4523−1577]]

[1] CTIC Technological Centre, Gijón, Spain
{daniel.ibaseta,martin.alvarez,fidel.diez}@fundacionctic.org
[2] Department of Computer Science and Engineering, University of Oviedo, Spain
jmolleda@uniovi.es

Abstract. Managing energy consumption in buildings is of utmost importance given the fact that 20% of the total energy consumed worldwide comes from the buildings sector. The miniaturization of electronic and mechanical systems, together with low-power wireless communications, facilitate the development and deployment of building energy management systems (BEMS) based on Internet of Things (IoT) platforms. It is well known that IoT solutions create silos suffering from interoperability issues. In this paper, we propose an expert system based on the W3C Web of Things (WoT), a paradigm that seeks to counter the interoperability issues in the IoT. The proposed system implements a set of rules, fed by a time series database, that mange several sensors and actuators addressed through Web technologies using WoT. The goal of this expert system is to be the core of a BEMS, and thus, it will be able to optimize energy consumption as well as to enable smart retrofit of existing buildings. In addition, the expert system is combined with a graphical user interface featuring several web-based dashboards, making it possible for an administrator to remotely supervise the operation of the whole building.

Keywords: Smart building, Energy management, Expert system, Internet of Things (IoT), Web of Things (WoT)

1 Introduction

In recent times, monitoring, managing and optimizing energy consumption is of utmost importance in almost every sector, such as industrial, transportation, and residential, among others. According to the Department of Energy of the U.S. Energy Information Administration, energy consumed in the buildings sector (including both residential and commercial) accounts for 20.1% of the total delivered energy consumed worldwide in 2018, and it is expected to reach 22% in 2050 [9]. Monitoring and managing building energy consumption require a wide variety of interconnected sensors and actuators to be placed or integrated into the different building appliances (the air conditioner, the building hot water pump, to name but a few). These sensors and actuators are the key devices of the Building Energy Management System (BEMS).

The Internet of Things (IoT) has proven its usefulness in the last years in the energy monitoring domain, as well as in energy management and saving. However, platforms are usually based on dedicated software and tightly coupled services and hardware, creating silo solutions and fragmentation. The Web of Things (WoT) is a paradigm devised by the World Wide Web Consortium (W3C) on top of the IoT concept focussed on countering the interoperability issues in the IoT. It extends the Web architecture to incorporate smart devices into the Web. In IoT, each element is referred to as a *Thing*, whereas in WoT each element is referred to as a *Web Thing*.

The W3C WoT paradigm is based on four building blocks, as shown in Figure 1: *i*) Thing, the abstraction of a physical or virtual entity represented in IoT applications; *ii*) Thing Description, the structured data (using a descriptive JSON file) that augments a Thing providing metadata about the Thing, its interactions, data model, communication and security; *iii*) Binding Templates, the collection of communication metadata that explains how to interact with different IoT platforms; and *iv*) Scripting API, an optional building block that eases the development of IoT applications by providing a runtime system for IoT applications similar to a Web browser.

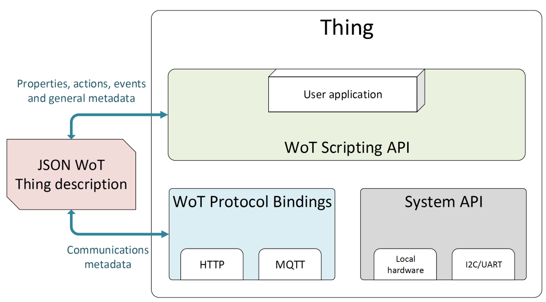

Fig. 1. WoT architecture concept

WoT offers an interesting approach to solve the issues faced during the design stage of systems that include legacy devices and heterogeneous technologies. The main differences between IoT and WoT are:

- The counteracting of fragmentation between different sensors and actuators by providing a machine-interpretable description [2].
- WoT brings IoT with machine-readable descriptions of sensors, actuators and controllers [2], providing it with an interoperability layer and enabling semantic interpretation of systems.

– WoT is designed to work specifically with the World Wide Web protocols, avoiding the technology fragmentation and lack of interoperability between different families of devices.

In the last decade, several systems have been developed to monitor and control building appliances based on IoT or Web technologies. However, only a few integrate all the available data to offer an intelligent system to minimize energy consumption and maximize comfort. In [4], a decentralized system for energy efficiency for monitored buildings is proposed. This system, called CONDE, focuses on decentralization for system efficiency and scalability. In [6], another decentralized solution is proposed. This solution is based on electricity demand, input power from solar panels and current pricing of grid energy to decide when to buy or sell energy to the grid and manage all the energy demand efficiently using a decision engine. In [3], a platform is defined using indoor air parameters (such as temperature and humidity) compared with outdoor temperature and energy price to optimize energy consumption. In [8], the solution is based on presence. Using CO_2 and infrared sensors, the system can decide when to set the temperature to a comfortable range or to turn off room appliances to save energy. Another example of a rule engine is [7], which provides good performance and maintains a good level of comfort using comfort thresholds, adequately capturing dependencies and interactions between components.

In this paper we propose a rule-based expert system for building energy monitoring and management using several sensors and actuators addressed through Web technologies based on the W3C WoT paradigm. The context of this work is the HEART Project [1], which aims to design a tool that integrates multiple components of a building, enabling smart retrofit. In this project, a complete building energy system (indoor air ventilation system, energy system, and water system) is monitored and controlled by a group of smart devices.

2 Expert system for building energy management

The decision-making ability of a human expert can be emulated in computer systems through an expert system [5]. The system proposed in this work is composed of: *i*) a WoT proxy that handles all the deployed devices in the building; *ii*) a database that stores all the data gathered from the devices; *iii*) a building device manager that reads the data from the WoT proxy; *iv*) a device monitor for visualization; and *v*) the BEMS engine that implements the custom logic. Figure 2 shows the architecture of the decision engine of the expert system proposed in this work.

The custom logic is a combination of real-time data and a set of facts and rules. These facts, together with the current values acquired from the environment sensors, conform the input data of a proposed set of rules. As a consequence, the result provided by the custom logic actuate on some building devices (i.e. modify a thermostat, switch on or off a fan, start pumping water in a tank, etc.). Figure 3 shows a control-flow diagram from which the set of rules of the expert system is derived.

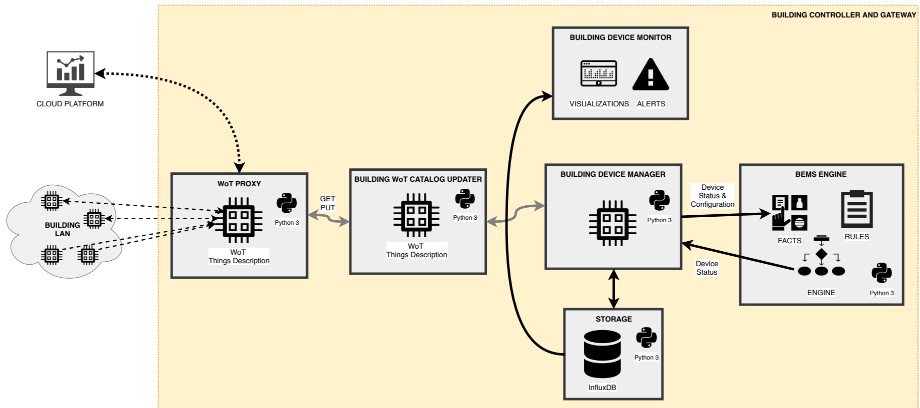

Fig. 2. Decision engine architecture

The rules implemented in the proposed expert system respond to the requirements of the HEART Project [1] and are focused only for testing and prototyping the final Artificial Intelligence solution that will be implemented in the project. Figure 4 shows an example of a rule implemented in Python. These rules consist of two parts: *i*) The first part, where the input variables are declared, and the comparisons and thresholds are defined; and *ii*) a function which involves the "@Rule" previously defined and stores the result in the time series database.

The proposed system is built upon InfluxDB[3], an open source time series database designed to handle high write and query loads. This database provides real-time querying, as well as write and query capabilities through a built-in HTTP API.

The proposed expert system controls the status of the thermal storage devices, such as domestic hot water (DHW) tanks, the main water storage, and the phase change material (PCM) tank, where the main building heat pump stores the heated water. The status of these devices varies depending on specific environmental parameters, such as the temperature of water in each tank, and on some predefined values that set some specific thresholds for the rule engine.

In the proposed architecture, all components are deployed using Docker, and orchestrated using Docker Compose. The BEMS software module implements the rule engine to manage the energy flow of the system. This component is based on Experta[4], a Python library for building expert systems, which enables the development of expert systems based on a heuristic solution. This library is inspired on CLIPS[5] (C Language Integrated Production System), a rule-based programming language developed at the NASA's Johnson Space Center, released in 1996 as open source.

---

[3] https://www.influxdata.com/products/influxdb-overview/
[4] https://pypi.org/project/experta/
[5] http://www.clipsrules.net

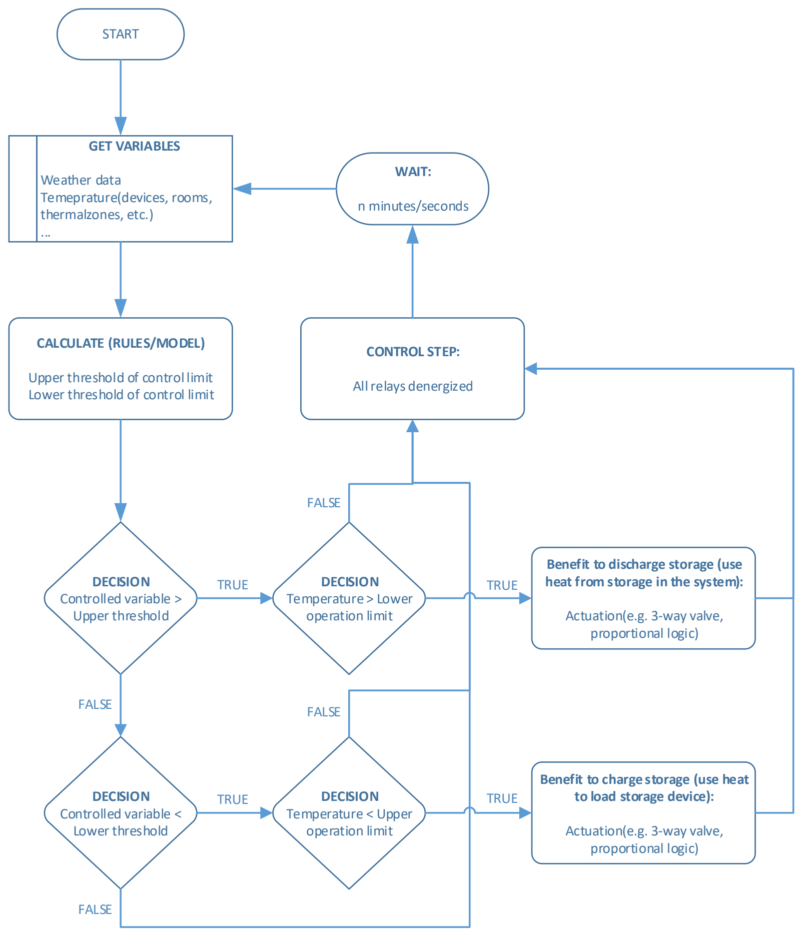

Fig. 3. Decision engine control-flow

```python
# PCM STORAGE DISCHARGING
@Rule(
    AS.storageDeviceStatus << PcmStorageStatusFact(
        id=MATCH.idDevice,
        value=MATCH.currentValue,
        status=NE('discharging')),
    PcmStorageConfigurationFact(
        id=MATCH.idDevice,
        upper_threshold=MATCH.upperThreshold,
        lower_threshold=MATCH.lowerThreshold,
        upper_operation_limit=MATCH.upperOperationLimit,
        lower_operation_limit=MATCH.lowerOperationLimit),
    TEST(lambda upperThreshold, currentValue : currentValue > upperThreshold),
    TEST(lambda lowerOperationLimit, currentValue : currentValue > lowerOperationLimit),
    salience=0)

def storage_discharge(self, idDevice, storageDeviceStatus, currentValue, upperThreshold, lowerThreshold):
    """Discharge storage of device, if the current value is greater than the upper threshold (also than the lower)"""
    print('PCM STORAGE `{0}`: LOWER [{1}] UPPER [{2}] CURRENT [{3}] ACTION -> [{4}]'.format(idDevice, lowerThreshold, upperThreshold, currentValue, 'DISCHARGE'))
    self.dataManager.saveStatus(PcmStorageStatus(id=idDevice, value=currentValue, status='discharging'))
    self.modify(storageDeviceStatus, status='discharging')
```

Fig. 4. Rule implementation example

The software is deployed on an UP Squared board, an x86 single board computer with an Intel®Celeron™N3350 processor, 8GB of RAM, one Wi-Fi and two Gigabit Ethernet interfaces. The board runs the Ubuntu Server 18.04.3 LTS operating system. It covers the requirements of the system, allowing to connect it to multiple networks simultaneously with enough bandwidth to attend all the requests. It continuously gathers information of the status of the devices in the building and exposes visual data through intuitive dashboards.

3 Experiments and results

The rule engine in the BEMS is executed periodically (the time interval can be fixed from seconds to minutes), gathering the latest values from the time series database and updating the status and parameters of the system on the same database accordingly.

The BEMS graphical user interface is composed of several dashboards. The behaviour of the overall system is visualized using time series and intuitive representations. An example of such visualizations can be seen in Figure 5, which represents when each device is being charged, discharged or idle. In addition, some information about the level of water storage in each device is also shown. Data shown in Figure 5 has been simulated using a script that fluctuates PCM Storage temperature to go below and above the defined thresholds periodically, so the charge/discharge cycle can be observed. This cycle is arbitrated by the expert system and Figure 5 shows that the temperature is not allowed to go down or up certain thresholds.

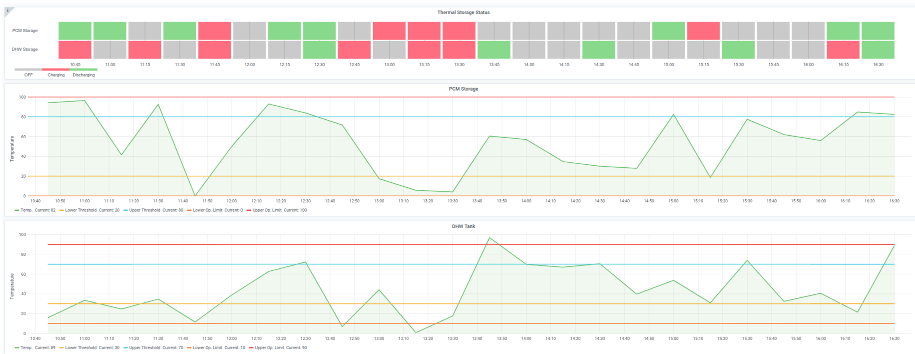

Fig. 5. PCM storage state visualization

Using another dashboard, the user can visualize the status of the fan coils, which depends on the thermal energy involved in the thermal energy exchange. Figure 6 shows the status of the fan coils with their properties. In this dashboard, the floor and fan coil ID can be filtered to show individual fan coils.

Usually, the number of hot water tanks ranges between 1 and 10, depending of the building, whereas the number of fan coils can grow up to several hundreds. Thus, a performance test was run using the proposed architecture to ensure that it can properly handle several hundreds of fan coils. The performance test consists on adding simulated fan coils to the BEMS. Table 1 shows the elapsed time to run the rule set and the RAM memory used by the UP Squared Board depending on the number of fan coils attached to the system.

# of fan coils	Rule processing time (s)	RAM Memory used (MB)
50	2	140
150	4	150
500	7	160
1000	14	180
2000	28	210

Table 1. Performance test, fan coils

4 Conclusions and future work

Building energy management is the key to building energy saving. Intelligent systems can be used to optimize the energy consumption on such environments.

Fig. 6. Fan coil visualization

In this paper we propose a rule-based expert system for building energy management based on the W3C recommendations for the Web of Things paradigm.

The proposed system is designed to be included in the HEART Project [1]. Simulations run in the system using live values from a time series database show the behavioural of the final building energy management system. The dashboards implemented in the graphical user interface make it possible for an administrator to supervise the operation of the whole building.

Future work will involve the adaptation of the expert system to the cloud, along with an extension for other sensors, actuators and devices. In addition, the system will be extended to be able to manage more than one building.

References

1. Heart Project, https://heartproject.eu/, [accessed 3 Jan 2020]
2. Web of Things at W3C, https://www.w3.org/WoT/, [accessed 4 Dec 2019]
3. Ain, Q.U., Iqbal, S., Khan, S., Malik, A., Ahmad, I., Javaid, N.: IoT Operating System Based Fuzzy Inference System for Home Energy Management System in Smart Buildings. Sensors **18**(9), 2802 (Aug 2018). https://doi.org/10.3390/s18092802
4. Farias, C., Pirmez, L., Delicato, F.C., Soares, H., dos Santos, I.L., Carmo, L.F.: A Control and Decision System for Smart Buildings. In: 2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing. pp. 254–261 (Dec 2013). https://doi.org/10.1109/UIC-ATC.2013.108
5. Jackson, P.: Introduction to Expert Systems, 3rd Ed. Addison-Wesley (1999)

6. Li, W., Logenthiran, T., Phan, V.T., Woo, W.L.: Intelligent housing development building management system (HDBMS) for optimized electricity bills. In: 2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I CPS Europe). pp. 1–6 (Jun 2017). https://doi.org/10.1109/EEEIC.2017.7977410
7. Merabet, G.H., Essaaidi, M., Brak, M.E., Benhaddou, D.: Agent Based for Comfort Control in Smart Building. In: 2017 International Renewable and Sustainable Energy Conference (IRSEC). pp. 1–4 (Dec 2017). https://doi.org/10.1109/IRSEC.2017.8477369
8. Sun, J., Zhang, Y.: Towards an Energy Efficient Architecture in Smart Building. In: 2015 International Conference on Computational Intelligence and Communication Networks (CICN). pp. 1589–1592 (Dec 2015). https://doi.org/10.1109/CICN.2015.302
9. U.S. Department of Energy: International Energy Outlook 2019 with projections to 2050. https://www.eia.gov/ieo (2019), [Online; accessed 10 Jan 2020]

Acknowledgments

This work has been funded by the European Union's Horizon 2020 Framework Program for Research and Innovation under grant agreement No. 768921 and by the University of Oviedo under project 2018/00061/012.