

BEMOSS: An Agent Platform to Facilitate Grid-Interactive Building Operation with IoT Devices

*M. Pipattanasomporn, M. Kuzlu, W. Khamphanchai, A. Saha, K. Rathinavel and S. Rahman
Virginia Tech – Advanced Research Institute, Arlington, VA 22203*

Abstract—The objective of this paper is to present a multi-agent system (MAS) that can enable integration of a number of IoT devices to allow grid-interactive and intelligent building operation. The proposed platform called BEMOSS – Building Energy Management Open Source Software is developed to allow sensing and control of HVAC, lighting and plug load controllers in small- and medium-sized commercial buildings. The main objectives are to improve energy efficiency, reduce energy consumption, and foster demand response (DR) implementation in buildings with OpenADR protocol. Functionalities of BEMOSS software platform and its integration with IoT devices are demonstrated in a laboratory test bed located at Virginia Tech Advanced Research Institute. Future research directions are also discussed.

Index Terms—Multi-agent system, demand response, grid interactive, IoT devices.

I. INTRODUCTION

AGENT-BASED technologies are being utilized in power system applications, as well as novel applications such as home and building energy management. Previous work on agent-based systems is mostly limited to a conceptual or proof-of-concept stage that lacks real-world hardware implementation [1, 2, 3]. Instead of simulation, our goal is to provide a low cost and practical solution for a building energy management application with an open source software product that handles everything between users and physical hardware devices.

We have developed a Building Energy Management Open Source Software (BEMOSS - URL: www.bemoss.org) that is engineered to improve sensing and control of equipment in small- and medium-sized commercial buildings. According to Energy Information Administration (EIA) [4], about half (49.9%) of all commercial buildings in the US are 5,000 square feet or smaller in size. Medium-sized commercial buildings (between 5,001 to 50,000 square feet) constitute almost the other half (44.0%). These buildings typically do not have Building Automation Systems (BAS) due to lack of building owner awareness and high cost of existing BAS solutions. BEMOSS has been designed to target small and medium-sized commercial buildings to address this gap. BEMOSS aims to optimize electricity usage to reduce energy consumption and help implement demand response (DR) programs in commercial buildings.

For our application, VOLTTRON™ [5, 6] – a distributed agent platform developed by Pacific Northwest National Laboratory (PNNL) – is chosen as the agent

development platform for BEMOSS. This is because it is capable of interfacing with hardware devices, maintaining security, managing platform resources, and allowing connections with electric utilities via OpenADR [7]. The BEMOSS concept has been preliminarily described in [8]. The alpha version of BEMOSS (v1.2, September 2014 release) has been made publically available on www.bemoss.org.

In this paper, we emphasize how BEMOSS -- which is designed to work with three major types of loads in buildings, i.e., HVAC, lighting and plug loads, and selected sensors/power meters -- is integrated with Internet of Things (IoT) devices. IoT is the network of physical objects that can be monitored and controlled through the Internet. It is a rapidly growing system of sensors and connected devices. It is expected that over 50 billion devices will be connected by 2020 [9].

This paper is organized as follows. Section II gives an overview of conventional load controllers and selected IoT devices currently accessible by BEMOSS. Section III briefly discusses BEMOSS software architecture and provides the list of BEMOSS accessible load controllers. Section IV goes deeper into discussing BEMOSS multi-agent system. Section V discusses available BEMOSS applications and its potential applications with IoT devices. Section VI describes laboratory test bed at Virginia Tech.

II. BEMOSS SYSTEM ARCHITECTURE AND THE LIST OF BEMOSS ACCESSIBLE LOAD CONTROLLERS

Fig. 1 illustrates a BEMOSS architecture for a small commercial building (< 5,000 square feet) with a few load controllers of each type.

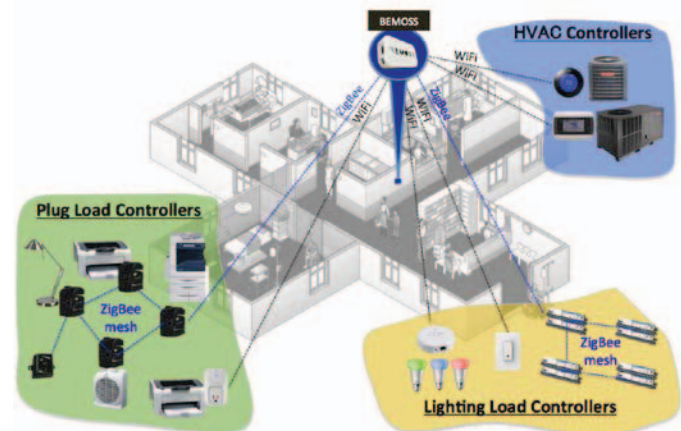


Fig. 1. BEMOSS system architecture for small commercial buildings.

In this architecture, only one single-board computer (e.g., a Odroid) with the BEMOSS solution is sufficient to enable monitoring and control features of all load controllers in the building. This embedded system can communicate with different types of load controllers, i.e., thermostats, lighting and plug load controllers, as well as sensors/power meters via wireless (either Wi-Fi or ZigBee) or wired (i.e., Ethernet – notshown) communications. Local and remote monitoring and control via a smart phone or a tablet are also enabled.

BEMOSS is designed to work with a number of load controllers and sensors/power meters in buildings that operate using different communication technologies and data exchange protocols. Currently, BEMOSS supports the following prevalent communication technologies: Ethernet (IEEE 802.3), Serial (RS-485), ZigBee (IEEE 802.15.4) and Wi-Fi (IEEE 802.11); and data exchange protocols: BACnet, Modbus, Web, ZigBee API, OpenADR and Smart Energy (SE) protocols.

List of BEMOSS accessible load controllers as of August 2015 are summarized in Table I.

TABLE I.
BEMOSS ACCESSIBLE LOAD CONTROLLERS

Device Model	Vendor	Communication
HVAC controller		
CT30 - WiFi USNAP	RadioThermostat	WiFi
CT50 -WiFi USNAP	RadioThermostat	WiFi
CT80 -WiFi/ZigBeeSE USNAP	RadioThermostat	ZigBee SE
ICM I3 series	ICM Controls	WiFi
Nest thermostat	Google	WiFi
EXL-01610 thermostat	Exact Logic	BACnet MS/TP
VC1000 series VAV	Prolon	Modbus TCP
PL-M1000RTU controller	Prolon	Modbus TCP
Lighting load controller		
Philips Hue	Philips	WiFi/Ethernet
WeMo light switch	Belkin	WiFi
Step-dimmed ballast	Douglas	ZigBee(customized)
LMRC-210 controller	Wattstopper	BACnet
Plug load controller		
WeMo smart plug	Belkin	WiFi
WeMo insight switch	Belkin	WiFi
Digi XBee smart plug	Digi	ZigBee API
LMPL-201 controller	Wattstopper	BACnet
Sensor		
Digi XBee sensor	Digi	ZigBee API
LMPC-100 occupancy sensor	Wattstopper	BACnet
LMLS-400 photosensor	Wattstopper	BACnet
Proteus MS motion sensor	Proteus	WiFi
Aeon Labs ambient light sensor	Aeon Labs	WiFi
Environmental sensor	NetAtmo	WiFi
Power meter		
Dent PowerScout 3+, Ethernet	Dent	BACnet IP
	Dent	Modbus TCP
Dent PowerScout 3+, Serial	Dent	BACnet MS/TP
	Dent	Modbus RTU
Wattnode WNC-3Y-208-MB	Wattnode	Modbus RTU

To enable communication with an electric utility, BEMOSS is integrated with a cloud service application to allow receiving signals via OpenADR. OpenADR is a communication data model designed to interact with utility's demand response (DR) signals. Through OpenADR, BEMOSS can receive electricity price signals or DR signals and take actions accordingly.

III. BEMOSS SOFTWARE ARCHITECTURE

The entire BEMOSS software architecture is illustrated in Fig. 2. As shown, it comprises four layers: (1) User Interface (UI); (2) Application; (3) Operating system and agent; and (4) API translator. In addition to these, all layers use databases to store metadata (e.g. device information) and time-series data (e.g. power measurement), and share important information.

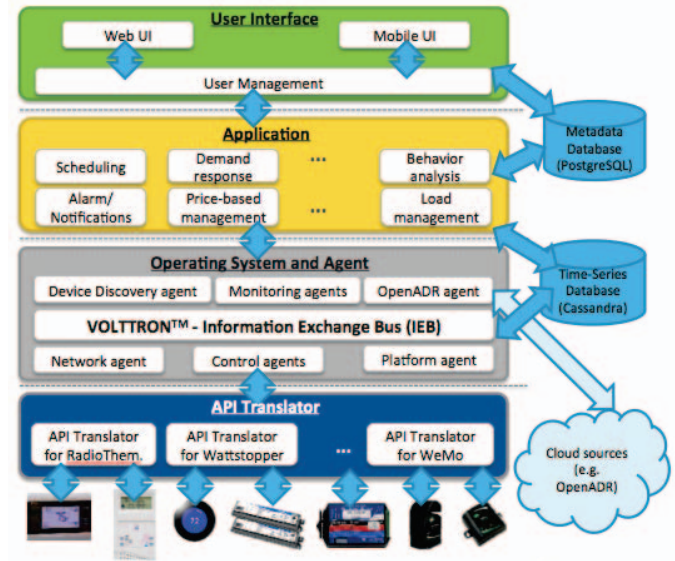


Fig. 2. BEMOSS software architecture.

Each layer is explained in more details below.

BEMOSS UI Layer consists of user management, web browser interface and mobile interface. BEMOSS UI is a responsive user platform to allow seamless interaction with the connected devices using a web browser from any type of device (smartphone, tablet, and laptop, desktop). User management is implemented using role-based access control to allow different levels of access to user based on their role in the system. The UI layer has built-in authentication and authorization, thus providing secure web access to the connected devices for both monitoring and control.

Application Layer includes various applications, which allow intelligent control of hardware devices interfaced with BEMOSS. Examples of possible applications include: fault detection & diagnostic, load shape analysis, demand response, price-based management, planning and scheduling, behavior pattern analysis, load management, as well as alarms and notification services.

Operating System and Agent Layer is where VOLTRON™ and BEMOSS agents reside. VOLTRON™'s Information Exchange Bus (IEB) serves as a medium that enables communication among all BEMOSS agents. Communication with the UI layer is also established using IEB. BEMOSS agents include device discovery agent, control agents, monitoring agents, and other service agents. This layer is explained in more details in the next section.

API Translator Layer takes care of the communication between BEMOSS agents and all physical hardware devices. API translators have been developed to allow BEMOSS agents to communicate with a group of devices based on their

unique Application Programming Interface (APIs). Basically, API translators provide a layer of abstraction for BEMOSS agents so that agents can obtain readings and send control commands to devices regardless of their APIs, using simple functions, i.e., `getDeviceStatus` and `setDeviceStatus`.

BEMOSS Database: In BEMOSS, there are two types of databases, one to store metadata information about different devices (using PostgreSQL) and the other to store time-series data (using Cassandra[10]).

IV. BEMOSS MULTI-AGENT SYSTEM

This section describes the BEMOSS multi-agent system and its architecture. Fig. 3 illustrates BEMOSS operating system and agent layer in more details. In this layer, BEMOSS utilizes VOLTTRON™ built-in modules, including Information Exchange Bus (IEB), agent authentication and authorization (AA), communication services (CS), resource management (RM), directory services (DS) and agent instantiation and packaging (AIP) modules. A number of BEMOSS agents have been developed and they communicate with each other using VOLTTRON™'s IEB.

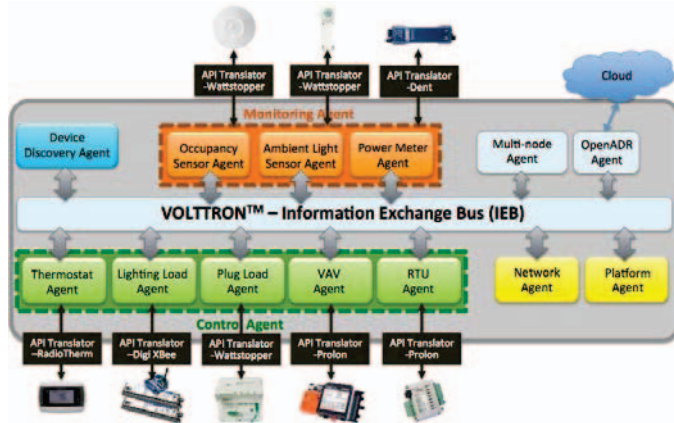


Fig. 3. BEMOSS operating system and agent layer.

A. Types of BEMOSS Agent

BEMOSS agents can be categorized into: (1) device discovery agent; (2) control agents; (3) monitoring agents; and (4) system agents, as summarized in Table II.

The *device discovery agent* is responsible for detecting new devices (either controllers or sensors) in a building and initiating each control/monitoring agent associated with the detected device.

Each *control/monitoring agent* has a one-to-one mapping with a hardware device. That is, a control agent is responsible for communicating with a load controller, which can be either a thermostat (e.g., RadioThermostat CT-50), a lighting load controller (e.g., Philips Hue), a plug load controller (e.g., WeMo smart plug), a VAV (Variable Air Volume Controller) or an RTU (Remote Terminal Unit); and a monitoring agent is responsible for obtaining readings from a sensor or a power meter. Each control/monitoring agent talks with its associated device via an API translator.

System agents include the network agent, the multi-node agent, the platform agent, and the OpenADR agent. The network agent facilitates agent mobility; the multi-node agent

facilitates multi-node communication; the platform agent supervises overall BEMOSS operation; and the OpenADR agent allows connection with an electric utility via the OpenADR protocol.

TABLE II.
BEMOSS AGENTS AND RESPONSIBILITIES

Agent	Description
Device discovery agent	Detect the presence of devices in buildings; query addresses, model numbers; identify their capability (APIs); launch associated control agents
Control agent	
Thermostat agent	Enable sensing and control of a thermostat (e.g., read temperature, change thermostat mode, change temperature set point, change fan mode)
Plug load agent	Enable sensing and control of a plug load controller (e.g., read/change its ON/OFF status)
Lighting load agent	Enable sensing and control of a lighting load controller (e.g., read/change its ON/OFF status, brightness level)
VAV agent	Enable sensing and control of a VAV controller
RTU agent	Enable sensing and control of an RTU controller
Monitoring agent	
Occupancy sensor	Obtain readings form an occupancy sensor
Motion sensor	Obtain readings from a motion sensor
Ambient light sensor	Obtain readings form an ambient light sensor
Environmental sensor	Obtain readings from an environmental sensor
Power meter agent	Obtain readings form a power meter
System agent	
Network agent	Facilitate agent mobility
Multi-node agent	Facilitate multi-node communication
Platform agent	Monitor agents' activities and their behaviors
OpenADR agent	Enable interaction with an electric utility using OpenADR protocol

B. BEMOSS Agent Initiation

Fig. 4 illustrates how a device discovery agent initiates a new control or monitoring agent once a new device is discovered in a building.

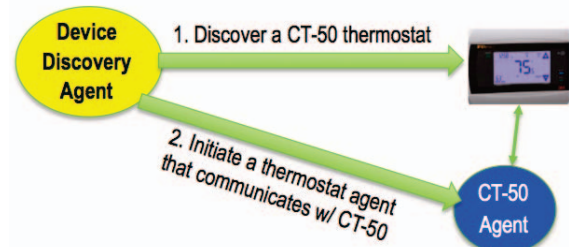


Fig. 4. Device discovery agent discovers a new device and initiates an associated control agent.

Whenever a building engineer connects a new device (e.g., a CT-50 thermostat) to a building network, the device is detected by the device discovery agent. Then, the device discovery agent initiates a thermostat agent that communicates with the CT-50 thermostat by creating the agent launch file, using the following information: agent ID, device type, device model, device IP address, agent message publish/subscribe addresses and topics to communicate with different layers of BEMOSS platform. A control/monitoring agent, with the provided launch file, is then initiated to start monitoring and control of the newly detected device, and connecting with BEMOSS databases.

C. BEMOSS Agent Behavior

In general, agent behaviors are the ability of an agent to react to changes in its environment and neighboring entities in pursuit of a system goal(s) or its own goal(s). BEMOSS control/monitoring agents are designed to have the following four behaviors.

- *Device_Monitor behavior*: This behavior constantly polls device information at a specific time interval, e.g., every 15 seconds. This allows a control agent to periodically update its knowledge on the current status of a device. However, for some devices that are capable of sending broadcast messages when their current status or settings are changed (e.g., Nest thermostat), this behavior acts as a listener to those events.

- *Update_Device_Status behavior*: This behavior is triggered when other entities would like to obtain the current status of a device. For example, when a user requests a thermostat page of BEMOSS UI, BEMOSS UI will send a request to the relevant thermostat agent to obtain updated readings from the thermostat. The agent also sends updates to the UI layer every time the device status changes, which is then processed and pushed to the UI web page, thus enabling push-style live updates (thus avoiding polling).

- *Device_Control behavior*: This behavior is triggered when there is a need to send a control command to a specific device. For example, a user would like to set a new device control parameter (e.g., thermostat set point, heat/cool mode and fan mode) to the CT-50 thermostat. In this case, when a user requests a new set point update for a thermostat, the UI layer processes this request and sends a message to the corresponding thermostat agent, via the IEB. The thermostat agent will then update the specific device control parameter by sending a control command to the device, and also sends a success/failure notification to the UI layer thus updating the status to the user.

- *Identify_Device behavior*: This behavior allows a user to distinguish a specific device from others by making visually identifiable changes on the device. It is activated by clicking an “identify device” button on BEMOSS UI. Once this command is sent, the associated agent will change a selected device parameter to allow visual identification of device location. For example, a blinking LED (if available) on a thermostat can help identify its location.

E. Agent Communications

For the current implementation, information exchanges among agents utilize the publish/subscribe mechanism facilitated by the VOLTRON™ built-in IEB module. Regarding interoperability among BEMOSS agents, there is the need for a common language (ontology) and communication protocols for agents to communicate and understand each other. Work is on-going to allow BEMOSS agents to comply with Foundation of Intelligent Physical Agents (FIPA) standards where agents communicate and update their knowledge by exchanging FIPA-ACL (Agent Communication Language) messages.

V. BEMOSS APPLICATIONS WITH IOT DEVICES

The BEMOSS platform opens up tremendous opportunities to integrate emerging IoT devices to building operation.

A. BEMOSS Integration with Selected IoT Devices

Emerging IoT devices may include smart thermostats, lighting and plug load controllers, and sensors that are accessible via the Internet. When working with IoT devices, BEMOSS acts as an IoT hub to aggregate things that use different protocols, such as Web, ZigBee API, and Smart Energy (SE). This is accomplished with API translators in the API translator layer. API translators provide a standardized approach to allow mapping from IoT device APIs (or spec sheets) to a format that the BEMOSS operating system can understand. This allows BEMOSS to communicate with IoT devices seamlessly. Examples of IoT devices integrated with BEMOSS are:

Google Nest Thermostat (URL: <https://nest.com/>) is a smart thermostat that can learn occupant’s schedule and program itself to lower heating and cooling bills. Nest uses a Firebase system to host data (e.g., room temperature, temperature set point, thermostat mode, humidity, fan status and battery health) on the cloud. Data are stored in the form of a JSON document, which can be retrieved by a user for information display on a web browser or a smart phone. Using Google Nest’s available API, an API translator is developed to allow a BEMOSS agent (in this case, a control agent associated with a Nest thermostat) to interact with a Nest thermostat via REST interface (alternatively, the Nest Firebase system can also be used) to obtain readings on current temperature ($^{\circ}F$), temperature set point ($^{\circ}F$), fan mode (on/auto), thermostat mode (off/heat/cool/auto/away), and battery health (measured in Volt); and to change temperature set point, thermostat mode and fan mode. See Fig. 5.

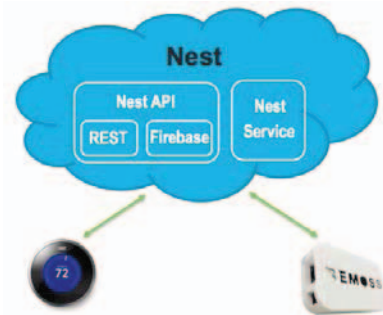


Fig. 5. BEMOSS communicates with a Nest thermostat via the cloud.

Once BEMOSS sends a command to change Nest’s temperature set point, the new data value is updated in real time and stored as a standard JSON document on the Nest Firebase system, which also reflects the change on the physical device.

Philips Hue (URL: <http://www2.meethue.com/>) is a set of wireless LED light bulbs with a bridge that allows a user to control brightness and intensity of LED light bulbs through a smart phone app. BEMOSS has been integrated with Philips Hue using a Philips Hue agent and an API translator for Philips Hue. Fig. 6 illustrates how BEMOSS is connected with a Philips Hue hub. This enables a user to send an ON/OFF command, change the brightness and light color, set lighting operating schedule and retrieve historical usage information through its BEMOSS UI.

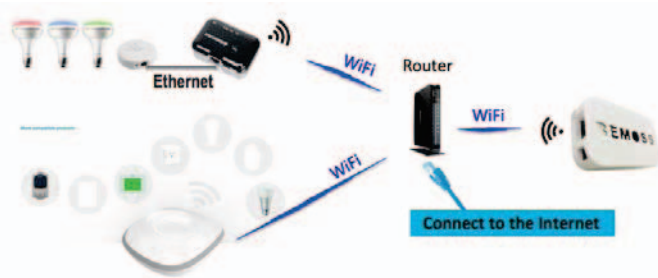


Fig. 6. BEMOSS communicates with a Philips Hue hub and a SmartThings hub via WiFi interface.

Smart Things (URL: <http://www.smarthings.com/>) is a platform that allows different kinds of compatible sensors and controllers to be monitored and controlled via a SmartThings hub. As shown in Fig. 7, the hub is connected to the Internet via an Internet router and supports ZigBee, Z-Wave as well as IP-accessible devices. BEMOSS is configured to interface with SmartThings using a SmartThings agent and a SmartThing API translator. BEMOSS can accept inputs from SmartThings-compatible sensors, including motion sensors, occupancy sensors and temperature/humidity sensors, and send command to turn on and off SmartPower outlets.

B. BEMOSS Applications

Currently available BEMOSS applications are thermostat scheduler, lighting load scheduler, and plug load scheduler which are used to schedule the corresponding devices to change their control settings according to time of the day.

BEMOSS is also integrated with **IFTTT**(If This Then That, URL: <https://ifttt.com/>) to allow alarms and notifications. IFTTT is a web-based automation service that allows users to perform an action depending on a trigger event. IFTTT integration allows BEMOSS to send emails/SMS notifications to notify a building engineer of an important event in a system, e.g., device failure or malfunction. Integration of IFTTT services with BEMOSS is accomplished by adding an interface between IFTTT and the agent. The term “interface” can be any channels available in IFTTT, as depicted in Fig. 7.

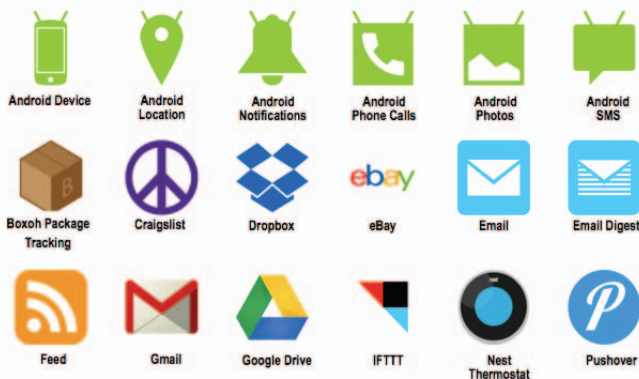


Fig. 7. Examples of channels that can be triggered by IFTTT.

Other potential BEMOSS applications include:

1) *Integration of machine learning algorithms to get better understanding of power consumption in buildings* – Data such as indoor/outdoor temperature, relative humidity, occupancy, or illuminance can be collected over time to allow BEMOSS

agents to understand usage patterns of power consumption. Using machine learning techniques, such as linear regression, support vector machine (SVM), or neural networks (NN), BEMOSS agents can allow effective DR implementation that does not significantly impact occupant comforts during a grid emergency condition.

2) *Integration of algorithms to manage a large amount of data collected from load controllers/sensors* – apparently a building may have thousands of data points to be collected either in one-minute intervals or less. Thus, there can be accumulation of tremendous amount of data. It is therefore important to efficiently manage, store and process such data. Data filtering techniques and distributed database approaches can be explored.

3) *Integration of algorithms to manage multiple buildings in a transaction-based energy network* – another possibility of BEMOSS applications is to allow multiple buildings to collaboratively respond to utility’s DR signals to reduce peak power demand, leading to a more efficient utilization of the electrical infrastructure.

VI. LABORATORY TEST-BED

A laboratory as shown in Fig. 8 has been set up at Virginia Tech Advanced Research Institute in Arlington, VA to demonstrate and test features and capabilities of BEMOSS in sensing and control of IoT devices and simulating connection with an electric utility using OpenADR.

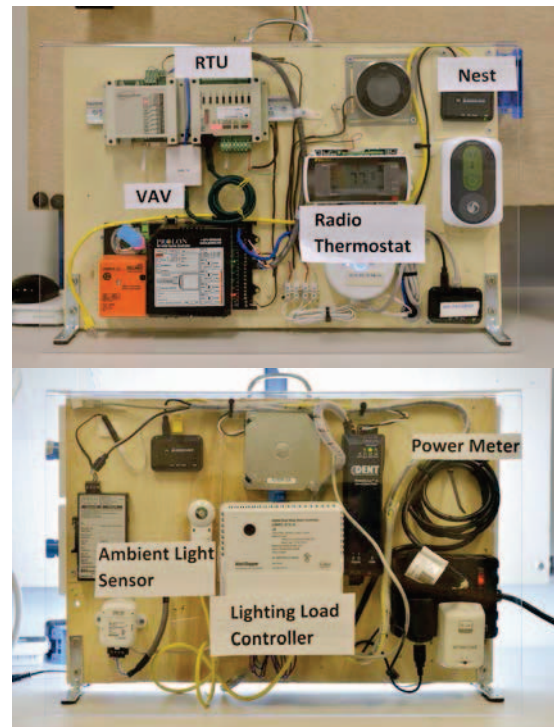


Fig. 8. BEMOSS laboratory set up at Virginia Tech – Advanced Research Institute in Arlington, VA.

Fig. 9 shows a BEMOSS UI dashboard page that displays all devices detected in a building by type, i.e., HVAC load controllers, lighting load controllers, plug load controllers and sensors/power meters.

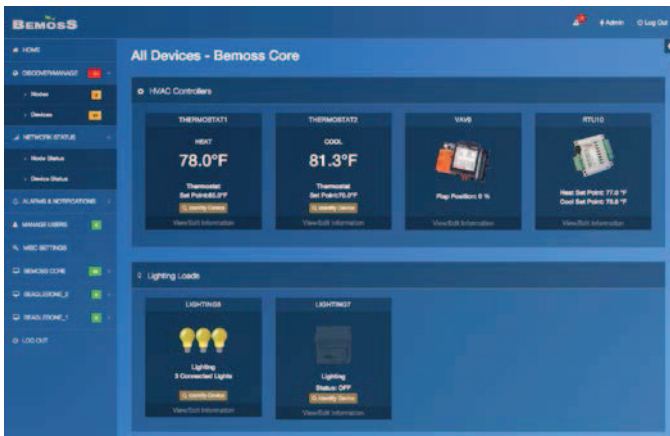


Fig. 9. BEMOSS UI dashboard page.

VII. SUMMARY AND FUTURE RESEARCH DIRECTIONS

This paper presents the BEMOSS platform for building energy management, and describes how IoT devices can be integrated in building operation using an agent-based technology. It is also the aim of this paper to identify future research directions in this area. Three future research directions are identified: (1) integration of machine learning and data analytics techniques to allow agents to understand usage patterns and implement self-learning algorithms to enable user comfort and save energy; (2) integration of distributed database and algorithm to manage a large amount of data; and (3) integration of algorithms to manage multiple buildings in a transaction-based energy network.

VIII. ACKNOWLEDGEMENT

This work is supported in part by the U.S. Department of Energy under Grant# DE-EE-0006352.0000.

IX. REFERENCES

- [1] A.M. Mzahm, M.S. Ahmad, and A.Y.C. Tang, "Agents of Things (AoT): An intelligent operational concept of the Internet of Things (IoT)," *Intelligent Systems Design and Applications (ISDA), 2013 13th International Conference on*, vol., no., pp.159,164, 8-10 Dec. 2013.
- [2] B. Manate, V.I. Munteanu, and T.-F. Fortis, "Towards a Scalable Multi-agent Architecture for Managing IoT Data," *P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2013 Eighth International Conference on*, vol., no., pp.270,275, 28-30 Oct. 2013.
- [3] M. Roscia, M. Longo and G.C. Lazaroiu, "Smart City by multi-agent systems," *Renewable Energy Research and Applications (ICRERA), 2013 International Conference on*, vol., no., pp.371,376, 20-23 Oct. 2013.
- [4] EIA, Commercial Buildings Energy Consumption Survey Data (CBECS), 2003 [Online]. Available: <http://www.eia.gov/consumption/commercial/data/2003/index.cfm?view=consumption#e1a>.
- [5] PNNL, VOLTRON™ – An intelligent agent platform for the smart grid, 2015 [Online]. Available: <http://gridoptics.pnnl.gov/VOLTRON/>.
- [6] J. Haack, B. Akyol, N. Tenney, B. Carpenter, R. Pratt, and T. Carroll, "VOLTRON™: An agent platform for integrating electric vehicles and Smart Grid," *Connected Vehicles and Expo (ICCVE), 2013 International Conference on*, vol., no., pp.81,86, 2-6 Dec. 2013.
- [7] OpenADR [Online]. Available: <http://www.openadr.org/>.
- [8] W. Khamphanchai, A. Saha, K. Rathinavel, M. Kuzlu, M. Pipattanasomporn, S. Rahman, B. Akyol, and J. Haack, "Conceptual architecture of building energy management open source software (BEMOSS)," *Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), 2014 IEEE PES*, vol., no., pp.1,6, 12-15 Oct. 2014.
- [9] Cisco, The Internet of Things, 2015 [Online]. Available: <http://share.cisco.com/internet-of-things.html>.
- [10] Apache Cassandra™ [Online]. Available: <http://cassandra.apache.org/>.

X. BIOGRAPHIES

Manisa Pipattanasomporn (S'01, M'06, SM'11 – IEEE) is an associate professor of Virginia Tech's Department of Electrical and Computer Engineering. She serves as one of the principal investigators (PIs) of multiple research grants from the U.S. National Science Foundation, the U.S. Department of Defense and the U.S. Department of Energy, on research topics related to smart grid, microgrid, energy efficiency, load control, renewable energy and electric vehicles. Her research interests include renewable energy systems, energy efficiency, distributed energy resources, and the smart grid.

Murat Kuzlu (M'11, SM'14 - IEEE) is an assistant professor of Virginia Tech's Department of Electrical and Computer Engineering. He received his B.Sc., M.Sc., and Ph.D. degrees in Electronics and Telecommunications Engineering from Kocaeli University, Turkey, in 2001, 2004, and 2010, respectively. In 2006, he joined the Energy Institute of TUBITAK-MAM, where he worked as a senior researcher at the Power Electronic Technologies Department. His research interests include smart grid, demand response, smart metering systems (AMR, AMI, AMM), wireless communication and embedded systems.

Warodom Khamphanchai (S'11 - IEEE) is a PhD candidate in the Department of Electrical and Computer Engineering at Virginia Tech, USA. His research interests are home/building energy management systems, Internet of things, multi-agent systems, machine learning, data analytics, demand response, cyber-physical systems, and power system operation and control. His research focus is to develop applications for energy saving, peak demand reduction, and demand response in small- and medium-sized commercial buildings. He is the active member of the IEEE Power and Energy Society, the IEEE Internet of Things Society, the IEEE Computer Society, and TechShop DC/Arlington. He received his Master degree in Electric Power System Management Specialization, Energy Field of Study from the School of Environment, Resources, and Development (SERD), Asian Institute of Technology (AIT), Thailand in 2011 and the B.Eng. degree in Electrical Engineering from Chulalongkorn University, Thailand in 2009. Before joining Virginia Tech in 2011, He worked as a power distribution engineer at the electric power system research division, research department, Provincial Electricity Authority of Thailand (PEA).

Avijit Saha (S'06 – IEEE) received his B.Sc. degree in Electrical and Electronic Engineering (EEE) from Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh, in 2009 and MS degree in Electrical Engineering from Virginia Tech, US in 2013. He served as a Lecturer at the department of EEE in Ahsanullah University of Science and Technology, Dhaka, Bangladesh from March 2010 to July 2011. Currently he is working towards his Ph.D. degree in Electrical Engineering in Virginia Tech. His research interests include the areas of smart grid, demand response, energy management systems, embedded systems, communication technologies, software engineering, data analysis and learning, and smart grid security.

Kruthika Rathinavel (S'14 – IEEE) received her Bachelor of Engineering (Honors) degree in Electronics and Instrumentation from Birla Institute of Technology and Science (BITS, Pilani), Goa, India in 2008. She has over four years of experience as a software consultant, when she was associated with CGI (previously Logica), CAI (Tyco Electronics) and NTT Data. She is currently pursuing her MS degree in the Electrical and Computer Engineering Department, Virginia Tech. Her research interests are internet of things, information storage and retrieval, network security, software engineering, web development and machine learning.

Saifur Rahman (S'75, M'78, SM'83, F'98 - IEEE) is the director of the Advanced Research Institute at Virginia Tech where he is the Joseph Loring Professor of electrical and computer engineering. He also directs the Center for Energy and the Global Environment at the university. From 2009-2013 he served as a vice president of the IEEE Power & Energy Society and a member of its Governing Board. He is a member-at-large of the IEEE-USA Energy Policy Committee. Professor Rahman was the chair of the US National Science Foundation Advisory Committee for International Science and Engineering from 2010 to 2013. Between 1996 and 1999 he served as a program director in engineering at NSF. In 2006 he served as the vice president of the IEEE Publications Board, and a member of the IEEE Board of Governors. He is a distinguished lecturer of IEEE PES, and has published in the areas of smart grid, conventional and renewable energy systems, load forecasting, uncertainty evaluation and infrastructure planning.