Available online at www.sciencedirect.com

ScienceDirect

Energy Procedia 153 (2018) 295–302

Energy Procedia

www.elsevier.com/locate/procedia

5th International Conference on Energy and Environment Research, ICEER 2018

A Control Methodology for Building Energy Management Systems (BEMS) in Heat Networks with Distributed Generation

Sean Rhys Jones[a]*, John Beardmore[b], Mark Gillott[a], Rabah Boukhanouf[a], Gavin Walker[c]

[a]Department of Architecture and Built Environment, University of Nottingham, Nottingham, NG7 2RX, UK
[b]T4 Sustainability Ltd, Ilkeston, DE7 8EF, UK
[c]Energy Technologies Building, University of Nottingham, Nottingham, NG7 2RX, UK

Abstract

Reducing building energy consumption is important to achieve sustainable development, as a result, there is a need to investigate better energy systems with well-designed management infrastructure. At the Creative Energy Homes, a low-temperature heat network with distributed generation links seven properties. The aim of the system is to investigate the efficiency benefits of low-temperature heating, while at the same time testing the prosumer concept. For homeowners to be prosumers, they can buy and sell heat to a network. This system is the first of its kind, and as a result required bespoke Building Energy Management Systems (BEMS). This paper focuses on the hardware, software, and system operation used as part of project SCENIC (Smart Controlled Energy Networks Integrated in Communities). The project utilises a simple and cheap hardware configuration involving relays, IO boards and RaspberryPi microcomputers. An open-source Building energy Management System (oBeMS) platform is used for monitoring and control.

Keywords: open-source Building energy Management Systems; Building Energy Management Systems; Distributed Generation; Low-temperature District Heating; Control Systems.

* Corresponding author. Sean Rhys Jones. Tel.: +44 115 74 84842
E-mail address: sean.jones@nottingham.ac.uk

10.1016/j.egypro.2018.10.011

1. Introduction

The energy consumption of residential buildings has increased mainly due to heating, ventilation and air conditioning (HVAC) systems, and user behaviour. According to the International Energy Agency (IEA), currently residential and commercial buildings account for up to 32% of total final energy consumption [1]. As energy consumption through the burning of fossil fuels is understood to contribute to climate change, it is imperative to secure clean and efficient energy systems for buildings. The requirements for improving energy consumption are three-fold. We require affordable energy, sustainable energy, and better energy management infrastructure. The latter involves optimising generation and distribution linked to use in buildings [1]. The umbrella term for energy management equipment in buildings is Building Energy Management Systems (BEMS).

The main objectives of BEMS are to increase occupant satisfaction by creating a comfortable and productive environment, and to reduce energy consumption, operational costs and environmental impacts including carbon emissions [2-5]. Additionally, BEMS have monitoring and communications facilities, may save staff time, and offer commissioning benefits [6]. BEMS are split into two categories:

- Predictive energy management
- Real-time control algorithms

Predictive energy management includes forecasting and updated information for optimisation [7]. Whereas, real-time controls are generally used in electrical systems for optimal scheduling of domestic appliances [7]. This paper describes a bespoke real-time control platform used in Project SCENIC (Smart Controlled Energy Networks Integrated in Communities) with the aim of developing predictive management in the future. SCENIC acts as a pilot study for the much larger – Project SCENe (Sustainable Community Energy Networks) looking at accelerating the adoption of community energy systems.

Project SCENIC links seven homes at the University of Nottingham's "Creative Energy Homes" test site in a community low-temperature heat network. Each of the seven homes has distributed generation, and all expect one has the ability to supply energy to a central thermal store. As a result, home occupiers can become prosumers as well as consumers - where they are both buyers and sellers energy to a double loop network (see Figure 1). This system is the first of its kind to combine distributed generation within community heat networks.

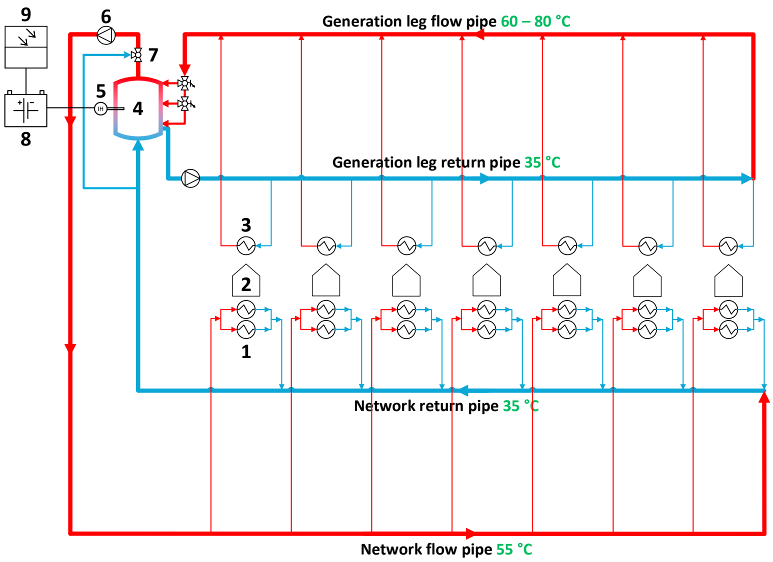

Fig. 1. Project SCENIC community heat network with distributed generation. 1 – Consumer Heat Interface Unit (HIU) with heat exchangers for space heating (SH) and domestic hot water (DHW); 2 – Building; 3 – Prosumer HIU for output to the network; 4 – Central thermal store; 5 – Immersion heater; 6 – Pump; 7 – Blending vlave; 8 – Battery; 9 – Distributed solar PV.

To avoid limiting the scope project SCENIC an “open-source Building energy Management System” (oBeMS) is used to provide a software framework and support hardware [8]. Monitoring and control of building plant is done by wireless sensors and a network of simple microcomputers. Data and control decisions are stored digitally for code debugging and system optimisation. Software server modules are used to provide sensor interfaces used by client applications carrying out high-level monitoring and decision making. oBeMS also comes with a customisable web user interface for space heating control. For a previous example of a design using oBeMS see [9] on rapid deployment modular building. This system is offered under the Mozilla Public License Version 2.0 (T4 Sustainability Ltd., 2013, Mozilla Public License Version 2.0).

In order to control equipment and facilitate prosumer participation, an integrated approach to sensing, metering, data logging, user interface, decision making and control functions is required. Using a RaspberryPi (RPi) microcontroller and direct digital control (DDC) of solid state relays allows for flexibility as the control algorithms can be expressed and modified in software without physical rewiring.

This paper describes the hard- and soft-ware, system operation, and the control strategies chosen to optimise the buildings distributed generation in a community low-temperature district heat (LTDH) network. Thanks to the flexibility of oBeMS, each buildings control strategy can be programmed to help understand the optimal equipment control regime. The results from each strategy will be used to understand the most economically and environmentally advantageous operation plan. Once heat generation and distribution have been optimised, further work will develop the optimisation between heat and electrical power.

2. Hardware

Within this section, the network of hardware in the system is described. Figure 2 illustrates the basic hardware setup throughout the site.

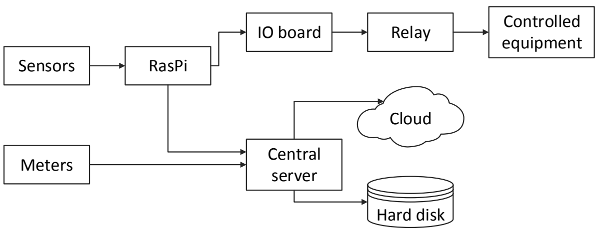

Fig. 2. Hardware setup with data and command flow

A key component of the overall hardware design is the control box in each house where the RPi, supporting hardware board, and solid state relays are connected (see Figure 3). From these boxes the control algorithms are run, and equipment control decisions are made.

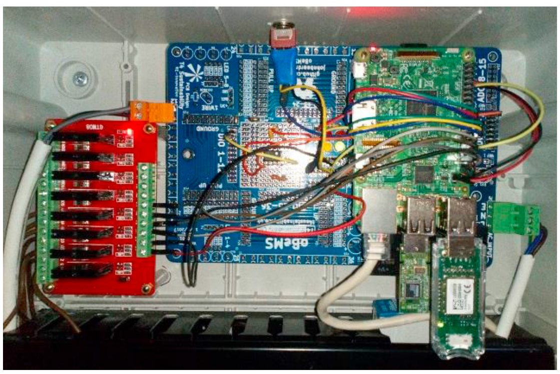

Fig. 3. 8 channel relays (left – red), supporting hardware oBeMS IO board (centre – blue), RPi microcomputer (left – green)

2.1. Sensors and meters

In this project, most sensors transmit data using the EnOcean communication protocol which can be received by RPi EnOcean USB receivers. Two, however, use the support hardware of the oBeMS project, using 10 bit A to D inputs to read analogue sensors which send temperature data as a voltage at 10mV per degree centigrade.

Heat energy data is collected with the MBus communication protocol in a private network of wireless and wired meters, MBus receivers and an MBus master data logger.

2.2. IO board, microcomputer and central server

The oBeMS board is used as a way of connecting the RPi to real-world wiring. The board also comes with a flexible open source software package that provides a framework within which to implement experimental control systems.

The RaspberryPi 3 Model B Vi.2 (RPi) provides the distributed processing power for the control system. The oBeMS support hardware has many essential features for the project, particularly A to D and D to A converters, a number of power supply options and a hardware watchdog timer.

A Linux server using the Ubuntu LTS (Long Term Support) operating system is used for storing and processing data. The central server is connected to each RPi through the University of Nottingham IP network, and to each building in the system via an M-Bus master.

2.3. Controlled equipment

To optimise the system, heat generation must be scheduled. As a result, there is auxiliary equipment such as, Heat Interface Units (HIU), pumps, and valves that must be controlled to enable heat flow. It is important to note; there are two types of HIU used in the system, pro- and consumer units. The HIUs allow for heat exchange between the network and buildings. Consumer HIUs allow space heating and domestic hot water production in the buildings, whereas prosumer HIUs allow heat generators to supply heat to the network, and ultimately to the thermals store (see Figure 1).

The valves contained within the consumer HIU's operate automatically when a 230V electrical signal is sent to the unit. However, as community prosumer participation has never been done before, the HIU is bespoke and required additional control for heat export. It was necessary to retrofit an additional Belimo LR230A valve to the unit that prevents unnecessary network-side flow.

The heat generating technologies are distributed throughout the network. The specification of each is described in Table 1.

Table 1. Distributed heat generators supply energy to the thermal store

Heat output equipment	Model
Solar thermal evacuated tube	Solar-Lux
Solar thermal flat plate	auroTHERM
Solar thermal evacuated tube	Navitron 47mm
Biomass boiler	Okofen Pellematic PES
Condensing gas boiler	Eco Elite 24 System ErP
Condensing gas boiler	Worchester Greenstar Ri
Immersion heater	Howden WR613 12 kW

The system controls incorporates a variety of pumps including those in the HIUs, solar thermal flow stations, gas and biomass boiler pumps, and large heat network pumps used to circulate water around the prosumer and consumer loops.

3. System Operation

3.1. Initiation

The control decisions are informed by four temperatures sensors. All control functions can be derived from data received from these four sensors:

- Room temperature sensors
- DHW cylinder or central thermal store temperature sensor
- Solar thermal collector temperature sensor

In addition, the control code takes into consideration whether the system is in island or local mode (where individual buildings are not connected to the heat network), or whether the system is in heat main mode. This information is derived from a switch on the control box.

Finally, a call for heat is derived from a web-based user interface designed to schedule heating within each house.

3.2. Action

The key control code can be broken down into heat network mode and local mode. If there is a call for heating and we are in local mode, the pro- and consumer HIU's are closed. If we are in heat main mode a more complex control strategy is required (see Figure 4).

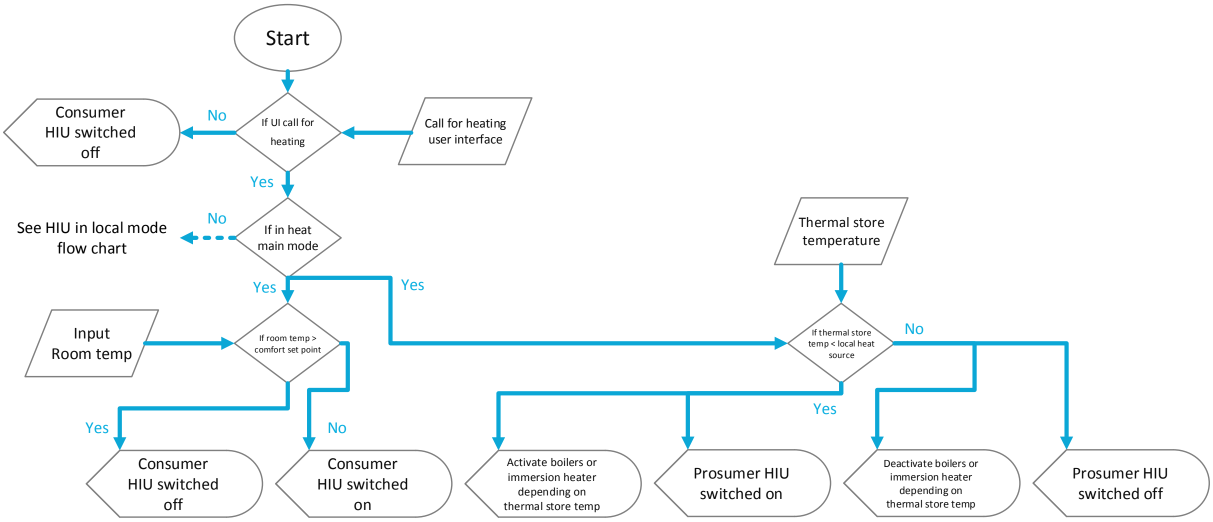

Fig. 4. Flow chart illustrating heat main mode control code layout. Including the start of the program (oval), data input (rhombus), decision to be made (diamond), and output operation (hybrid).

Solar thermal controls are somewhat separate from the rest of the system. In heat main mode where heat is supplied from the houses to the thermal store, the store temperature is a data input that informs the decisions on solar heat export functions. When the solar thermal controls are in local mode – only supplying heat to individual houses, the DHW cylinder temperature informs HIU operation.

Of note, to control which technology is exporting, the temperature thresholds for activation are staggered. Effectively the heat generators are prioritised based on their carbon emissions and sustainability credentials. Alternatively, system operators whose primary concern is cost could implement a similar hierarchy based on delivered cost per kWh to water for each heat source.

3.3. Completion

The actioning mechanisms in oBeMS works by a number of functions suggesting a state for control equipment, for example, open or close, on or off. These are sequenced so that safety critical issues are considered last. At the end of the analysis of the most recent data, the last suggested state is actioned. This prevents conflicting commands being sent, and in effect gives the safety critical code the ‘last word’ in the decision making process.

Once a decision has been made a logic level signal is sent from the RPi's GPIO pins to initiate switching of the solid state relays. The decision making code is run every minute in a loop controlled by an escapement mechanism which ensures accurate timing.

4. Software modules and data structures

The oBeMS system takes an approach to the design of building management systems which is modular and scalable. It allows implementation on single board systems at one extreme, and systems distributed across a site or sites at the other. Distributed controls have been implemented on the RPi because it is a low-cost platform which supports a powerful operating system and tool chains, and can be fully networked. To protect the integrity of the system, most nodes in the system are implemented with read-only file systems and hardware watchdog timers. An addition node may be added to archive raw data from the system, and decisions made. This node, generally described as ‘the server’, should use magnetic disks rather than the current generation of solid state disks, as these are more tolerant of frequent update.

4.1. Software modules

The oBeMS system is implemented as a number of C++ progrms, each running as a separate process. To date, all modules interact using an IP address to identify where the process being contacted is running, and a port number to identify the particular service which is required.

The process being contacted is a server process, and the process which initiates the connection is a client process. Within the oBeMS system, it is normal for a client process to communicate with multiple server processes in multiple location to obtain the information it needs to inform its decisions.

4.2. Data structures

The oBeMS system uses an object-oriented style of coding. In an object oriented software environment, objects (usually represented as classes) describe elements of the real world at various levels of detail. Objects typically have methods and properties. Properties hold state; data values. Methods provide ways to access or manipulate data and other objects. The relationship between software objects reflects our understanding, of the physical system controlled. The oBeMS framework is implemented in the C++ language which extends the original C language with powerful and efficient object-oriented extensions.

4.3. Network architecture

The oBeMS system uses Internet Protocols (IP) to transmit messages from machine to machine. IPv4 is used in preference to IPv6 because despite the technical superiority of v6, v4 is ubiquitous and change is held back by inertia throughout the network industry. Any distributed computer system needs to be able to access multiple machines which may be seen as multiple nodes in a network.

4.4. ObemsClientTemplate

ObemsClientTemplate is the prototype executive process within the oBeMS software, and from it, equipment is controlled. ObemsClientTemplate comprises a number of classes with various methods (functions), at the core of which is WorldAndTime.

WorldandTime (see Figure 5), is the root class of the system, acting as the anchor point for both the descriptions of the quantities being measured (thermal, electrical and concentration data descriptors), and the time series measurements that are made (one snapshot of data for all sensors per minute).

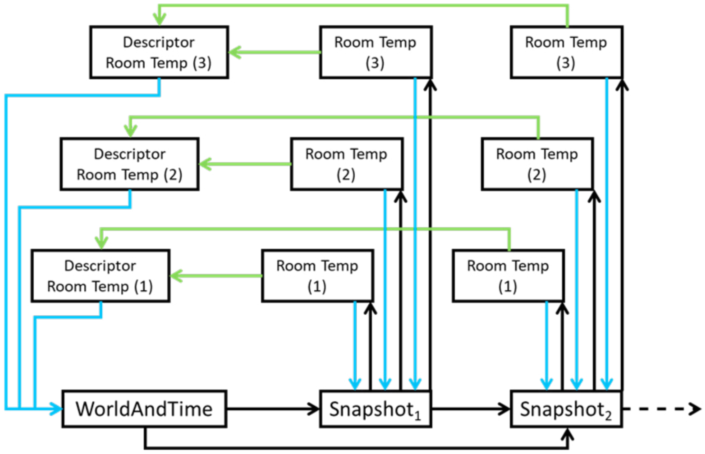

Fig. 5. Flow chart representing WorldAndTime code

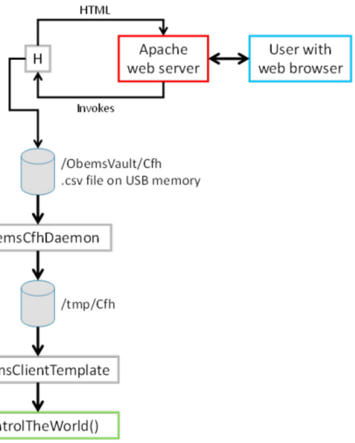

Fig. 6. Flow chart representing the oBeMS user interface

In the example below "Room Temp (1), (2) and (3)" represent thermal data items stored in a vector within the Snapshot, whereas "Descriptor Room Temp (1), (2) and (3)" represent thermal data item descriptors containing all of the detail about the data (metadata) and sensors. Each thermal data item has a link to its descriptor, for identification, and a pointer back to the Snapshot. Each Snapshot is linked to the last by the AppendSnapshot() function, and each data item within the snapshots is linked to the corresponding data item descriptor.

4.5. User interface

The user interface is a CGI program (H) accessed via the web server. It has two functions. Firstly, it allows users to set the heating to on/off four different times throughout the day. Secondly, it allows actions taken by the code to be written to the status screen, explaining the systems actions. The overall user interface set up can be seen in Figure 6. H is invoked via the web server, and the H code dynamically generates HTML which Apache serves to the user. H puts a file called "Cfh" in the ObemsVault directory, which is held in non-volatile USB flash memory. "Cfh" is a .csv file which contains information on heating set times. It is usually read-only to minimise the risk of file system or data corruption in the event of an unscheduled power down of the system, however H has the ability to make the ObemsVault file system read-write temporarily to update the heating schedule. Cfh.csv is read by ObemsCfhDaemon which writes files to /tmp/Cfh (Ramdisk), that are later read by the ControlTheWorld function in ObemsClientTemplate. This is the core within the ObemsClientTemplate executive process that controls equipment within the oBeMS system.

5. Control Strategy

The controls and hardware have been designed for flexibility, and as a result, research questions can be tested on the optimisation of heating systems in communities. Initially, we aim to discover which is the most economic and environmentally advantageous option for generation. Under these two categories, we can program the controls to run scenarios providing data for system optimisation.

6. Discussion and Conclusion

Buildings and their energy consumption play an important role in sustainable development, and as a result, well-designed BEMS can significantly reduce burdens on the global environment if it is affordable. Within the BEMS literature, the focus has largely been on electrical controls, probably as electricity consumption is increasing, and there are greater carbon and cost savings to be made per kWh compared to heat. However, if heat can be generated sustainably, the benefits of electrical heating will reduce.

Within this paper, a bespoke BEMS for low-temperature community heat networks with distributed generation is described. The system is the first of its kind and aims to test for optimisation of the prosumer concept. The oBeMS development platform is used for data logging and equipment control. The flexibility of the oBeMS software package allows for customisable controls, scenario testing and:

- Lower development, installation and maintenance costs
- Low-cost intra- and inter-site networking
- Archiving, aggregation, processing and analysis can be done by any tool

Previously, oBeMS software had largely been used in data logging applications, however, during project SCENIC, significant refinements and extensions have been made to the system. In order to focus on distributed generation scheduling in the network, a number of compromises were made in the development of the controls. For example:

- A simplified user interface – In the software, a seven day timer is provided which allows up to four on/off calls for heat per day through a web interface. The temperature set points however are changeable only in the software.
- No weather compensated heating - the thermal store is oversized and therefore allows transient demand to be satisfied without the need for increased heat generation during brief periods of colder weather.
- No artificial intelligence or learning algorithms are used - satisfactory control is achieved without learning algorithms. At the site level, a larger site would allow the integration of some predictive occupancy and energy consumption algorithms. With only seven houses patterns are less obvious, and the small number of participants renders the data on which learning decisions would have to be made very noisy.

By its nature, oBeMS is a development platform, and so moving forward there is scope to make the controls more sophisticated and deal with the comprises discussed.

Acknowledgements

Project SCENIC is part of SCENe, funded by Energy Research Accelerator (ERA) and Innovate UK.

References

[1] MOLINA-SOLANA, M., ROSA, M., RUIZ, D. M., GÓMEZ-ROMERO, J. & MARTIN-BAUTISTA, M. J. 2017. Data science for building energy management. *Renewable and Sustainable Energy Reviews* 70, 598-609.
[2] CLEMENTS-CROOME, D. 2004. *Intelligent buildings: Design, construction and operation,* London, Thomas Telford.
[3] BRE TRUST 2014. Information paper - Operating BEMS - A practical guide to building energy management systems.
[4] DUANGSUWAN, J. & KECHENG, L. 2008. Multi-agent control of shared zones in intelligent buildings. International conference on computer science and software engineering.
[5] JIANG, Z. 2006. Agent-based control framework for distributed energy resources microgrids. *IEEE International Conference on Intelligent Agent Technology*
[6] LEVERMORE, G. J. 2000. *Building Energy Management Systems - Application to Low-Energy HVAC and Natural Ventilation Control,* New York, E & FN Spon.
[7] CHAOUCH, H. & SLAMA, J. B. H. 2014. Modeling and simulation of appliances scheduling in the smart home for managing energy. *International conference on electrical sciences and technologies* Maghreb: CISTEM.
[8] T4 SUSTAINABILITY LTD. 2013. *Index of /oBeMS* [Online]. Available: http://t4sustainability.co.uk/oBeMS/ [Accessed 12th April 2018].
[9] CERANIC, B., BEARDMORE, J. & COX, A. 2018. Rapid deployment modular building solutions and climatic adaptability: Case based study of a novel approach to “thermal capacity on demand”. *Energy and Buildings,* 167, 124-135