Energy Research and Development Division
FINAL PROJECT REPORT

IMPLEMENTATION OF DEMAND RESPONSE IN A UNIVERSITY CAMPUS

Prepared for: California Energy Commission
Prepared by: Smart Grid Energy Research Center
University of California Los Angeles

AUGUST 2015
CEC-500-2015-087

PREPARED BY:

Primary Authors:
Shiv Prabhu
Saurabh Trikande
Omar Sheik
Rajit Gadh

University of California, Los Angeles
44-120 Engr IV, 420 Westwood Plaza
Los Angeles, CA 90095
Phone: 310-825-5378 | Fax: 310-206-4830
http://smartgrid.ucla.edu/

Contract Number: 500-11-013

Prepared for:

California Energy Commission

Matthew Fung
Contract Manager

Fernando Pina
Office Manager
Energy Systems Research Office

Laurie ten Hope
Deputy Director
ENERGY RESEARCH AND DEVELOPMENT DIVISION

Robert P. Oglesby
Executive Director

DISCLAIMER

This report was prepared as the result of work sponsored by the California Energy Commission. It does not necessarily represent the views of the Energy Commission, its employees or the State of California. The Energy Commission, the State of California, its employees, contractors and subcontractors make no warranty, express or implied, and assume no legal liability for the information in this report; nor does any party represent that the uses of this information will not infringe upon privately owned rights. This report has not been approved or disapproved by the California Energy Commission nor has the California Energy Commission passed upon the accuracy or adequacy of the information in this report.

ACKNOWLEDGEMENTS

The following team members contributed to the data preparation and analysis and writing this report.

- Shikar Dogra
- Harsha Bhat
- Saurabh Trikande
- Arturo Diaz
- Omar Sheikh
- Shiv Prabhu

The research team would like to acknowledge and thank everyone who contributed to the success of this research project. The following people were involved in the design, test bed site identification, research, development, implementation, and execution of demand response programs and data collection for the Demand Response System.

- Shikhar Dogra
- Harsha Vaman Bhat
- Phani Madhav Yannam
- Surabh Trikande
- Soohan Kim
- Sudarchan Vasista Srinivasan Rangan
- Vikas Tikoo
- Omar Sheikh
- Josh Chynoweth
- Kenneth MacKenzie
- Anna Purtell
- Shiv Prabhu
- Rajit Gadh

Finally, the research team extends sincere gratitude to facilities team of Housing & Hospitality Services – University Apartments, University of California Los Angeles for their excellent assistance in preparing, maintaining, and troubleshooting the Demand response test bed. Research team also extends thanks to office staff of Smart Gird Energy Research Center, University of California Los Angeles for administrative and procurement services.

PREFACE

The California Energy Commission Energy Research and Development Division supports public interest energy research and development that will help improve the quality of life in California by bringing environmentally safe, affordable, and reliable energy services and products to the marketplace.

The Energy Research and Development Division conducts public interest research, development, and demonstration (RD&D) projects to benefit California.

The Energy Research and Development Division strives to conduct the most promising public interest energy research by partnering with RD&D entities, including individuals, businesses, utilities, and public or private research institutions.

Energy Research and Development Division funding efforts are focused on the following RD&D program areas:

- Buildings End-Use Energy Efficiency
- Energy Innovations Small Grants
- Energy-Related Environmental Research
- Energy Technology Systems Integration
- Environmentally Preferred Advanced Generation
- Industrial/Agricultural/Water End-Use Energy Efficiency
- Renewable Energy Technologies
- Transportation

Implementation of Demand Response in a University Campus is the final report for the Implementation of Demand Response in a University Campus project (contract number 500-11-013) conducted by Smart Grid Research Energy Center, University of California Los Angeles. The information from this project contributes to Energy Research and Development Division's Energy Technology Systems Integration Program.

For more information about the Energy Research and Development Division, please visit the Energy Commission's website at www.energy.ca.gov/research/ or contact the Energy Commission at 916-327-1551.

ABSTRACT

Automated demand response has the potential to allow electric utilities to independently monitor and control customer premise loads to reduce peak electricity demand. Despite its conveience, widespread implementation of this energy management system depends on several uncertain factors:

- Customer acceptance of utility-automated demand response programs
- Reliable transmission of demand response signals from their source to desired loads (such as heating, ventilation, and air conditioning systems, electrical appliances, plug loads, lighting systems, and electric vehicles) through adequately robust communications infrastructure
- Automatic and intelligent reduction of a given amount of electrical load within a given time interval in a utility provider's service area

In this report, the research team describes the design, simulation, and results of an automated demand response test bed that provides a platform to investigate intelligent electrical load reduction. The test bed includes a variable refrigerant flow, ductless heating, ventilation, and air conditioning systems, and a nano-scale battery that will typically be part of aggregate demand response resources in small commercial and residential demand response markets. The design emphasis of the test bed was to develop an architecture that would facilitate automatic resource selection for load curtailment or on-site supply to support energy shed, shape, shift or substitute modes for demand adjustment. The test bed achieved automated load curtailment without participant intervention, minimizing the customer inconvenience factor that has substantially lowered the participation rate in utility demand response programs.

Keywords: Demand response, OpenADR, variable refrigerant flow HVAC, battery storage, test bed

Please use the following citation for this report:

Prabhu, Shiv; Saurabh, Trikande; Omar, Sheik; Rajit, Gadh. (UCLA Smart Grid Energy Research Center). 2015. *Implementation of Demand Response in a University Campus***.** California Energy Commission. CEC-500-2015-087.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS ..... **i**

PREFACE ..... **ii**

ABSTRACT ..... **iii**

TABLE OF CONTENTS ..... **iv**

LIST OF FIGURES ..... **vi**

LIST OF TABLES ..... **vii**

EXECUTIVE SUMMARY **1**

Introduction 1

Project Purpose 1

Project Process and Results 2

Project Benefits 2

CHAPTER 1: Introduction **4**

1.1 Motivation 5

1.2 Location and Details of Test-bed 5

1.3 Automated Demand Response (AutoDR) Test-bed 5

1.3.1 Automated Demand Response System 5

1.4 Ductless HVAC System as Demand Response Resource 6

1.5 On-site Battery Storage as Demand Response Resource 8

1.6 Residential Demand Response Modeling and Simulation 8

CHAPTER 2: OpenADR on WINSmartGrid Platform **9**

2.1 ReWINS Technology 9

2.2 WINSmartGrid Technology 10

2.2.1 Smart Gateway 11

2.2.2 Appliance Controller 13

2.3 OpenADR Clients on WINSmartGrid 14

CHAPTER 3: Automatic Demand Response Architecture **15**

3.1 Open Automated Demand Response 15

3.2 Demand Response Automation Server (DRAS) 15
3.2.1 Architecture 15
3.2.2 Functional Description of the Architecture 18
3.2.3 DR Event 22
3.3 Experiments and Results 22
3.4 Observations 24
CHAPTER 4: Small Commercial & Industrial HVAC System as Demand Response Resource 26
4.1 BAC-HD150 Adapter Functions 26
4.2 BACnet Commands Generated in Response to DR Events 27
4.3 Experiments and Results 28
4.4 Observations 31
CHAPTER 5: Battery Storage as Demand Response Resource 32
5.1 Battery Rig 32
5.2 Automatic Demand Reponses Controller 33
5.2.1 Battery Management System 34
5.3 Experiments and Results 35
5.3.1 Time-bound and Preference-Dictated Demand Reponses Curtailment Tests 36
5.4 Observations 39
CHAPTER 6: Residential Demand Response Modeling and Simulation 40
6.1 Electric Service Model 40
6.2 Simulation with three ESMs 41
6.3 Observations 43
CHAPTER 7: Project Accomplishments and Conclusion 45
7.1 Automatic Demand Response System 45
7.2 Ductless HVAC System as Demand Resource 46
7.3 Storage Battery as Demand Resource 46
7.4 Residential Demand Response Modeling and Simulation 47
7.5 Technology Transfer Activities 47

GLOSSARY **49**

REFERENCES **51**

APPENDIX A: Sample DR Event **A-1**

APPENDIX B: Mitsubishi CITY-MULTI HVAC System Specification **B-1**

APPENDIX C: BACnet Commands for HVAC Load Curtailment Tests **C-1**

APPENDIX D: Battery Specification **D-1**

APPENDIX E: Backup Power for Critical Load during DR Event **E-1**

APPENDIX F: Electrical Service Model **F-1**

APPENDIX G: Residential Demand Response Simulation Results **G-1**

LIST OF FIGURES

Figure 1: CITY MULTI R2-Series Two-pipe Simultaneous Cooling and Heating HVAC System . 6

Figure 2: CITY MULTI HVAC System Outdoor and Indoor Units 7

Figure 3: Schematic of ReWINS Platform 9

Figure 4: ReWINS Hardware for Monitoring Perishable Assets 10

Figure 5: WINSmartGrid Smart Gateway 11

Figure 6: Preference Manager 12

Figure 7: WINSmartGrid Appliance Controller 14

Figure 8: REST Client-Server Service Topology of the Architecture 16

Figure 9: OpenADR-based DRAS Architecture Showing Facility and Customer/DRAS Client Interfaces 18

Figure 10: DRAS Sub-Systems and Functions 19

Figure 11: DR Event Structure 22

Figure 12: CITY MULTI HVAC System Configuration with BAC-HD150 Connected to M-NET26

Figure 13: CITY MULTI HVAC Group Configurations with BAC-HD150 Connected to M-NET 27

Figure 14: LiFePO4 Storage Battery Test Rig 33

Figure 15: Auto Demand Response Controller with On-Board OpenADR Client 34

Figure 16: Battery Management System 34
Figure 17: Battery Management System Charging 36
Figure 18: Battery Management System Discharging 36
Figure 19: Lamp Bank Used As Critical Load Supported by the Battery 37
Figure 20: Cell Voltage and Amperes During the 10 Minute DR Event 38
Figure 21: Cell Temperature During 10 Minute DR Event 38
Figure 22: Relationship of ESMs and ESOs in Smart Grid 40
Figure 23: Electrical Service Model Hierarchy and XML Data Model 41
Figure 24: Simulator with the Processing Engines 42
Figure 25: Activity Diagram for Randomizer Engine 43
Figure 26: Demand Response Event Scheduling 1
Figure 27: Cell Voltage and Amperes During the 60 Minute DR Event 2
Figure 28: Cell Temperature During 60 Minute DR Event 2

LIST OF TABLES

Table 1: Test Results of DRAS Server Installation 23
Table 2: Test Results of DRAS to BAC-HD150 Communications 23
Table 3: Test Results of DRAS to Battery Rig Communications 24
Table 4: BACnet Protocol Packet Structure 27
Table 5: Identification of BACnet Components for the Planned Tests 28
Table 6: Test to Reduce Compressor Frequency 29
Table 7: Test to Sequence Zonal On/Off 29
Table 8: Test to Turn Off Indoor Units Selectively 30
Table 9: Test to Minimize Demand Power Spikes 30
Table 10: Various Load Curtailment Strategies Tested 31

EXECUTIVE SUMMARY

Introduction

Demand response (DR) is a method of maintaining electric grid stability in which end-use customers voluntarily modify the amount or timing of their energy consumption in response to energy pricing or grid reliability signals. This method began as a strategy to avoid system emergencies in the electric grid during periods of peak electricity demand. With technological advances and innovations, DR has evolved to provide a variety of additional services that promote energy efficiency, system reliability, reduce energy cost, reduce greenhouse gases, and integrate renewable generation; however, these rapid developments have yet to reach electric utilities' desired range of customers, particularly the residential service base, due to:

- The lack of smart end-to-end solutions that keep customers engaged with discernable stream of benefits.
- The lack of DR friendly, do-it-yourself, and inexpensive control technology for home owners.
- The dearth of intuitive DR programs and strategies.
- Above all, the scarcity of analytical tools that measure and validate DR participation and load curtailment reliably.

In this project, the research team explored solutions for some of the aforementioned issues that are restricting access to new developments in DR by proposing:

- Lightweight *automated demand response* (ADR) architecture.
- Electric load control technology with integrated *open automated demand response* (OpenADR) clients. OpenADR is a communication protocol that allows electrical utilities to send DR signals to their customers.
- Design of demand response resource specific DR strategies.

The scope of the project includes the design, development and testing of an ADR test bed that incorporates these strategies on the University of California, Los Angeles (UCLA) campus.

Project Purpose

The purpose of this project was to explore the feasability of an OpenADR-friendly end-to-end solution that could:

- Host OpenADR software on the (ReWINS) server technology developed at the Smart Grid Energy Research Center (SMERC) at UCLA.
- Have OpenADR accommodate customer preferences and execute DR strategies in response to DR events automatically with little-to-no intervention from the customer.
- Support intelligent DR programs and strategies that work with AutoDR.

Project Process and Results

A test bed consisting of a ductless heating, ventilation, and air conditioning (HVAC) system in a graduate housing complex and a storage battery rig, each with an OpenADR-friendly control system, was added to the existing UCLA DR test bed using smart appliances, smart lighting, and electric vehicle (EV) charging infrastructure. The HVAC system and battery rig presented a challenging but crucial opportunity to experiment in providing demand response resources for the residential domain. HVAC systems generally are a large electrical load that has high potential for electrical load curtailment, and storage batteries might soon become part of the energy landscape.

To perform the research, the research team divided the project into four parts:

- *AutoDR Architecture*: A services-oriented architecture was selected suitable for the smart controls and use of common hypertext transfer protocol (HTTP) methods that simplify implementation and minimize network traffic. The architecture allowed efficient exchange of planned DR events and signals and managed electrical loads.
- *Ductless HVAC as DR resource*: An existing HVAC system was retrofitted to comply with BACnet (data communication protocol for building automation and control networks) and integrated with a bridge OpenADR client that translated DR events to BACnet signals. The translation proved successful in generating BACnet signals for controlling zone and indoor temperatures, and responding to HVAC DR strategies.
- *Behind-the-meter battery storage as DR resource*: A battery pack with an integrated battery management system was interfaced with an intelligent controller hosting an OpenADR client. Battery control proved effective in both switching energy supply to the designated electrical load for a set duration and switching back to the charge cycle in response to DR strategies.
- *DR modeler and simulator*: A DR simulator supported by a data model called Electric Service Model that represents residential service, electric loads, and user priorities and preferences successfully selected the target curtailment that complied with the customer's preferences.

Project Benefits

In spite of the small scale of the project for an AutoDR solution, the successful demonstration of (1) an OpenADR client design hosted on control platforms, (2) DR resource specific curtailment strategies, and (3) the execution of DR strategies guided by associated user preferences, has provided a system to alleviate some of the aforementioned challenges inherent in implementing residential DR programs.

The DR programs and strategies exhibited in this project can be part of the DR portfolio on the UCLA campus and are expected to guide campus renovation and modernization projects. SMERC also believes that these technologies and strategies would benefit California utilities by providing means to directly automate discrete end-point loads that are OpenADR conversant and render them capable of participating in demand response programs with minimal human

assistance. An automated network of smart loads would also provide detailed grid visibility in real-time to utilities to develop analytics and forecasting systems facilitate dispatching strategies. The proposed framework is also adaptable to all types of customers and can be deployed under aggregated management and control schemes.

CHAPTER 1: Introduction

Smart grid is an intelligent electrical grid that will distribute electricity from distributed generating resources to consumers under the supervision of two-way digital communications that will monitor, regulate, and control generation, distribution, storage, and consumption. It is anticipated that this automation will boost distributed control, minimize energy losses, enhance reliability, and promote effectual consumption across the electrical ecosystem economically (Mitchell et al. 2014; Smith 2010).

This transformation from traditional electric grid demands innovations on multiple fronts on a large scale, warranting experimental research, investigation, and validation before technologies, architectures, and operations can be assimilated by the industry. Smart grid research test-beds allow testing of advanced technologies, business processes, market models, innovative strategies and programs required to investigate and evaluate the new enabling components of smart grid such as renewable energy generation, electric vehicle infrastructure, microgrids, communication networks, control strategies, and customer engagement (Jarred 2014; Epstien et al. 2005).

Demand response (DR) is emerging as an application in the smart grid ecosystem. DR programs essentially depend on reduction in electricity demand during grid stress and primarily rely on load management by modifying power consumption to align power demand and supply. For DR to be successful, a collection of technologies, strategies, and software components in addition to customers' commitment is crucial. Since there are a myriad of potential solutions for achieving this goal, a preliminary evaluation of such solutions would assist in selecting a best-of-breed solution. A test-bed consisting of these technologies, software components and means to design and execute DR strategies would be invaluable as there are challenges to successful integration of demand response solutions. Challenges are systems architecture and integration, reliable control technology and strategies, consumer technology, fragile cyber security, fostering customer interest and engagement, measurement and validation, and policy lag and confusion (O'Connell et al. 2014; Spam, Jin, and Earle 2013).

This research project attempts to develop solutions that may mitigate some of the above challenges and pave way towards durable solutions. The focus was on developing a system architecture that supports automatic DR for a large UC campus with a captive distribution system that produces 85 percent of the campus power by a cogeneration plant, managing the campus distribution and auxiliary systems, deploying DR control intelligence, and using battery storage as a demand response.

In addition, a low-cost DR simulator was developed that may be used to estimate the curtailable load available for a particular DR event based on the customer preferences to provide a curtailable baseline, evaluate the impact of aggregation of types of curtailable loads during a DR event and use it as a DR learning tool for dispatchers and facility managers.

The following sections describe the development of the above solutions and planned experiments for monitoring and controlling of DR resources and different strategies using them.

1.1 Motivation

A research focus of the project was to investigate the potential of enabling automated control of residential electric loads by adding onboard capability to directly process OpenADR signals and manage control cycles to minimize electricity use within the constraints of customer specified end-use preferences. The investigation involved evaluating controller design, communications reliability, and robustness, system health and measuring DR performance.

1.2 Location and Details of Test-bed

The test-bed is multilocational and the demand response resources include a Variable Refrigerant Flow, ductless HVAC system in a graduate student apartment block and the battery storage rig in UCLA's Smart Grid Energy Research Center (SMERC) lab. The HVAC system supports 83 apartments in two separate buildings with separate compressor / condenser units for the buildings. Each apartment has an independent indoor unit and a remote control unit. The battery storage rig is equivalent to an electric vehicle (EV) conversion kit and is deployed in the lab in conditioned environment for safety reasons.

1.3 Automated Demand Response (AutoDR) Test-bed

The test-bed was designed to support automated DR that enables reduction in electrical loads in response to utility / facility requests without human intervention using a REST architecture based solution that supports secure two-way communication between the smart control clients and the demand response server.

1.3.1 Automated Demand Response System

The architecture of the OpenADR specification compliant Demand Response Automation Server meet the interoperability requirements of an architecture that is integrated and hosted within the UCLA facilities IT infrastructure. The distributed client-server architecture supports continuous, two-way, secure messaging between facilities and the participating customers.

Service-orientation is based on the Representational State Transfer (REST) paradigm developed by UCLA. This exploits the benefits provided by REST Web services that are lightweight, maintainable, and scalable. In the context of the constrained smart demand-side controls that are essential for automating DR, the REST Web Services provides many advantages, such as reduced network traffic, minimum coupling between client and server components, lightweight requests and responses, and explicit use of HTTP methods (Shelby 2010; Laine 2011).

This architecture is ideal for Web-enabling the smart controls in building automation systems (BAS) or *home area networks* (HAN) as it facilitates networking of clients (smart controls) with a variety of capabilities using the uniform HTTP interface with the Web. The use of HTTP protocol and basic Web axioms in REST simplifies interaction between server and client on smart controls of devices, appliances, and gadgets.

Chapter 3 describes the experimentation plan for the DR architecture in HAN to support smart controls for a variety of residential smart electrical loads.

1.4 Ductless HVAC System as Demand Response Resource

Mitsubishi Electric's CITY MULTI R2-Series Variable Refrigerant Flow (VRF) with Ductless Heat Pump (DHP) air conditioning system installed in multiunit buildings with a total of 83 single graduate student units has been identified as a demand response resource. This multizone HVAC system is capable of simultaneously cooling and heating different zones in the building. It allows setting of optimum temperature for each apartment / room based on location, ambient conditions, and occupancy. Figures 1 and 2 show the diagram of the CITY MULTI HVAC system with the outdoor and indoor units where different DR strategies can be experimented.

Figure 1: CITY MULTI R2-Series Two-pipe Simultaneous Cooling and Heating HVAC System

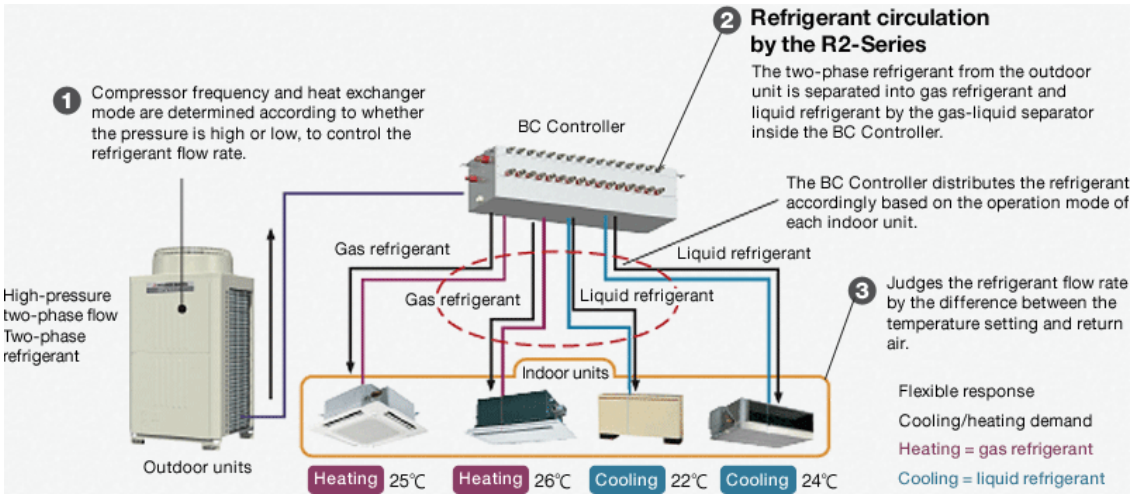


Image Credit: Mitsubishi Electric's multi-zone HVAC system
[http://www.mitsubishielectric.com/whatschanging/ecochanges/heatpump/]

Figure 2: CITY MULTI HVAC System Outdoor and Indoor Units

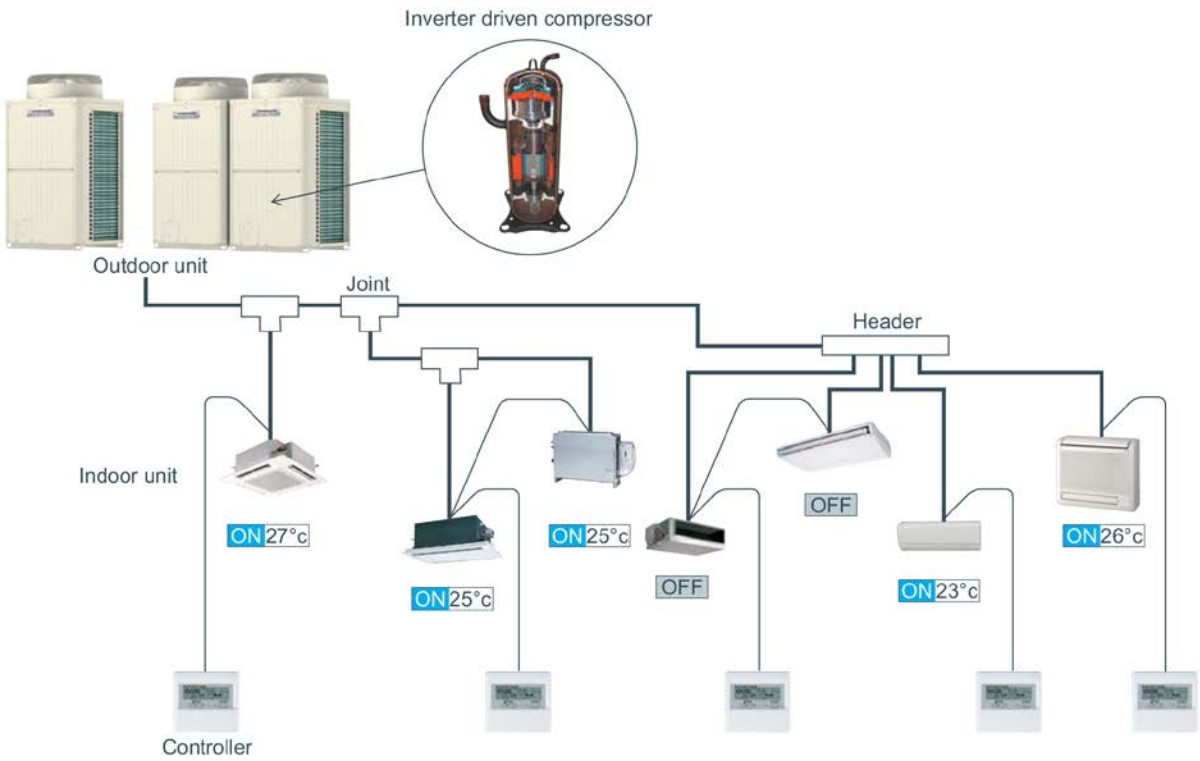


Image Credit: CITY MULTI - Mitsubishi Air Conditioning Systems Catalog, CM11AS-1, Oct. 2011

CITY MULTI system uses a proprietary 24 to 30 volts D.C. digital communication called Mitsubishi Network (M-NET), where the voltage is generated from the outdoor unit to the indoor units and is not compliant with OpenADR. To make it OpenADR complaint it had to be retrofitted with BAC-HD150, BACnet/Internet Protocol (IP) controller that bridges BACnet to M-NET.

With this setup, the plan was to investigate viability of different DR strategies by interfacing it with the OpenADR system. Since BAC-HD150 is a dedicated hardware unit, it is not possible to host an OpenADR client onboard. As a result, OpenADR client will be hosted as a bridge client that translates OpenADR event signals to BACnet commands.

The operations of the following system components can be used to develop DR curtailment strategies:

BAC-HD150: Adapter that enables the CITY MULTI controls using BACnet. It does protocol translation from BACnet to M-NET the proprietary digital communication technology. Native BACnet/IP support on BAC-HD150 allows the HVAC system to be IP compatible, and leverage the existing network infrastructure and facilitate convergence of building automation and IT systems enabling interoperability.

Outdoor Unit: Heat recovery unit that simultaneously cools and heats different zones in a building and allows the control of the compressor frequency and starting the units in sequence.

Indoor Unit: Variable capacity and multimode operation: fan, cool dry, auto, heat which is usable in control strategies, through zonal on/off sequencing and turning selective units.

Chapter 4 details the examination of the bridge OpenADR client that translates OpenADR events to BACnet commands for different DR strategies.

1.5 On-site Battery Storage as Demand Response Resource

Behind-the-meter battery storage as a demand response resource is gaining popularity as it has the potential to provide a cost-effective solution to reduce peak demand or demand charges. To test this hypothesis a small battery test bed was developed with a battery pack rating of 76 V, 40Ah, 3.0kWh. The battery pack has 24 $LiFePO_4$ cells with cell rating of 3.2V, 40Ah, 128Wh controlled by a battery management system.

To make this battery storage system DR compliant, an OpenADR client controller was developed with the capability to parse OpenADR event signal on board and run 'load leveling' strategies by alternately storing and generating energy. This solution can be extended to provide intelligent energy management for a typical residential unit by optimally managing power-from-grid for charging, renewable generation, and to keep critical loads running during grid disturbances.

Development, experimentation, and results of this test rig are discussed in Chapter 5.

1.6 Residential Demand Response Modeling and Simulation

Peak demand reduction potential in residential energy market is very promising and is mostly untapped. Getting a definitive estimate of the controllable load base is challenging due to the granularity, diversity of electrical loads, and diversity in end-use.

In this project a consumer level, low cost simulation tool is proposed to assist in identifying the potential load base. To support this, an *Extensible Markup Language* (XML)-based Electrical Service Model (ESM) was developed. This model is a hierarchic representation of a residential dwelling unit as an electrical demand response resource. The hierarchy consists of the electric service size, circuit and subcircuit size, appliances, plug-loads, lighting, and so forth. The model forms the base for the DR simulation modules in association with end-use preference settings for individual electrical controllable loads. The simulator may be used to run interruptible / curtailable DR events against individual ESM, aggregate ESMs, and aggregate load types from available ESMs. It is also useful as a learning tool.

Modeling ESMs, simulating a DR event run and results of the proposed curtailment service is presented in detail in Chapter 6.

CHAPTER 2:
OpenADR on WINSmartGrid Platform

WINSmartGrid platform used in this project is based on Reconfigurable Wireless Integrated Network Sensors (ReWINS) technology developed in SMERC UCLA (Ramamurthy, Lal et al. 2005; Ramamurthy, Prabhu et al. 2007). ReWINS is a generic smart sensor platform with plug-n-play support for variety of hardware interfaces, payload processing and communications.

Salient features of ReWINS are

- General purpose hardware interface for diverse sensors and actuators that can be customized for an application through firmware downloads
- Data processing infrastructure in the backend to implement APIs and applications

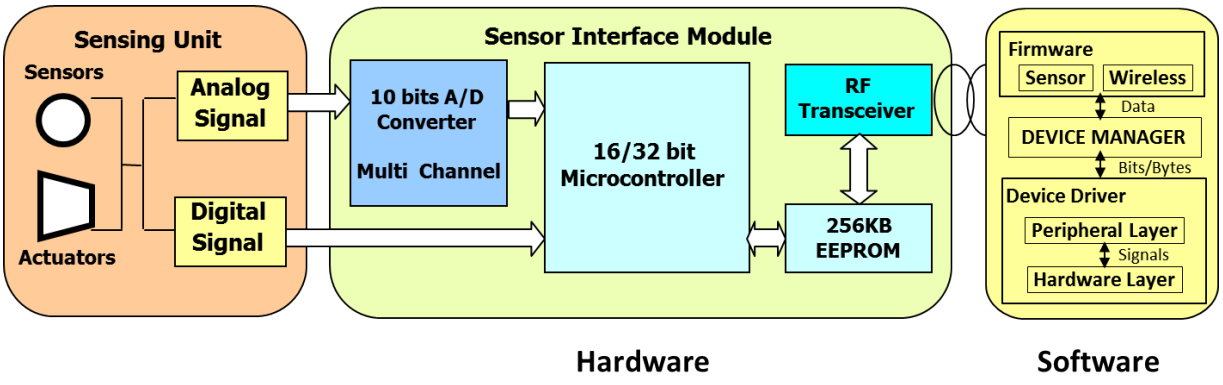
2.1 ReWINS Technology

ReWINS has a modular plug-n-play architecture as shown in the schematic in Figure 3. It has two main components – a sensor interface and a radio frequency (RF) interface. The microcontroller acts as an intelligent interface between them and supports data collection, data processing, and wireless transmissions. The modules are replaceable in the field without disturbing the other modules. The firmware is updatable to support the replaced modules. The updates relate to sensor run-time parameters, microcontroller program variables, and system software modules.

The software stack of the ReWINS consists of three layers as shown in Figure 3:

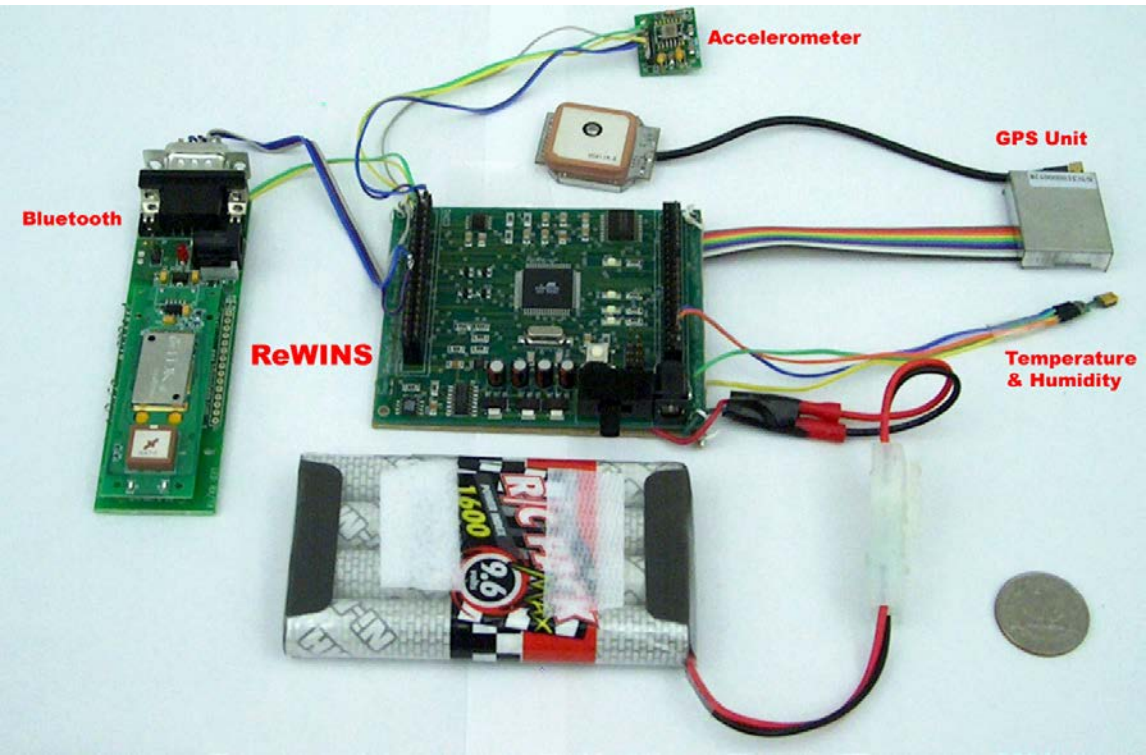
- Device Driver: directly interfaces with the hardware interface and extracts digital data
- Device Manager: interfaces with the device drivers and exposes a multiple-data channel interface to the firmware layer
- Firmware: synthesizes the data by fusing the data from multiple channels and provides application specific functionalities like real-time performance and data communications protocol with central control unit

Figure 3: Schematic of ReWINS Platform



One of ReWINS configurations is shown in Figure 4. In this configuration, the platform is setup to monitor an accelerometer, a GPS unit, a temperature sensor, and a humidity sensor. The wireless communications is provided by Bluetooth modem. This configuration was used in demonstrating remote monitoring of perishable goods, such as food products and pharmaceuticals. In this configuration, it logs time, temperature, humidity, acceleration, location and raises alerts based on preset thresholds.

Figure 4: ReWINS Hardware for Monitoring Perishable Assets



2.2 WINSmartGrid Technology

UCLA developed created WINSmartGrid to extend ReWINS technology to support remote intelligent electrical load controls by implementing the firmware architecture on a commercial off-the-shelf single board computer (SBC). The design has progressed through multiple SBCs like Arduino and Raspberry Pi. SBCs are single circuit boards with integrated processor, memory, input/output pins or slots, communications, and video interfaces that operate at low power, are small sized, cost-effective and provide full-sized computer's functionality. They are capable of running multiple flavors of operating systems selectable as required by the applications and support embedded databases for high-performance data logging needs. This configuration and features of SBCs has proved an ideal control platform for specific WINSmartGrid embedded applications being prototyped.

In the current version of the technology, the Home Area Network (HAN) control is provided through two WINSmartGRID platforms called the Smart Gateway and the Appliance

Controller. In addition to the capabilities of ReWINS, WINSmartGrid platform firmware has an embedded OpenADR client. The smart gateway connects to the network and monitors the availability of a DR event. On receiving a valid DR event, the smart gateway manages the preset load management protocol of all the appliance controllers connected to the smart gateway in the HAN setup.

2.2.1 Smart Gateway

The smart gateway is implemented using the SBC with Wi-Fi and a ZigBee dongle. It includes a real-time clock and keeps track of time using network time protocol (NTP) connected to a public NTP server. The gateway connects to the main server using secure shell (SSH), which is left open for secured and encrypted communication; thus allowing a channel for PUSH and PULL interaction methods for data exchange and updates to the firmware.

Smart gateway connects the HAN to external network systems, such as the utility company's network, and manages all devices within the HAN. The gateway has six primary functions: network communication and appliance management, storage of electricity usage records and electrical load states of appliances, privacy protection of the end users' energy data, DR signal response generation, and consumer preferences management. Figure 5 shows the smart gateways being tested and Figure 6 shows a typical preference selection screen.

Figure 5: WINSmartGrid Smart Gateway

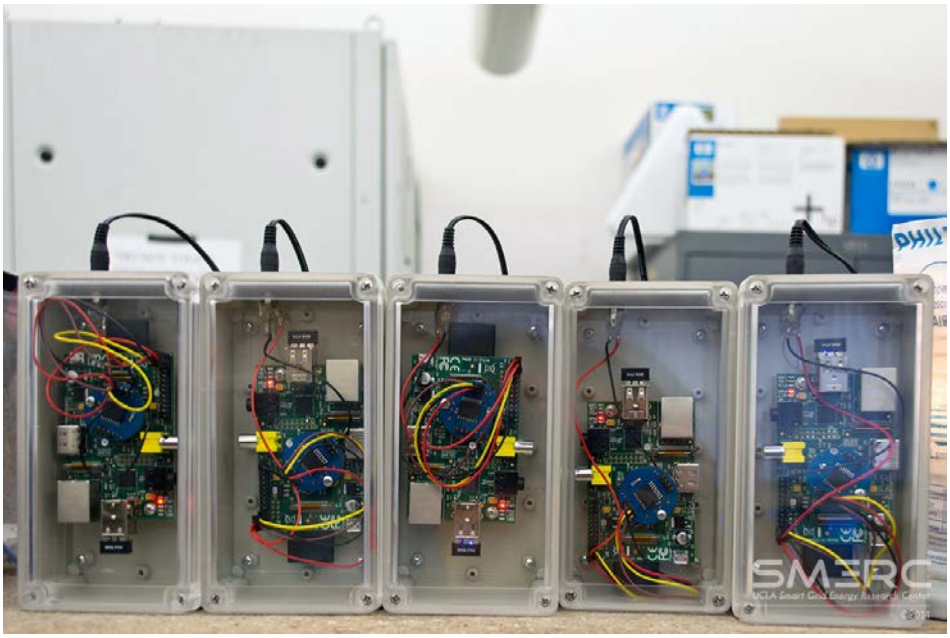
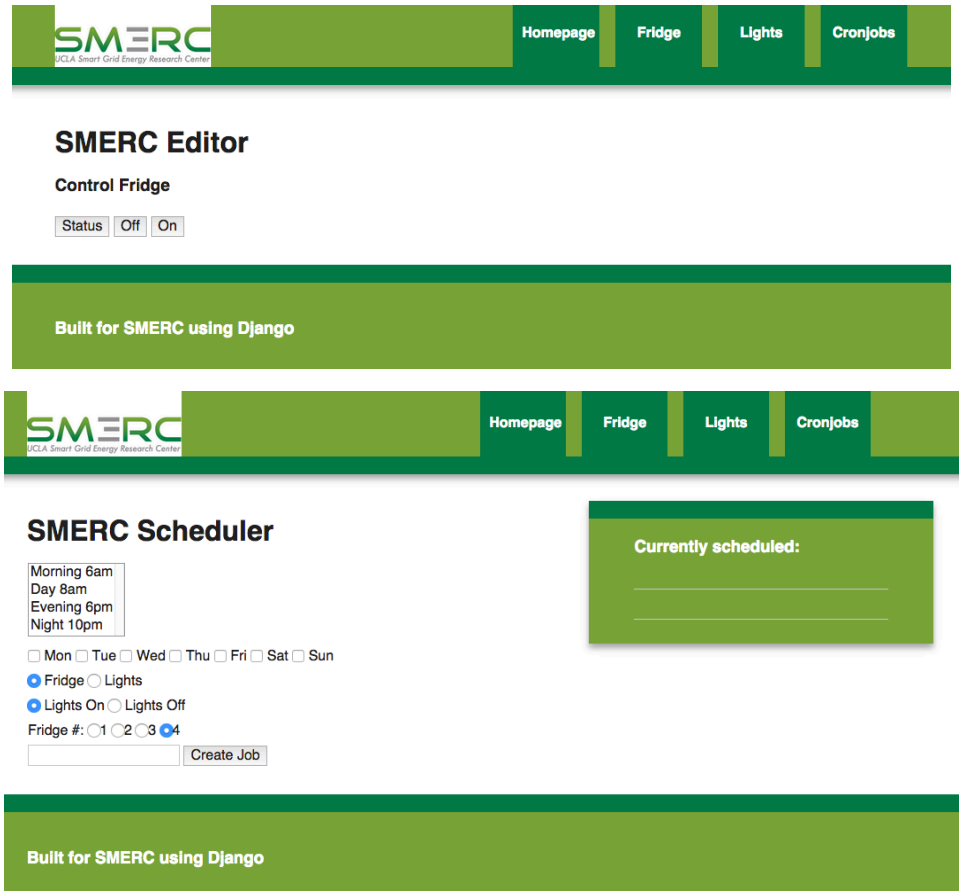


Figure 6: Preference Manager



The gateway maintains two-way communication between the HAN's end devices and external systems, in the utility company or campus facility using onboard Ethernet or Wi-Fi modems. The gateway also acts as a network endpoint inside the customers' premises establishing network connectivity with network-enabled electrical loads, such as appliances, digital thermostats, load control units such as smart plugs 15 A/30A, electrical outlets, and switches.

The gateway collects electricity usage records and individual electrical load states of appliances within the HAN and stores this data on the main server's database., the gateway also collects electricity usage data from the submeter in the controllers in intervals of as low as a second and communicates that information to the utility for monitoring; this data is stored by the gateway and the main server's database. For the electric load states of appliances, the appliance controller of each end-device communicates with the smart gateway to transmit the device's energy status. The data from the smart meter and appliance controllers are used by the gateway to manage energy consumption profiles of registered loads within the HAN. In addition, the gateway protects consumer privacy by limiting the consumer's energy data that the utility company can see.

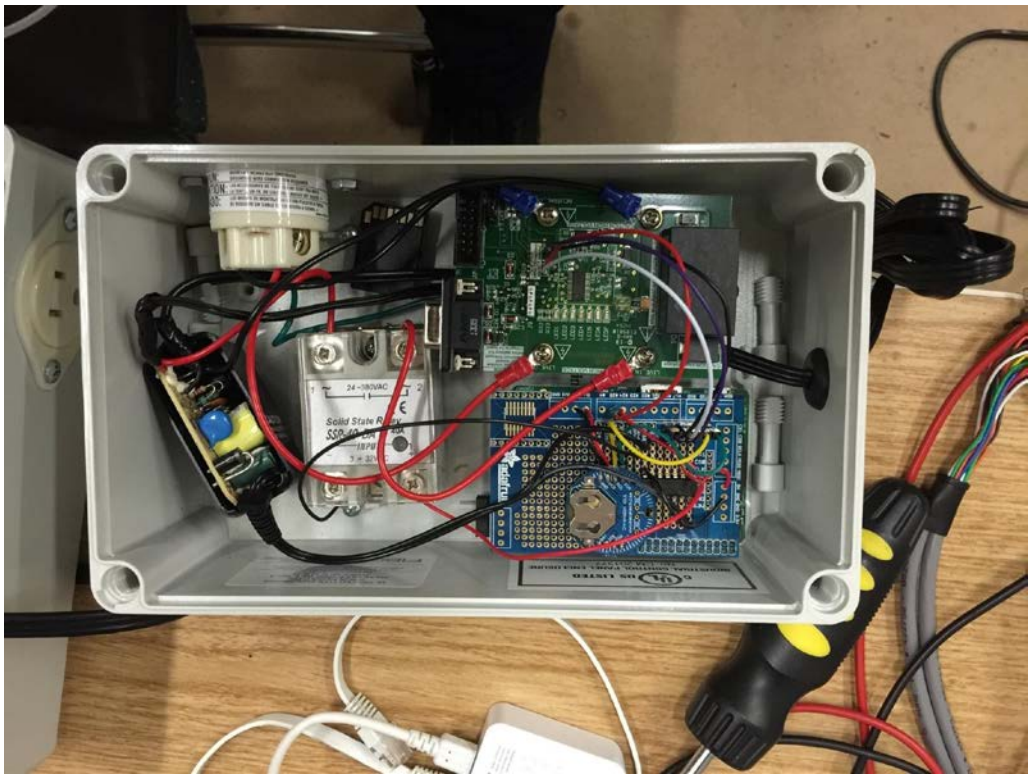
Whenever the gateway receives a DR event signal, it generates a translated DR signal for each appliance. The translated DR signal stores the following data: timestamp when the signal is generated; duration of the DR event in seconds; and timestamps for when the curtailment event goes into effect and when it ends. These data are stored in the translated demand response signal that the gateway pushes to the end device's appliance controller for fulfillment of the load curtailment based on consumer preferences. Once the curtailment ends, the appliance controller collects the energy status information from the appliance and transmits the post-DR event data to the gateway, which pushes the data to the main server.

2.2.2 Appliance Controller

The appliance controller comprises a controller and relay that can be connected to a range of appliances operating at 110V or 240V as shown in Figure 7. The controller communicates with the smart gateway and sends appliance control signals to the relay. Its design is similar to the gateway, containing a single board computer with a real-time clock, Wi-Fi, and ZigBee dongle. Its enclosure includes an Ethernet port and a power connector. The relay receives appliance control signals that it uses to manage the internal circuits of the appliance. It encloses a relay module and a microcontroller that is programmed to take commands via Universal Asynchronous Receiver / Transmitter (UART) and adjust the relays accordingly.

The appliance controllers are connected directly to the HAN's appliances using DB9 interface to provide power and UART communications and provide several functions. First, they establish network connectivity between the appliances and the smart in-home gateway using Ethernet, Wi-Fi, and/or ZigBee. The controller also directly modifies the appliance's states and functions to manage the electrical load per the direction of the gateway, in response to a DR event and in compliance with the customer's preferences. Finally, the appliance controller sends updates to the in-home gateway regarding energy consumption

Figure 7: WINSmartGrid Appliance Controller



2.3 OpenADR Clients on WINSmartGrid

The OpenADR client implemented on WINSmartGrid supports two versions namely, smart client and bridge client. The smart client enables direct integration of OpenADR actions to load management actions of the electrical load. Whereas, the bridge client is an intermediate node in the DRAS to electrical load (HAN node) pipeline and enables automating electrical loads by integrating control units and converting OpenADR event data into load management actions or convert OpenADR events into native protocol control signals for managing the loads.

The OpenADR based AutoDR using WINSmartGRID platform is discussed in detail in the next chapter.

CHAPTER 3:
Automatic Demand Response Architecture

The Automated Demand Response architecture used in the current report describes the elements and their relationships of an automated demand response system that enables reduction in electric load during periods requested by utilities or facilities with minimal to no human involvement. The behind-the-meter load curtailment in homes and buildings is directly initiated by an external signal from the utility or facility through preprogrammed response strategies. This curtailment is made possible by deploying discrete, automated grid- and network-friendly control nodes inside the homes and buildings that convert the utility signals to electricity-use load modifications.

3.1 Open Automated Demand Response

AutoDR is Open Automated Demand Response (OpenADR) communications specification compliant. According to OpenADR Alliance, "OpenADR is an open and interoperable information exchange model, where the message format is standardized so that dynamic price and reliability signals can be delivered in a uniform and interoperable manner amongst the stakeholders"(Ghatikar and Rolf 2011; Piette et al. 2009).

Even though the information exchange model is standardized, automated demand response remains a challenge. The end-to-end grid- and network-friendly automation pipeline is fragile because of a variety of reasons. Chief among them, based on UCLA's research, is the need for costly retrofits to existing homes and buildings. This is caused by the hardware required for protocol translation, and managing heterogeneity of automation controls to prepare for automated demand response controls with OpenADR.

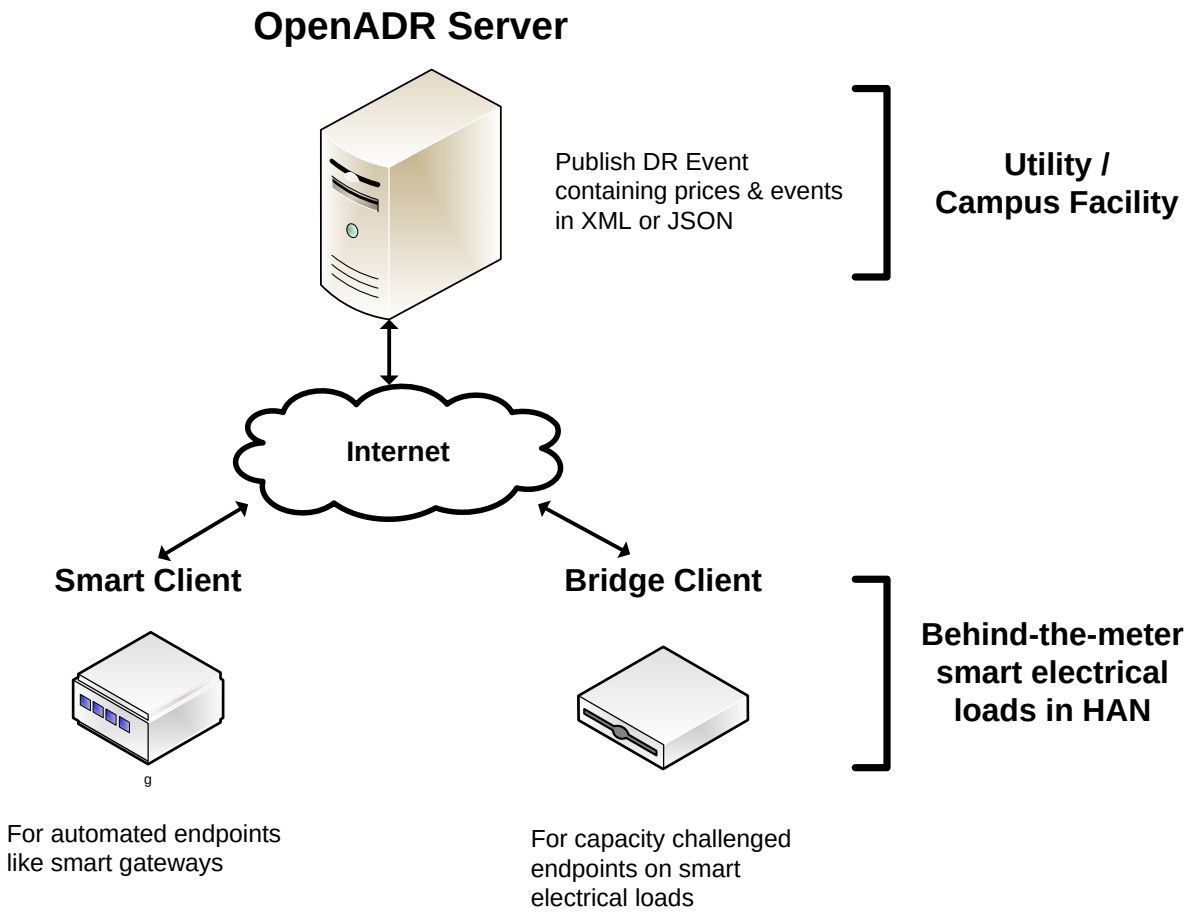
3.2 Demand Response Automation Server (DRAS)

The architecture of the DRAS complies with the OpenADR specification and the interfaces meet the interoperability requirements of an architecture that is integrated and hosted within the UCLA facilities IT infrastructure.

3.2.1 Architecture

The service-oriented, client-server architecture developed by us supports continuous, two-way, secure messaging between facilities and the participating customers. The client-server service topology of the architecture is as shown in Figure 8.

Figure 8: REST Client-Server Service Topology of the Architecture



Important components of the system that support the test-bed are the service-based technology stack, the DRAS server, smart clients, and the bridge clients.

Technology Stack: The AutoDR server-side solution has been implemented using Windows Communication Foundation (WCF), a very flexible, reliable, secure, transacted and interoperable framework from Microsoft that supports building service-oriented applications. The feature being the ability to send data as asynchronous messages from one service endpoint to another and the server-side endpoint can be part of a continuously available service hosted on Internet Information Services (IIS), the web server in Windows environment. The flexibility of creating the messages has been used to package the DR event in XML or JSON by the same service interface. The salient features of the framework for this solution as developed by us are:

- Asynchronous messaging that facilitates *fire and forget* communication where the client and server need not wait for request-response processes
- Tightly integrated security provided by the authentication and authorization security context to messages

- Demonstrated performance improvement over standard web services
- Cross-platform support where each platform works with its native data type.

WCF provides a unified software platform for developing distributed architecture that is a requirement of AutoDR.

DRAS Server: Manages the DR event information model containing prices and curtailment values. The server delivers the model in XML or *JavaScript Object Notation* (JSON).

Smart Client: Control node of Home Area Network (HAN) with onboard OpenADR client that directly receives and acts on OpenADR signals to initiate changes in electricity consumption.

Bridge Client: Intermediate node of HAN that receives OpenADR signal and converts it into DR event operation modes and statuses that effect changes to the electricity consumption of the electric load.

Features of the architecture are:

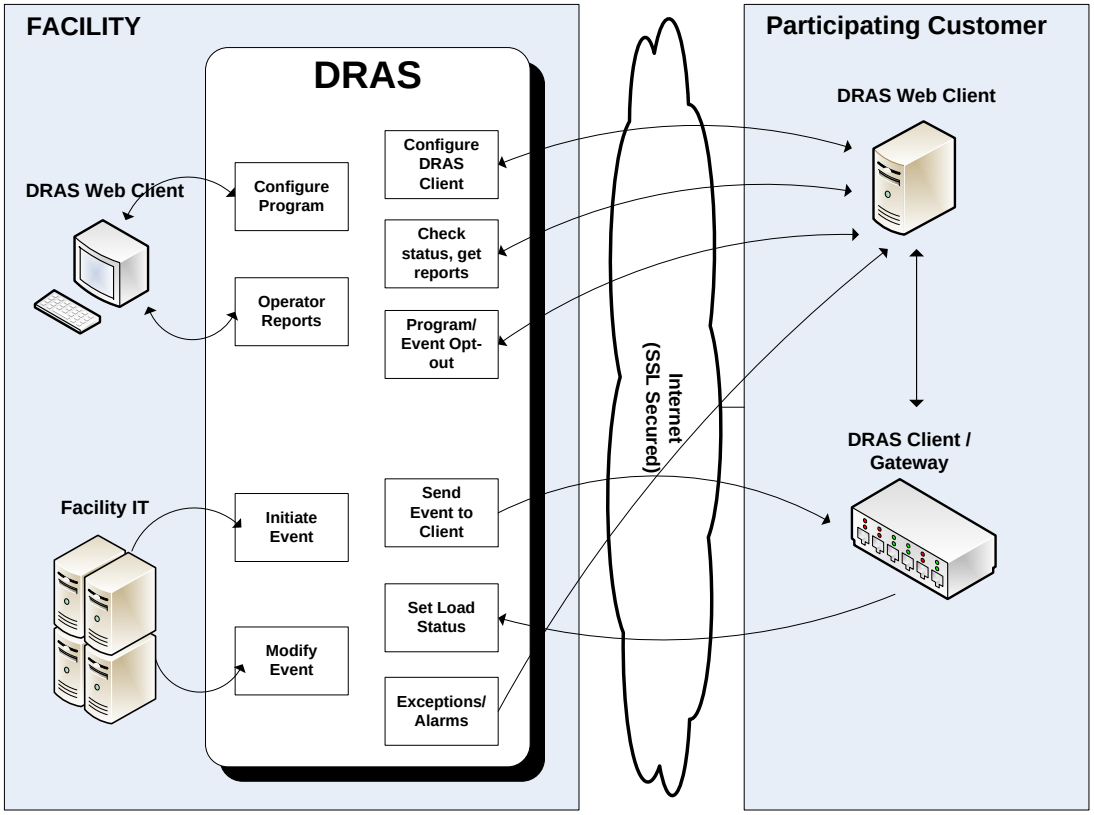
- Signaling / Messaging – Facilitate continuous, 2-way, and secure communications
- Signal Translation – Translate DR event information to Internet signals compliant with the customer facility's network and communication technologies.
- Automation – Initiate load change action on receipt of DR activation message utilizing the preprogrammed DR strategies that is selectable and controlled by the customers
- Notification – The timing of notification of the events may be defined by facilities, providing the customers end-use participation strategies

The architecture has two distinct interfaces:

- Utility/Facility interfaces, and
- Customer / DRAS client interfaces

The diagram in Figure 9 depicts the schematic of the DRAS service topology.

Figure 9: OpenADR-based DRAS Architecture Showing Facility and Customer/DRAS Client Interfaces



This DRAS architecture used by us facilitates communications and messaging between facility IT and the participating customers who manage the consumption of electricity. In this architecture the main messaging payload would be data and information related to emergency and reliability.

3.2.2 Functional Description of the Architecture

The intention of the OpenADR specification is to specify various functions that exist in a complaint DRAS. It is not intended to specify the implementation technology or the details of each of the functions that support the different interfaces as shown in Figure 9. The specification does not require how the information is stored internally or specify the design of the database schemas. The user interface of the interfaces is also implementation specific, whereby it can be customized as per the requirements of the service provider(s) and the planned DR programs.

Overall Functional Requirements

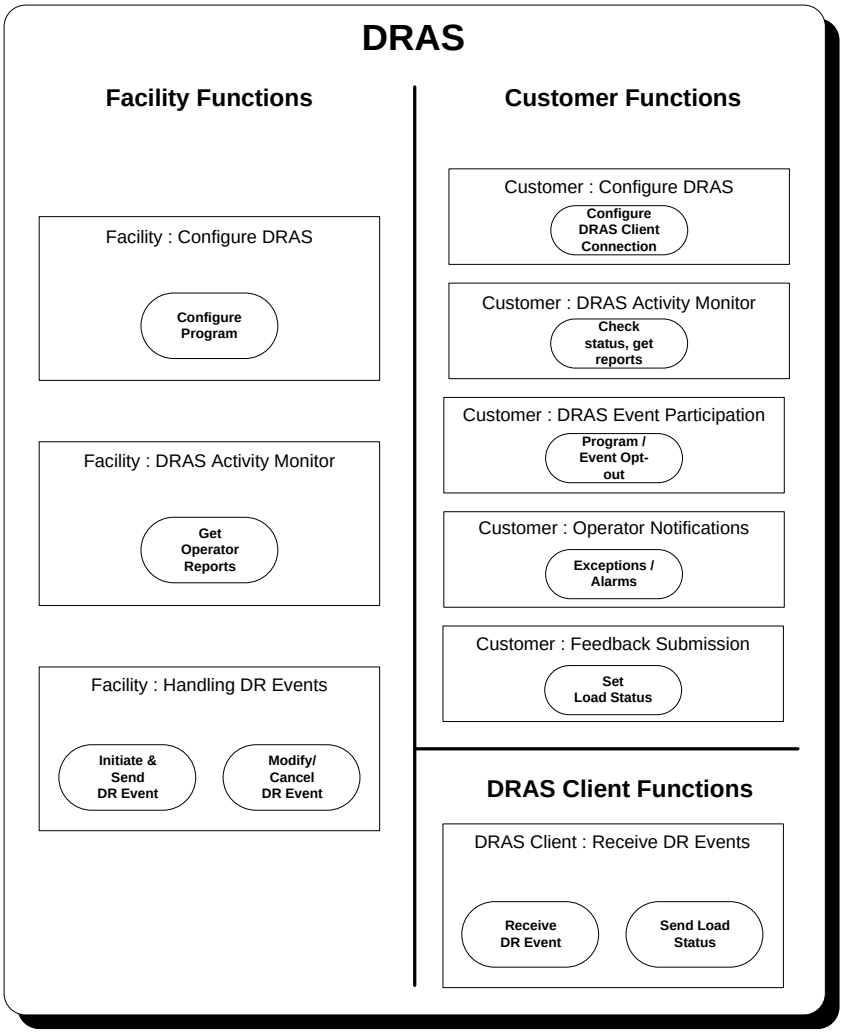
Functionally the DRAS system used complies with the following general requirements:

- The message and data exchanges adhere to security policies to insure all exchanges are authenticated, encrypted and maintain integrity.

- Latency of the DR event sent from server to clients is minimum
- DRAS maintains accurate time within 15 seconds
- Provide for gradual recovery from faults, minimizing data loss

The main subsystems/functions of the DRAS architecture are shown in Figure 10.

Figure 10: DRAS Sub-Systems and Functions



Utility-side Functions

The following are the functions of the utility/facility-side interface of the DRAS architecture:

1. *Utility DRAS Configuration*: Utility/facility sets up the DR program, and signs up participants to create a customers' record based on the record existing in the CIS for each participating customer. The program information that is included is –
    - Program definition:

    - Schedule – time-of-day, duration, and so forth
    - Type of event (levels) of the DR event
- Utility assigned account number
- Participant identification
- Password for participant
- Location

2. *Utility DRAS Activity Monitor*: Utility operator gets the operation reports as follows -
    - Communication logs
    - Current status of DRAS client
    - Last contact with clients
    - Signal logs
    - Participants manual settings (opt-out status)
    - Event status
3. *Utility DR Events' Handling*: Utility operator schedules and initiates an event. The information sent to DRAS would include –
    - Program type
    - Date and time of event
    - Date and time issued
    - Location
    - Participation list (list of customers/account numbers)

The event notifier of DRAS sends this information to the registered DRAS clients. The utility/facility operator sometimes decides to modify or cancel an already issued DR event.

Customer-side Functions

The following are the main functions of the customer-side interface of the DRAS architecture:

Customer DRAS Configuration: The customer configures the DRAS program parameters and the DRAS client connection information that would include –

- Participant ID and password
- Contact information (phone number, email)
- DRAS Client communication parameters:
    - DRAS IP address

    - Identification
    - Password
    - IP connection information
    - Polling frequency (if client polls)
- Load reduction potential (load use preferences per time per event level)
- Exception parameters
- Opt-out settings

Customer DRAS Activity Monitor: The DRAS provides the participant the following information -

- Event status
    - Current Outstanding Events
        - Load reduction potential from the aggregated impacted participants
        - Feedback from DRAS clients
    - Event logs

Customer Event Participation: This function enables the customer to opt-in / opt-out of the DR program. The opt-out would be for either the entire DR program or a single event. When in opt-out condition the DR events are not propagated to the DRAS client.

Customer Operator Notifications: DRAS to communicate any exception notification in case of errors or break in operation. These exceptions include machine and platform specific errors and communication failures.

Customer Feedback Submission: The feedback mechanism sends information to DRAS. This mechanism is used to record how the customer responded to the DR event. Data recorded includes –

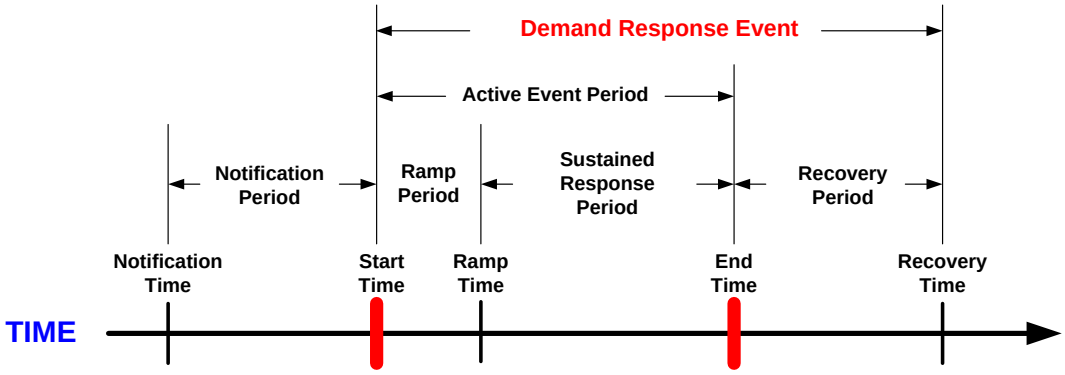
- Program identifier
- Customer identifier
- Date and time of the event
- Shed data in kW / kWh
- Real-time load
- Specific load reduction (Heating, Lighting, HVAC, Appliances – for opt-in customers)
- Event type

DRAS Client Reception of DR Event: The DRAS client / Gateway propagate the DR event signal to each of the controlled loads for load shed, shift or shape action. The status of the load under the active event is relayed back to the DRAS.

3.2.3 DR Event

On the UCLA campus, most of the buildings and especially the student dormitories are not metered individually hence, DR experiments focus only on load curtailment and not price-responsive event. The DR event is broadcastable in both XML and JSON formats. Figure 11 shows the structure of a DR event generated by DRAS. A DR Event is a signal from the utility or facility to demand response resources requesting load curtailment starting at a specific time, for a specified duration. Appendix A provides sample DR events generated by the DRAS server.

Figure 11: DR Event Structure



3.2.3.1 DR Event Periods

A DR event has distinct periods that demand different actions from the participating loads as per specifications and implemented in the architecture. The periods are:

Notification: is a time prior to the actual start of the event when curtailment is expected, during which time the participation in the event is adjustable by means such as substituting the participating loads and opting out of the event.

Active: is the time when the actual curtailment or change in load profile is expected from the participating resource.

Recovery: is the time during which the participating demand response resource(s) can recover from the curtailment.

3.3 Experiments and Results

To test the DR architecture, DRAS server, OpenADR clients, and different DR programs, the following main tests were run. These test cases were executed concurrently in response to every DR event broadcast by the DRAS server.

1) *Establishing connection between the DRAS server in SMERC with the BAC-HD150 adapters*

The goal of this test is to examine the connection between the DRAS server and the BACnet adapter through the OpenADR bridge client, and ensure curtailment actions on receipt of DR event. The curtailment actions are described in Chapter 4.

2) *Establishing connection between the DRAS server in SMERC with the storage battery control*
The goal of this test is to examine the connection between the DRAS server and the battery control, and ensure load-leveling actions by the battery pack on receipt of DR event. The 'load leveling' actions are described in Chapter 5.

Results of the tests on AutoDR infrastructure to evaluate system capabilities, communications reliability, and the robustness of the hardware / software interfaces can be summarized as follows:

Table 1: Test Results of DRAS Server Installation

Installation of DRAS server		
Pre-condition: Setup the server for both HTTP and HTTPS network transport		
Step	Description	Result (Success/Failure)
1.	Install the server's service stack in SMERC	Success
2.	Run DRAS server and monitor event logs	Success
3.	Schedule test DR event and log the event operation modes and states	Success

Table 2: Test Results of DRAS to BAC-HD150 Communications

Test: Establishing connection between the DRAS server in SMERC with the BAC-HD150 adapters and participate in DR events		
Precondition: BAC-HD150 installed in private network with restrictive TCP/IP port access and IP address of the bridge OpenADR client whitelisted in the firewall		
Step	Description	Result (Success/Failure)
1.	Create client account for the HVAC demand response resource	Success
2.	Install bridge OpenADR client on client machine in SMERC	Success
3.	Schedule test DR event with notification period, active period and event retirement	Success
4.	Receive DR event at bridge OpenADR client (event logged to file)	Success
5.	Install bridge OpenADR client on client machine in Hilgard network (HVAC private network)	Success
6.	Schedule test DR event	Success
7.	Receive DR event at bridge OpenADR client (event logged to file)	Success
8.	Poll status of BAC-HD150	Additional testing Note: network connectivity has issues*

*BAC-HD150 has been deployed on a private UCLA facilities network and the traffic flow from edge-node machine in SMERC to BAC-HD150 is one-way. The response from BAC-HD150 is being blocked by the facilities network.

Table 3: Test Results of DRAS to Battery Rig Communications

Test: Establishing connection between the DRAS server in SMERC with the storage battery control and participate in DR events		
Precondition: Deployed under TLS/SSL protection and client-server exchange is over HTTPS transport		
Step	Description	Result (Success/Failure)
1.	Create client account for the storage battery demand response resource	Success
2.	Install and connect the controller with smart OpenADR client	Success
3.	Schedule test DR event with notification period, active period and event retirement	Success
4.	Receive DR event at smart OpenADR client	Success
4a.	Poll for DR event with different schedules - 15 min., 5 min, 15 secs	Success
5.	Test load islanding with 'HIGH' DR event operation mode	Success
6.	Test charge/discharge with 'MODERATE' DR event operation mode	Success

3.4 Observations

The AutoDR software service-oriented architecture based on REST model proved well suited for AutoDR applications in most instances. Following are the observations of the architecture:

- Lightweight nature of services proved appropriate for the constrained smart controls like the smart gateway and appliance controller.
- Explicit use of HTTP methods simplified platform and language-independent implementation.
- Client-server requests and responses could be designed to be lightweight reducing network traffic.
- Comparison of DR event and control node responses in XML and JSON showed that JSON format reduced the payload size by 40 to 45 percent, again minimizing network traffic. However, this also implies the existing commonly used format of XML has an inherent payload size disadvantage that could become important in the context of fast DR in which delays need to be kept at a minimum.
- System fidelity was very satisfactory and message exchanges during AutoDR tests between the smart as well as bridge AutoDR clients were complete and reliable.
- One of the observations is pertaining to enterprise security when implementing AutoDR. At various points in the project, certain access types such as TCP ports, service URIs, access through firewalls, proxies and load balancers were blocked due to security reasons and traditional models where the entire network, security and access policies being handled by a single IT department as in the case of private enterprises. Getting

accesse is sometimes an organizational challenge. This challenge is a roadblock for rapid AutoDR scale up, which has more of an open nature than traditional enterprise security.

CHAPTER 4:
Small Commercial & Industrial HVAC System as Demand Response Resource

This project used a Mitsubishi CITY MULTI VRF HVAC system for the research, which is a sophisticated system, combined with high level of granular control and energy efficiency. The system comprises of a condensing unit located externally and a series of multiple indoor units to provide zonal cooling and/or heating. Salient feature of the system being the only 2-pipe heat recovery simultaneous heating and cooling system in the market. The configuration of the installed HVAC system in the test bed is shown in Figure 12. Specifications of the HVAC configuration are listed in Appendix B.

Figure 12: CITY MULTI HVAC System Configuration with BAC-HD150 Connected to M-NET

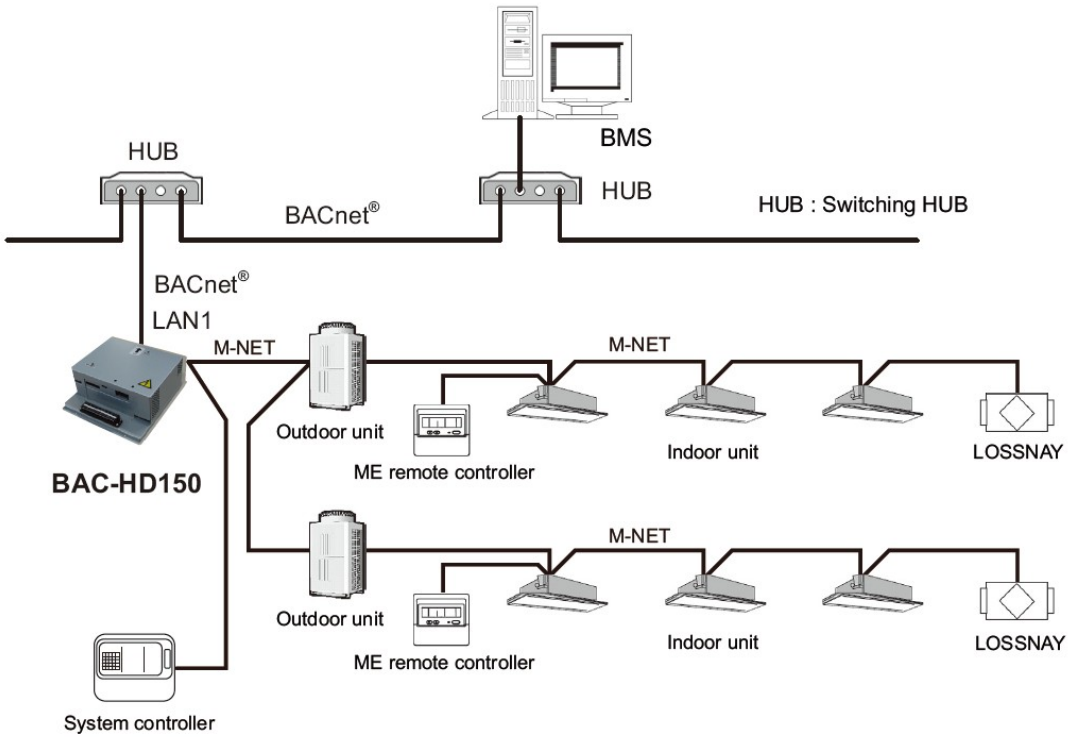


Image Credit: BM Adapter BAC-HD150 Instruction Book, Mitsubishi Electric Corporation, Dec 2009

4.1 BAC-HD150 Adapter Functions

BAC-HD150 adapter does the protocol translation from management module to BACnet commands and vice versa. It converts DRAS signals to BACnet commands transmitted over M-NET to the air-conditioning system, and in the reverse path collects status data from the air conditioning system and transmits it to the management module. To interface the BAC-HD150 adapter with the air conditioning system, the indoor units are defined into group configurations, which consist of three main numbers:

1) M-NET number : Identity of the BAC-HD150 module and it is fixed to 01
2) Group number : Identity of the target group consisting of multiple indoor units. Valid group number range is 1-50
3) Member number : Identity of the monitor/operation item of air conditioner

Figure 13 shows the relationships.

Figure 13: CITY MULTI HVAC Group Configurations with BAC-HD150 Connected to M-NET

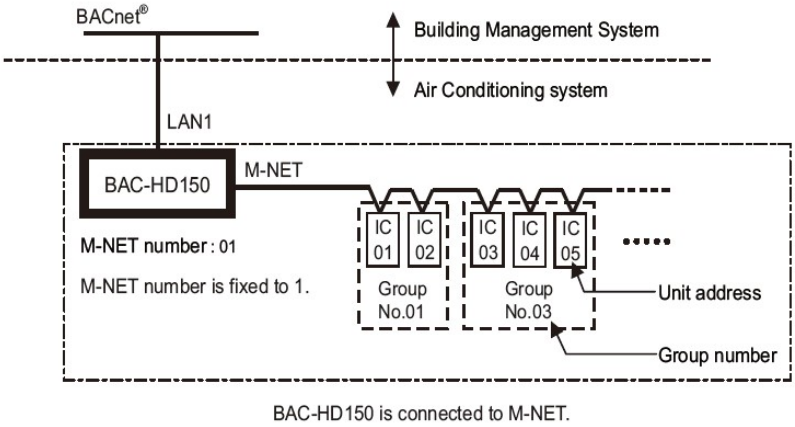


Image Credit: BM Adapter BAC-HD150 Instruction Book, Mitsubishi Electric Corporation, Dec 2009

4.2 BACnet Commands Generated in Response to DR Events

In response to a DR event, the bridge client translates the DR signal to BACnet commands. The BACnet communications protocol based on user datagram protocol (UDP)/IP used in the translation is as follows:

Table 4: BACnet Protocol Packet Structure

Ethernet	IP Header	UDP Header	BVLL	NCPI	APDU
Physical layer: Ethernet *Transmission medium*: 10BASE-T	*Class C private address is recommended.* *(*1) Subnet Mask*: 255.255.255.0 *1: Recommended range: [192.168.1.1] - [192.168.254.254] Do no use [192.168.0.0] and [192.168.255.255] as a device address.	*The default UDP port of unicast and broadcasting*: 47808 (0xBAC0)	*BVLC type (1 octet)*: Fixed to 0x81 (BVLL against BACnet®/IP) *BVLC function*: (1 octet) Unicast 0x0A *Broadcast*: 0x0B *BVLC length*: (2 octets) *Variable (BVLL header*: (4 octets) + NPCI data length + APDU data length)	*Version (1 octet)*: Fixed to 0x01 *Control (1 octet)*: Response is received. 0x04 *No response*: 0x00	*Data*: 1024 octets or less

4.3 Experiments and Results

The HVAC system with current configuration shown in Figure 11 provides a number of ways to curtail loads in response to a DR event. The following tests were attempted:

1) *Reducing compressor frequency (speed)*

   The goal of this test is to operate the compressor at reduced capacity to maintain minimally effective environmental conditions.

2) *Zonal on/off sequencing*

   The goal of this test is to sequentially switch on/off zones to minimize discomfort by minimizing temperature changes

3) *Turning off indoor units selectively*

   The goal of this test is to turn off one or more indoor units and allow conditioning (temperature/humidity) to drift from adjacent zones

4) *Minimize demand power spikes*

   The goal of this test is to start the outdoor condensing units in sequence to instead of simultaneously to spread out the demand spikes

Results of the tests can be summarized as follows:

Table 5: Identification of BACnet Components for the Planned Tests

Tests: **1. Reducing compressor frequency (speed)** **2. Zonal on/off sequencing** **3. Turning off indoor units selectively** **4. Minimize demand power spikes**		
Pre-condition: Identify the Indoor Control Groupings		
Step	Description	Result (Success/Failure)
1.	Identify the BACnet control item, control type, member number	Control items available
2.	Identify the possible object type, instance number and valid property values to control	Valid value ranges identified
3.	Develop Application Protocol Data Units (APDU) for the data packet for each of the selected control actions	Individual test packet frames developed

Table 6: Test to Reduce Compressor Frequency

Test 1: Reducing compressor frequency (speed)		
Operate the compressor at reduced capacity to maintain minimally effective environmental conditions		
Step	Description	Result (Success/Failure)
1.	*Frame the command packet* Control item: Fan Speed Control Type: Multistate Input Instance Number: 07	Success
2.	*Setup property values* Possible values: High, Mid2, Mid1, Low	Success
3.	Application Protocol Data Units (APDU) [01, 03, 07, <Value>]	Success Individual test packet frames developed

Table 7: Test to Sequence Zonal On/Off

Test 2: Zonal on/off sequencing		
Sequentially switch on/off zones to minimize discomfort by minimizing temperature changes		
Step	Description	Result (Success/Failure)
1.	*Frame the command packet* Control item: On Off Control Type: Binary Input Instance Number: 02	Success
2.	*Setup property values* Possible values: On, Off	Success
3.	Application Protocol Data Units (APDU) [01, 05, 02, <Value>]	Success: Individual test packet frames developed

Table 8: Test to Turn Off Indoor Units Selectively

Test 3: Turning off indoor units selectively		
Turn off one or more indoor units and allow conditioning (temperature/humidity) to drift from adjacent zones		
Step	Description	Result (Success/Failure)
1.	*Frame the command packet* Control item: On Off Control Type: Binary Input Instance Number: 02	Success
2.	*Setup property values* Possible values: On. Off	Success
3.	Application Protocol Data Units (APDU) [01, 05, 02, <Value>]	Success Individual test packet frames developed

Table 9: Test to Minimize Demand Power Spikes

Test 4: Minimize demand power spikes		
Start the outdoor condensing units in sequence to instead of simultaneously to spread out the demand spikes		
Step	Description	Result (Success/Failure)
1.	*Frame the command packet* Control item: Operational Mode State Control Type: Multiple Input Instance Number: 06	Success
2.	*Setup property values* Possible values: Cooling, Heating, Fan, Auto, Dry, Setback	Success
3.	Application Protocol Data Units (APDU) [01, 03, 06, <Value>]	Success Individual test packet frames developed

Table 10: Various Load Curtailment Strategies Tested

Load curtailment strategies tested		
The selection of the curtailment strategy and amount of curtailment in response to DR event operation modes were tested as follows:		
Results for DR event with different operation modes and an active period of 60 mins.		
DR Operation Modes		
MODERATE	**HIGH**	**SPECIAL**
Test 1: Reduced compressor frequency to 'Mid2' level every 0.25 x Active period of DR event	Test 1: Reduced compressor frequency to 'Low' level every 0.25 x Active period of DR event	Test 2: 4 zones were cycled between On/Off, every 30 mins of Active Period of DR event each
Test 2: 4 zones were cycled between On/Off, 0.25 x Active Period of DR event each	Test 2: 4 zones were cycled between On/Off, every 10 mins of Active Period of DR event each	Test 4: Start the 3 outdoor units in sequence Change cool/heat setpoints by 4 units
Test 3: 10-12 indoor units were cycled 0.25 x Active Period of DR event in alternate pairs	Test 3: 10-12 indoor units were cycled 10 mins of Active Period of DR event in alternate pairs	

All the tests were successful from the point of converting the OpenADR event to BACnet control packets by the bridge client. Sample BACnet commands for each of the load curtailment strategy are presented in Appendix C. The actual HVAC control groups curtailment is yet to be completed due to networking issues of UCLA network access policies and the shortcomings of the BACnet adapter of CITY MULTI system.

4.4 Observations

The AutoDR application for a VRF HVAC system proved successful in converting OpenADR event to BACnet signals. The curtailment strategies for different DR operation modes potentially work. The participation data was logged throughout the duration of the DR event, successfully demonstrating the HVAC system as a viable demand response resource on UCLA campus. The actual load curtailment utilizing the different components of the HVAC system has to be proven and the following issues may have to be resolved:

- Fixed program of cycling the zones or internal units will have to be refined based on the comfort factor of the tenants, which may arise due to the location of the zones and/or internal units and the ambient conditions.
- Time duration of each of the curtailment strategies will have to be studied again from the comfort factor and operational restrictions of the HVAC system warranting the time duration may have to be dynamic based on the ambient condition thresholds.

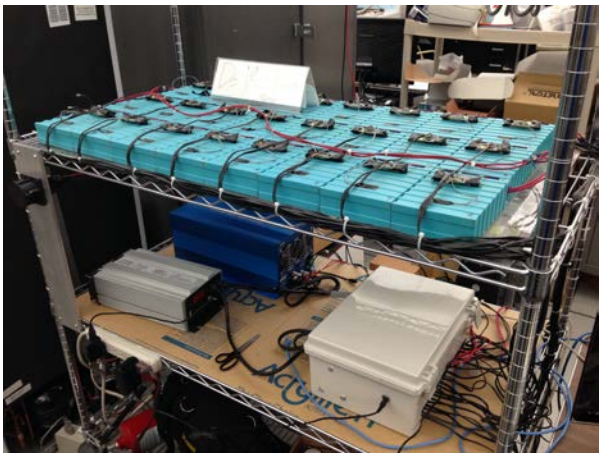
CHAPTER 5:
Battery Storage as Demand Response Resource

The battery storage and management system serves as secondary source of power for an islanded alternating current (AC) load. The system polls the server for a DR signal, once active the process is similar to that of an uninterruptible power supply, but in this scenario provides power to an islanded AC load using a pure sine wave power inverter. When there is no DR event, the system charges and balances the batteries. It also takes advantage of a preference settings system, should the user prefer charging the batteries later on in the evening, or providing a separate AC source during the afternoons. The proposed design provides storage solution for a single building independently from the status of the grid.

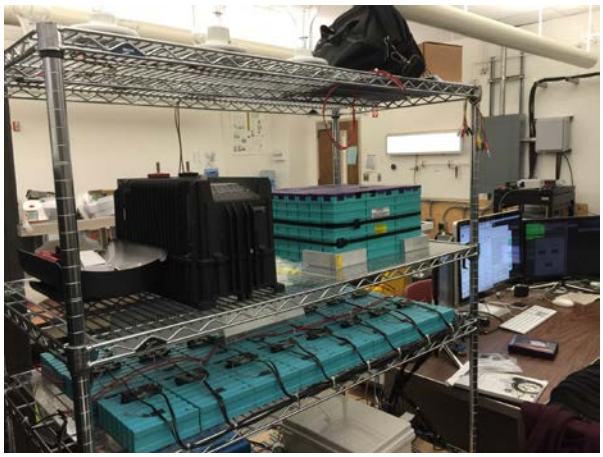
5.1 Battery Rig

The battery rig includes 24 LiFePO4 cells (specifications in Appendix D), battery management system, smart charger, and a grid-tied inverter. The setup is as shown in Figure 14. The battery management system serves as secondary source of power for an islanded AC load. The system polls the server for an OpenADR signal, once active the process is similar to that of an UPS, but in this scenario provides power to an islanded A/C load using a pure sine wave power inverter. When there is no Demand Response event the system charges and balances the batteries.

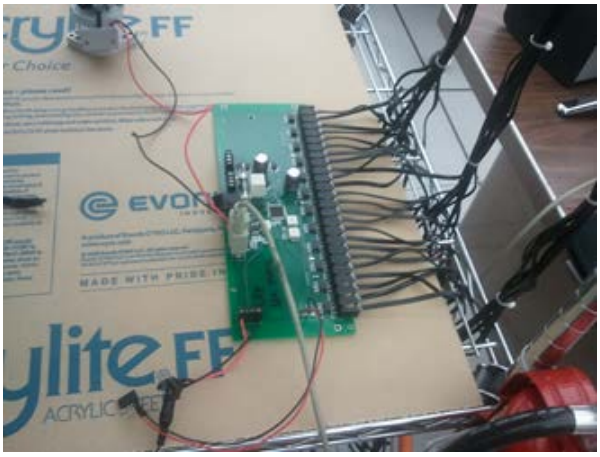
Figure 14: LiFePO4 Storage Battery Test Rig



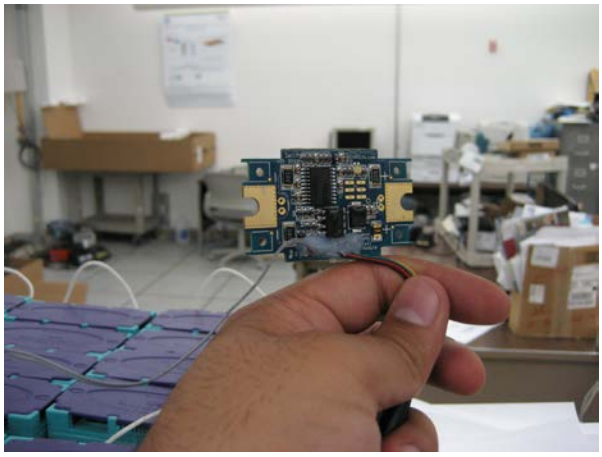
Connected battery cells



Battery cells and Inverter



BMS Controller

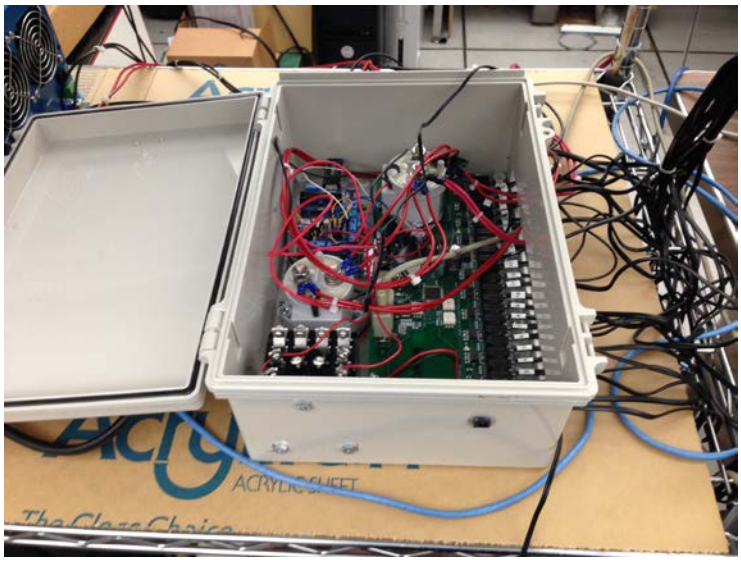


Cell Sense Board

5.2 Automatic Demand Reponses Controller

The Automatic demand response controller has been developed on a commercial-off-the-shelf (COTS) single board computer. The battery pack is integrated with the mains connection for the electrical load using 4-pole double throw switch controlled by the controller. The controller is also integrated with an inverter to convert DC power of battery pack to AC power. The OpenADR client resides in this controller and facilitates the direct parsing of the DR event to the battery resource control actions. Figure 15 shows the assembled controller.

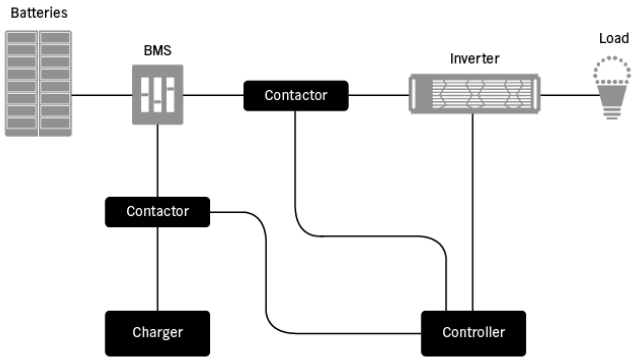
Figure 15: Auto Demand Response Controller with On-Board OpenADR Client



5.2.1 Battery Management System

The battery management system (BMS) serves as a secondary power source to drive local AC loads and functions as both a client service and a demand response resource. In both cases, the BMS either recharges the battery cells or discharges energy into the local grid. When the BMS is not providing energy services, then it charges and balances the batteries. When discharging energy, the BMS operates similarly to an uninterruptible power supply, but uses a pure sine wave inverter to convert the DC power of the battery pack to AC power. This discharging service is activated differently depending on whether the BMS is functioning as a client service or demand response resource. Figure 16 shows the system design of the BMS.

Figure 16: Battery Management System



5.3 Experiments and Results

Load leveling using the storage system in response to DR event can be done in a number of ways based on the setup of demand and supply sides of the storage system, and the participation preferences set by the consumer. The following tests will be attempted:

1) *Regulating loads*

   The goal of this test is to regulate the loads up or down in response to a DR event based on the preference settings that may allow partial load to be supplied by the battery and the rest by the grid, or curtail the loads and supply the grid from the battery

2) *Intelligent charging*

   The goal of this test is to charge the battery intelligently when the electricity rates are less, or attempt charging the battery with renewable generation by curtailing the loads during a DR event.

For the client service, users connect directly to the main server using an interface, programmed in Python, to start and stop discharging services. When starting the discharging service, the user enters the amount of energy needed in kilowatt-hours (kWh), the number of minutes the service is needed, and the critical lower voltage of an individual battery cell that indicates the lowest value the user is ready to reach. Using these data, the main server runs several subroutines. First, it compares the available battery pack voltage to the critical lower voltage input by the user. Afterward, it calculates the available kWh against the requested power; based on the results, the server activates the battery pack or leaves it in its current state. If the activation fails due to the critical voltage level or inability to provide energy for the requested time period, the server transmits to the Python interface the maximum time the battery pack can provide energy service. In addition, the user may stop the discharging service via the Python interface, which switches the energy load to the main power grid and returns the BMS to recharging mode.

For the demand response service, the BMS connects directly to the main server, bypassing the gateway, and uses an appliance controller to poll the server for an OpenADR event signal. Whenever a DR event is received, the BMS activates and provides discharging services; consequently, when there is no active DR event, the BMS charges and balances the batteries. The timing of these two functions is modifiable using the consumer preference engine, which in the test bed is run on the server instead of the gateway. For instance, users can choose to charge the batteries only during evening hours or have the BMS provides a separate AC source only during the afternoons. Figure 17 shows the status of the battery cells while charging (no DR event) and Figure 18, discharging (supporting an AC load under a DR event). In both figures, the first eight lines represent battery cells 0 – 7 that are not connected to any source, but are still shown and connected to the BMS. The other lines represent cells 8 – 23, which are connected to the charger and inverter, because the inverter is rated at 42.0-60.0V.

Figure 17: Battery Management System Charging

```
Cell: 8   ################################    3.442 V
Cell: 9   ################################    3.429 V
Cell: 10  ##############################      3.412 V
Cell: 11  ################################    3.446 V
Cell: 12  ################################    3.439 V
Cell: 13  ##################################  3.467 V
Cell: 14  ##############################      3.401 V
Cell: 15  ##################################  3.486 V
Cell: 16  ##############################      3.403 V
Cell: 17  ################################    3.428 V
Cell: 18  ##############################      3.391 V
Cell: 19  ##############################      3.414 V
Cell: 20  ##############################      3.399 V
Cell: 21  ################################    3.433 V
Cell: 22  ##############################      3.411 V
Cell: 23  ################################    3.442 V
```

Figure 18: Battery Management System Discharging

```
Cell: 8   ##################    3.175 V
Cell: 9   ##################    3.178 V
Cell: 10  ################      3.146 V
Cell: 11  ##################    3.164 V
Cell: 12  ##################    3.187 V
Cell: 13  ################      3.142 V
Cell: 14  ##################    3.166 V
Cell: 15  ##############        3.087 V
Cell: 16  ################      3.148 V
Cell: 17  ############          3.061 V
Cell: 18  ##################    3.189 V
Cell: 19  ##############        3.115 V
Cell: 20  ##########          3.036 V
Cell: 21  ##################    3.185 V
Cell: 22  ################      3.129 V
Cell: 23  ##################    3.184 V
```

5.3.1 Time-bound and Preference-Dictated Demand Reponses Curtailment Tests

The preference engine of the smart OpenADR client establishes the context of the battery resource in participating in a DR event. The preferences that are currently supported are:

a) Supply to critical load for 'X' minutes

b) Supply to critical load for 'X' minutes under BMS supervision and no preference settings

5.3.3.1 10-min Curtailment Test

The battery rig was tested successfully for a DR event of 10-minute duration. The preference setting was to supply power to critical load, which is a bank of six 150 W lamps as shown in Figure 19. All cells of battery were fully charged with approximately 3.4 V / cell and total voltage of 80.4 V for the battery pack. Figure 20 shows the readings of the battery cell voltage and cell temperature during discharge mode to supply the critical load and Figure 20 shows the temperatures of the cells. The lamp bank was switched on during the active period of the event

for 10 min and on the completion of the event, the battery bank was switched to charge mode as shown in Figure 21.

Figure 19: Lamp Bank Used As Critical Load Supported by the Battery

Figure 20: Cell Voltage and Amperes During the 10 Minute DR Event

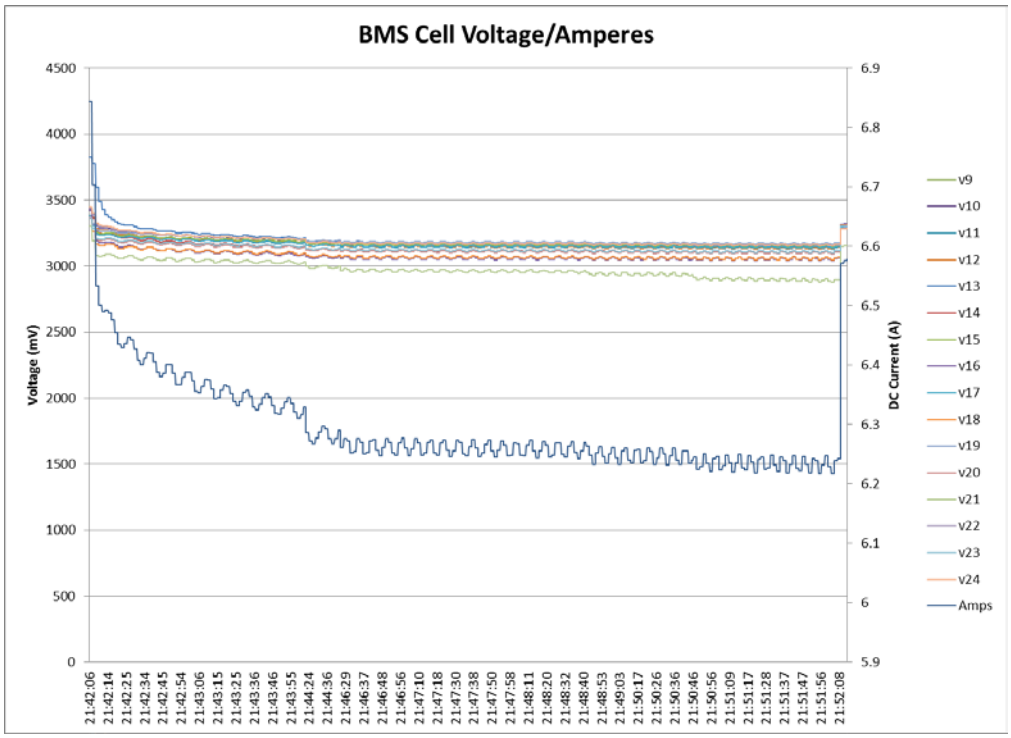
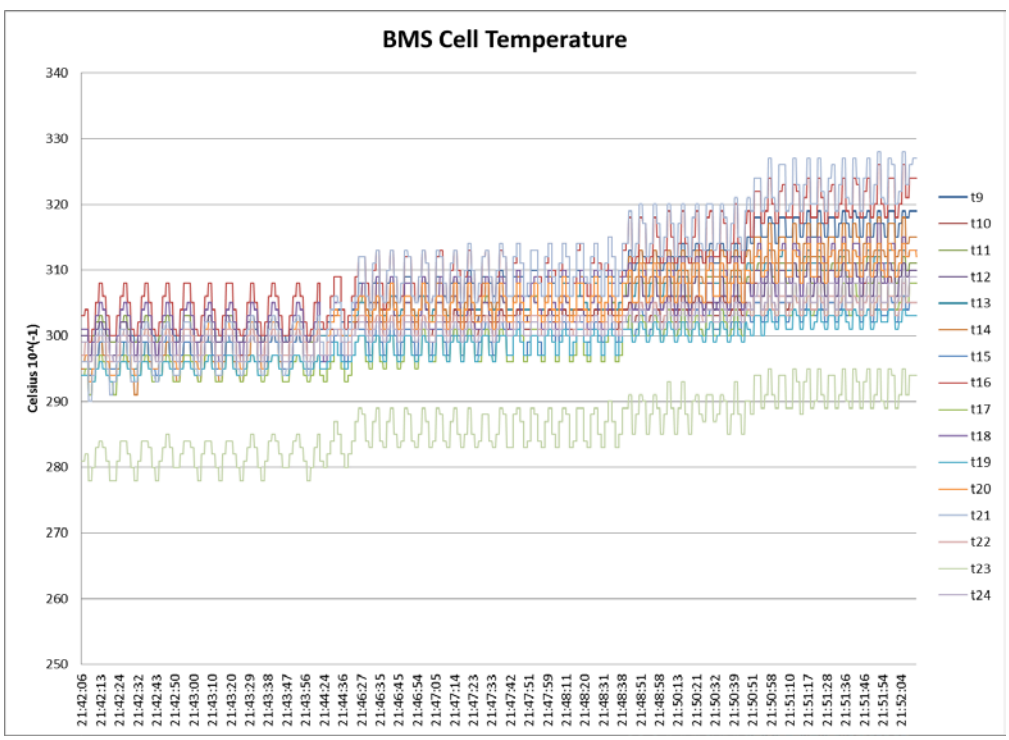


Figure 21: Cell Temperature During 10 Minute DR Event



5.3.3.2 60 min Charge Curtailment Test

Curtailment test for 60-minute duration is presented in Appendix E. The test was successful for a DR event of 60 min when control was switched to supply power to the critical load of six 150 W lamps. All cells of battery were fully charged with approximately 3.4 v / cell and total voltage of 80.4 V for the battery pack. In this test, the battery storage system had preference settings set to supplying critical load and no limits on the battery parameters in the control program. The BMS monitors the cell voltage (3,8V peak during charging, 2.5V cut-off low during discharging), cell operating temperature (-20^0 C to 55^0 C) and battery capacity (40Ah), and a breach of any of these values switches the battery control to safe state. In this test when one of the cell voltages breached 2.5 V, the threshold, the controller opted out of the DR event and switched the battery pack to charge mode.

5.4 Observations

Battery storage for a single residential unit, serving as a demand response resource was demonstrated to be feasible. Though the battery pack used in this project was very small, it is possible to demonstrate relevant control strategies in response to a DR event. With an adequate battery system, it is possible to execute the DR strategies such as 'peak shaving', 'load islanding for critical loads', and 'grid energy balancing'. In this project intelligent load islanding was demonstrated. Load islanding is supported in the architecture under the direction of smart battery management system and user preferences. The BMS safeguards the battery system hardware and the user preferences allow DR curtailment to be executed as per the user requirements and needs.

It was observed that the system will have to be properly sized and tested for supporting DR strategies mentioned above to realize enhanced benefits by facilitating the user to participate in pricing, net metering, and ancillary programs. The demand response controller with smart OpenADR client that was developed in this project will be extended to support and evaluate these DR strategies.

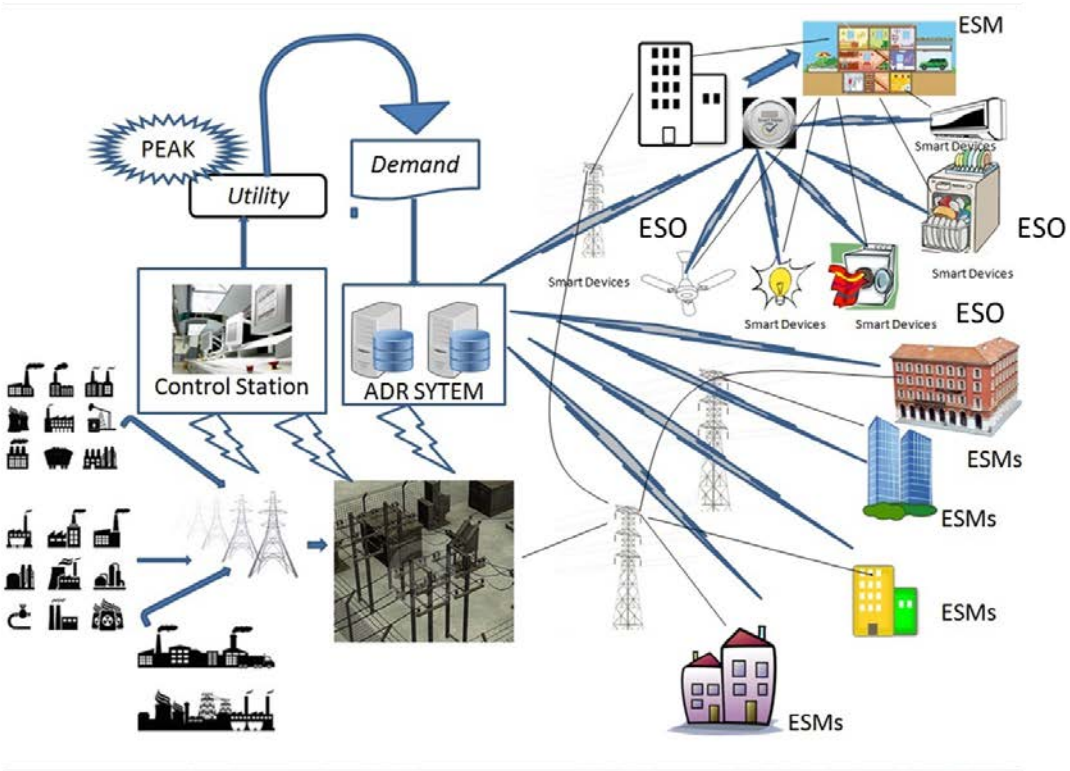
CHAPTER 6:
Residential Demand Response Modeling and Simulation

The DR simulation model facilitates generating an energy use model of a residential unit, through aggregation of individual electrical loads strictly satisfying the limitations and constraints of each of the electrical load. The model would take into consideration fixed number of electric appliances, gadgets, and plug-loads of designated electrical ratings in a residential unit.

6.1 Electric Service Model

The simulation model is represented by a proposed XML data model called 'Electric Service Model' (ESM). The hierarchy of the model would include the service size, circuits, and subcircuits within the breaker panel and the individual breakers, and is called the Electrical Service Objects (ESO). The relationship of ESMs and ESOs and their representation in the smart grid ecosystem is shown in Figure 22.

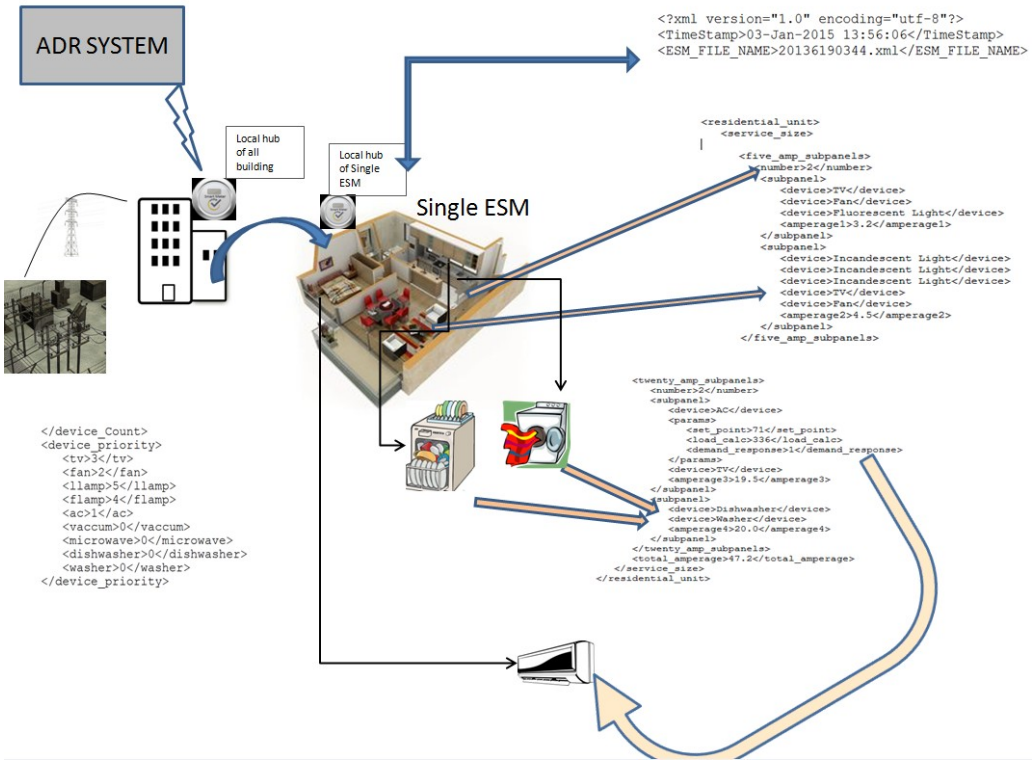
Figure 22: Relationship of ESMs and ESOs in Smart Grid



The motive to develop this model is to investigate a comprehensive extensible representation of smart grid components in a text-based language for describing the characteristics of the

components – specifications, configurations, status, and diagnostics details. The same model is used to represent end-use preferences of electrical loads and the load's participation in a DR event in this project. The ESM hierarchy with the physical manifestation of the service units is shown in Figure 23 and a sample XML dictionary of the ESM is presented in Appendix F.

Figure 23: Electrical Service Model Hierarchy and XML Data Model



6.2 Simulation with three ESMs

The simulator has three engines namely Randomizer, Priority Randomizer, Priority Timestamp Preference Engine. Once the demand and wattage for the service is available and the user's selection of operation either one of the engines are executed. Figure 24 shows the schematic of the simulator.

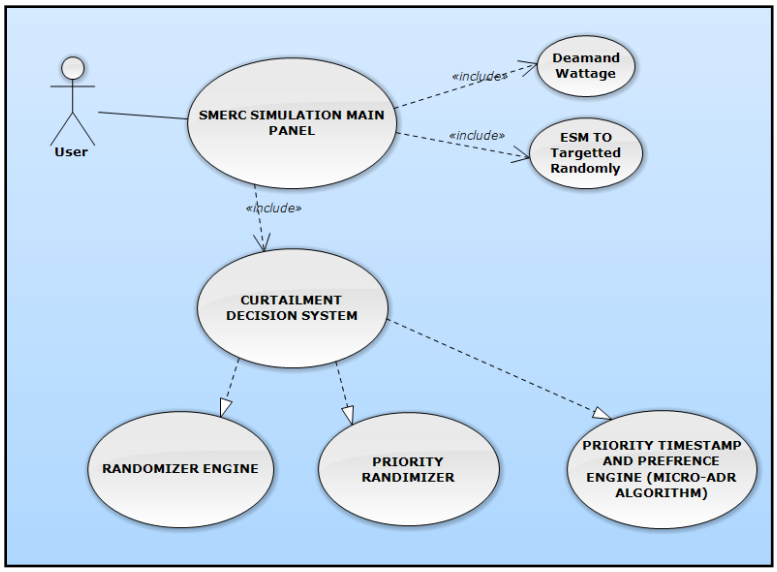
Salient points of the three simulator engines are:

Randomizer: It is a basic version of the randomized AutoDR. This AutoDR curtailment engine randomly chooses a participating ESM in the target service area (all participating ESMs) to send the demand signal.

Priority Randomizer: The AutoDR randomizer engine with priority accepts the priority of the devices from the user and stores it with the device model. The curtailment algorithm randomly curtails the individual devices in compliance of the set priorities.

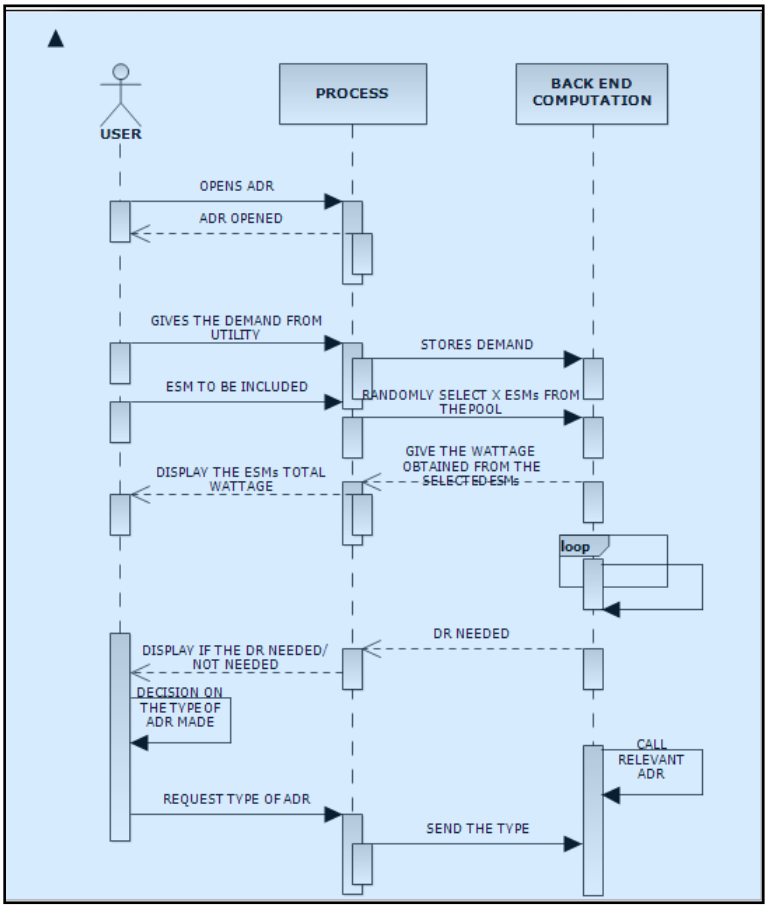
Priority Timestamp Preference Engine: The Priority Timestamp Preference randomizer overcomes the shortcoming of the Priority engine by taking into consideration users comfort and end-use time preference for a particular device. The algorithm takes into account the curtailment time and comfort preferences which minimize the discomfort which trickles more user-friendliness in the developed AutoDR system. The priority of the system remains the baseline for the curtailment. If the curtailment target is not met even after considering the users preference and comfortness, the user curtails the device based on the priority.

Figure 24: Simulator with the Processing Engines



The following activity diagram in Figure 25, demonstrates the execution of the randomizer engine

Figure 25: Activity Diagram for Randomizer Engine



The key steps of the simulation are:

1) AutoDR Requests for Demand and ESM count to proceed with the Process.
2) Once the Demand is given, it is sorted and the ESM count number of ESMs is pulled out from XML Database.
3) These XMLs are passed on to the preferred Engine.
4) The Activity is then followed by the Engine of relevant type.

The complete simulation run for five ESMs is shown in Appendix G.

6.3 Observations

The DR modeler and simulator have been successful in modeling residential electrical service and experimenting with DR load curtailment programs. The proposed human readable, XML-

based Electric Service Model is capable of representing electrical service, electrical loads, and user priorities and preferences concisely and is successful in supporting the workflow of residential DR. Though it was tested for a campus environment, it appears extensible and scalable to larger communities and utility-scale service base. The load selection engines in response to DR events also provided satisfactory results. These engines need to be tested for larger service bases to ascertain their effectiveness. The engines were successful in selecting optimum loads for a given DR event meeting the user priorities and end-use preferences for the campus sample. The modeler and the selection engines further need to be evaluated in their compatibility for supporting other DR load shed, shape and shift strategies, and price-based programs that are cogent residential DR programs.

CHAPTER 7:
Project Accomplishments and Conclusion

This report describes the design, implementation and testing of an Automated Demand Response architecture. The research included two main tasks. The first task was to design and develop an AutoDR test bed. The second task was to implement the test bed on UCLA campus and conduct field tests.

AutoDR technology and software platform developed at SMERC, UCLA was used to investigate Demand Response on UCLA campus. The AutoDR research test bed consists of a sophisticated ductless HVAC system in graduate student housing, and a LiFePO4 battery storage rig in SMERC. These demand response resources were chosen to augment the UCLA DR test bed that already has grid-friendly smart appliances, smart lighting, and EV charging infrastructure, all technologies developed in SMERC under sponsored research from various agencies and organizations. A residential DR modeler and simulator was also developed in this project to assess if a formal data model for residential electric service with the electrical loads, and associated user priorities and preferences can be used to scope target curtailment service in response to a DR event.

The research goal was to develop an Automated Demand Response system that would be compliant with the OpenADR specification, and would permit investigation of different load curtailment DR strategies and programs. The accomplishments of the work done in this project can be summarized under four distinct technology components.

7.1 Automatic Demand Response System

The REST architecture-based AutoDR system has been successfully investigated for both residential and commercial DR programs. A part of this system is the development of the WINSmartGrid OpenADR client platforms using the Reconfigurable Wireless Integrated Network Sensors (ReWINS) technology developed in SMERC. The clients are of two types: Smart Client and Bridge Client. Smart client, which is a platform that can process OpenADR events and directly manage the electrical load, and a bridge client, which is an intermediate node in the network path to receive and process OpenADR events and translate them into a downline control node host protocol or confer smart control to electrical loads.

The clients performed as desired, receiving the OpenADR events and converting them to load curtailment actions directly in the case of storage battery rig and indirectly through translation to BACnet in the case of the HVAC system. These clients are designed for evolution and can be extended to support other electrical loads, control protocols, and DR strategies.

The AutoDR system with these clients was not tested with a large number of nodes in the network as part of research investigation in this project. This would present an interesting environment to prove a network of large number of heterogeneous nodes with variety of control schemes, communication modalities, and DR strategies and programs. This work will be continued utilizing the UCLA DR test bed.

7.2 Ductless HVAC System as Demand Resource

The CITY MULTI ductless HVAC system from Mitsubishi was selected as a demand response resource as its control system provides the opportunity to investigate multiple strategies for demand response. The ductless, VRF HVAC system being latest generation and very economical to operate has the potential of proliferating on campus. The DR experiments conducted as part of this project can assist campus facilities to consider these HVAC systems as part of a demand response portfolio.

The bridge OpenADR client successfully translated DR events into BACnet signals compatible with the proprietary BACnet adapter. BACnet signals for curtailment strategies covering control of zones, indoor units, compressor frequency, and disciplined switching of condenser units were faithfully generated by the client in response to DR event.

Actual curtailment of the HVAC components in response to the BACnet signals was not attained. Curtailment was not achieved due to security restrictions imposed by the facilities network domain that hosts the BACnet adapter with campus domain hosting in the current service topology. When scaling up such solutions, facilities departments in campus settings represent a challenge when it comes to security restrictions.

7.3 Storage Battery as Demand Resource

The LiFePO4 battery storage system provides storage solution for a single building independently from the status of the grid. As a demand response resource, it powers an islanded AC load in the AutoDR system. Participation of the battery system in a DR event is guided by a preference settings system that allows the user to select charging the batteries later on in the evening or providing a separate AC source during the afternoons. The research team concluded that battery systems are easy to make customer friendly when it comes to AutoDR.

The smart OpenADR client embedded in the battery controller successfully parsed DR events and executed the control schemes based on the preferences such as time of day, amount of battery charge or amount of battery discharge, and target critical load. The battery management system safeguards the battery system components against any breach of safe operating thresholds. During multiple experiments, it was observed that the BMS successfully interrupted any active DR strategy when threshold values for cell temperature, lower / upper bound cell voltage, and charge / discharge values of the battery pack are breached. The research team concluded that a battery system would be a component of grid-friendly homes with on-site renewable generation. Based on the speed of the response, a battery system can respond to rapid needs of load curtailment.

The small size of the battery used in this research precluded testing complex DR strategies such as peak shaving, load islanding for critical loads, and grid energy balancing. Those applications can be scaled up from the tested curtailment levels in this project; therefore, the system design employed in this work should be extendible to larger battery systems enabling participation in incentivized load curtailment programs.

7.4 Residential Demand Response Modeling and Simulation

Availability of a concrete curtailment capacity size for DR dispatch is a coveted commodity. However, it is very difficult if real-time visibility into the potential participating electrical loads is not available, which often is the case as penetration of two-way communicating smart electrical loads is limited. When smart loads will become prevalent, DR dispatch can become precise with the availability of real-time data. The DR modeler and simulator platform developed as part of this research is targeted as a proof-of-concept towards such a scenario. With the availability of real-time data from electrical loads, their election of end-use duration and priority, it would be feasible to better assess the potential curtailment capacity in response to a DR event.

To accomplish this, a data model to represent residential units and their electrical loads called Electric Service Model is proposed. The model also facilitates representation of the end-use preference and priority of electric loads. In response to a DR event, the simulator is capable of aggregating electrical loads across multiple residences satisfying the priorities and preferences as curtailment capacity. A conclusion is that selection engines are adept at aggregating loads during DR event runtime by analyzing the prioritized and preferred available loads. This simulator feature can be employed during the active period of an OpenADR event to investigate accuracy of aggregate curtailment capacity to the DR dispatch.

As mentioned earlier, these engines need to be evaluated for large service bases to ascertain their effectiveness, and their compatibility for supporting other DR load shed, shape and shift strategies, and price-based programs.

7.5 Technology Transfer Activities

The objective of the tech transfer activities is to advance the research and technologies developed and explore the value and potential for use in real-world commercial settings. To further the commercialization potential, the following activities were performed:

- Industry workshops: Periodic workshops were held in which industry and utilities were invited. At these workshops, UCLA's researchers and graduate students showed demonstrations of research, and technology developed to the external visitors. Feedback was received in terms of trends in electric utility sector and such feedback was used in advancing the research.
- Industry meetings: Industry meetings were held one-on-one with individual companies that presented potential synergies with respect to determining how to integrate this project's research ideas with specific vendor technologies to create useful and advanced technologies and applications.
- Industry partners program: The research team held focused workshops with industry partners in which over a dozen vendors participated. These vendors demonstrated how their technologies are used in utility applications to the research team. The vendors also provided an overview about their commercialization process from the vendor perspective. UCLA's commercialization officers attended the workshops and provided

insights on UCLA's internal commercialization process. An objective of such focused meetings was to determine how to do technology transfer and to create transfer-ready technologies, and to understand both perspectives – the internal UCLA perspective and external vendor perspective.

- Live demonstrations on campus: The research team has taken some of the research and technology and created live demonstrations on campus. For example, the smart appliances controllers that accept the OpenADR signals are available for demonstration on campus. In the future, such live demonstrations are expected to lead to possible opportunities for technology transfer.

Because of this research project, the research team created AutoDR technologies that are relevant to the commercial world.

GLOSSARY

Term	Definition
Automated Demand Response (AutoDR)	Automatic demand response is the minimal to no human involvement in achieving electricity curtailment in response to utility-side signals
Home Area Network (HAN)	Dedicated network that connects smart electrical loads such as devices and appliancesto the Smart Grid and allows them to be automated to interact with utility/ISO control signals
JavaScript Object Notation (JSON)	Lightweight text-based data-interchange format that is easy for people to read and write and easy for machines to generate or parse
Open Automated Demand Response (OpenADR)	A messaging protocol for exchanging price and grid reliability between utilities/ISO and consumers.
Extensible Markup Language (XML)	A standard to describe data that allows a flexible way to create information formats and electronically share structured data over the networks
AC	Alternating Current
Ah	Ampere-Hour
APDU	Application Protocol Data Units
BAS	Building Automation System
BMS	Battery Management System
COTS	Commercial-off-the-shelf
DHP	Ductless Heat Pump
DR	Demand Response
DRAS	Demand Response Automation Server
ESM	Electric Service Model
ESM	Electrical Service Model

ESO	Electrical Service Object
EV	Electric Vehicle
HTTP	Hypertext Transfer Protocols
HVAC	Heating, Ventilation, and Air Conditioning
IP	Internet Protocol
kWh	Kilowatt-Hour
NTP	Network Time Protocol
REST	Representational State Transfer
ReWINS	Reconfigurable Wireless Interface for Networking Sensors
SBC	Single Board Computer
SMERC	Smart Grid Energy Research Center
SSH	Secure Shell
TCP	Transmission Control Protocol
UART	Universal Asynchronous Receiver / Transmitter
UDP	User Datagram Protocol
UPS	Uninterruptible Power Supply
URI	Uniform Resource Identifier
VRF	Variable Refrigerant Flow
WCF	Windows Communications Foundation

REFERENCES

Epstein, Gary, Mark D'Antonio, Chris Schmidt, John Seryak, and Chris Smith. "Demand Response Enabling Technologies and Approaches for Industrial Facilities", in *Proc. 27th Industrial Energy Technology Conf.*, New Orleans, May 2005.

Ghatikar, Girish, and Beinery Rolf. "Smart Grid Standards and System Interoperability: A Precedent with OpenADR", in *Proc. of Grid-Interop 2011*, Phoenix, AZ, Dec. 2011.

Jarred, Michael W. "Delivering on the promise of California's Demand Response Programs". *Policy Matters, California Senate Office of Research*, Jun 2014.

Laine, Markku. "RESTful Web Services for the Internet of Things." Technical Report, Dept. of Media Technology, Aalto University, Finland, 2011.

Mitchell, Scott, Rafik. Sarhadian, Sean Guow, Bruce Coburn, John Lutton, Ishtiaq Chisti, Devin Rauss, and Carlos Haiad. "Residential Appliance Demand Response Testing", in *2014 ACEEE Summer Study on Energy Efficiency Buildings*, Pacific Grove, CA, Aug 2014

O'Connell, Niamh, Pierre Pinson, Henrik Madsen, and Mark O'Malley. "Benefits and challenges of electrical demand response: A critical review." *Renewable and Sustainable Energy Reviews*, vol. 39, pp. 686-699, Nov. 2014.

Piette, Mary Anne, Girish Ghatikar, Sila Kiliccote, Ed Koch, Dan Hennage, Peter Palensky, and Charles McParland. "Open Automated Demand Response Communications Specification (Version 1.0)." California Energy Commission, PIER Program, CEC-500-2009-063.

Ramamurthy, Harish, Shiv Prabhu, Rajit Gadh, and Asad Madni. "Wireless Industrial Monitoring and Control Using a Smart Sensor Platform." *J. IEEE Sensors*, vol. 7, no. 5, pp 611-618, May 2007.

Ramamurthy, Harish, Dhananjay Lal, Shiv Prabhu, and Rajit. Gadh. "ReWINS: A Distributed Multi-RF Sensor Control Network for Industrial Automation", in *Proc. Wireless Telecommunications Symposium (WTS 2005)*, Cal Poly Pomona, Pomona, CA, Apr. 2005.

Shelby, Zach. "Embedded Web Services." *J. IEEE Wireless Communications*, vol. 17, no. 6, pp 52-57, Dec. 2010.

Smith, Kelly. "Scaling Demand Response through Interoperability in Commercial Buildings", in *4th Annual Grid-Interop Forum*, Chicago, Dec 2010.

Spam, Bethany, Xin Jin, and Lieko. Earle. "Laboratory testing of Demand Response Enabled Household Appliances." Technical Report, NREL/TP-5500-60383, National Renewable Energy Laboratory, Oct. 2013.

APPENDIX A:
Sample DR Event

These are sample DR events generated by the DRAS server in XML or JSON. They show the different states of the event.

a) DR Event in XML

Notification:


```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<p:listOfEventState xmlns:p="http://www.openadr.org/DRAS/EventState">
 <p:eventStates programName="LADWP Demo [03.08.2015]" eventModNumber="0"
eventIdentifier="Client10_Test_345678_906745" drasClientID="dras_client10" eventStateID="1034654074"
schemaVersion="1.0" drasName="UCLA-DRAS 2.5.5" testEvent="false" offLine="false">
 <p:simpleDRModeData>
 <p:EventStatus>NONE</p:EventStatus>
 <p:OperationModeValue>NORMAL</p:OperationModeValue>
 <p:currentTime>0.0</p:currentTime>
 </p:simpleDRModeData>
 </p:eventStates>
</p:listOfEventState>
```


Far state:


```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<p:listOfEventState xmlns:p="http://www.openadr.org/DRAS/EventState"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://openadr.lbl.gov/src/1/EventState.xsd">
 <p:eventStates programName="LADWP Demo [03.08.2015]" eventModNumber="0"
eventIdentifier="Client10_Test_345678_906745" drasClientID="dras_client10" eventStateID="1034654074"
schemaVersion="1.0" drasName="UCLA-DRAS 2.5.5" testEvent="false" offLine="false">
 <p:simpleDRModeData>
 <p:EventStatus>FAR</p:EventStatus>
 <p:OperationModeValue>NORMAL</p:OperationModeValue>
 <p:currentTime>-60</p:currentTime>
 <p:operationModeSchedule>
 <p:modeSlot>
 <p:OperationModeValue>MODERATE</p:OperationModeValue>
 <p:modeTimeSlot>0</p:modeTimeSlot>
 </p:modeSlot>
 </p:operationModeSchedule>
 </p:simpleDRModeData>
 <p:drEventData>
 <p:notificationTime>2015-03-08T22:15:24.24-07:00</p:notificationTime>
 <p:startTime>2015-03-08T22:16:24.24-07:00</p:startTime>
 <p:endTime>2015-03-08T22:18:24.24-07:00</p:endTime>
 <p:eventInfoInstances>
 <p:eventInfoTypeID>LOAD_AMOUNT</p:eventInfoTypeID>
 <p:eventInfoName>bid</p:eventInfoName>
 <p:eventInfoValues>
 <p:value>0.0</p:value>
 <p:timeOffset>0</p:timeOffset>
 </p:eventInfoValues>
 <p:eventInfoValues>
 <p:value>0.0</p:value>
 <p:timeOffset>120</p:timeOffset>
 </p:eventInfoValues>
 </p:eventInfoInstances>
 <p:eventInfoInstances>
 <p:eventInfoTypeID>PRICE_ABSOLUTE</p:eventInfoTypeID>
 <p:eventInfoName>price</p:eventInfoName>
 <p:eventInfoValues>
```

```
<p:value>0.0</p:value>
<p:timeOffset>0</p:timeOffset>
</p:eventInfoValues>
<p:eventInfoValues>
<p:value>0.0</p:value>
<p:timeOffset>120</p:timeOffset>
</p:eventInfoValues>
</p:eventInfoInstances>
<p:eventInfoInstances>
<p:eventInfoTypeID>PRICE_MULTIPLE</p:eventInfoTypeID>
<p:eventInfoName>cpp_price</p:eventInfoName>
<p:eventInfoValues>
<p:value>0.0</p:value>
<p:timeOffset>0</p:timeOffset>
</p:eventInfoValues>
<p:eventInfoValues>
<p:value>0.0</p:value>
<p:timeOffset>120</p:timeOffset>
</p:eventInfoValues>
</p:eventInfoInstances>
</p:drEventData>
</p:eventStates>
</p:listOfEventState>
```


Active State:


```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<p:listOfEventState xmlns:p="http://www.openadr.org/DRAS/EventState"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://openadr.lbl.gov/src/1/EventState.xsd">
 <p:eventStates programName="LADWP Demo [03.08.2015]" eventModNumber="0"
eventIdentifier="Client10_Test_345678_906745" drasClientID="dras_client10" eventStateID="1034654074"
schemaVersion="1.0" drasName="UCLA-DRAS 2.5.5" testEvent="false" offLine="false">
 <p:simpleDRModeData>
 <p:EventStatus>ACTIVE</p:EventStatus>
 <p:OperationModeValue>MODERATE</p:OperationModeValue>
 <p:currentTime>15</p:currentTime>
 <p:operationModeSchedule>
 <p:modeSlot>
 <p:OperationModeValue>MODERATE</p:OperationModeValue>
 <p:modeTimeSlot>0</p:modeTimeSlot>
 </p:modeSlot>
 </p:operationModeSchedule>
 </p:simpleDRModeData>
 <p:drEventData>
 <p:notificationTime>2015-03-08T22:15:24.24-07:00</p:notificationTime>
 <p:startTime>2015-03-08T22:16:24.24-07:00</p:startTime>
 <p:endTime>2015-03-08T22:18:24.24-07:00</p:endTime>
 <p:eventInfoInstances>
 <p:eventInfoTypeID>LOAD_AMOUNT</p:eventInfoTypeID>
 <p:eventInfoName>bid</p:eventInfoName>
 <p:eventInfoValues>
 <p:value>0.0</p:value>
 <p:timeOffset>0</p:timeOffset>
 </p:eventInfoValues>
 <p:eventInfoValues>
 <p:value>0.0</p:value>
 <p:timeOffset>120</p:timeOffset>
 </p:eventInfoValues>
 </p:eventInfoInstances>
 <p:eventInfoInstances>
 <p:eventInfoTypeID>PRICE_ABSOLUTE</p:eventInfoTypeID>
 <p:eventInfoName>price</p:eventInfoName>
 <p:eventInfoValues>
 <p:value>0.0</p:value>
 <p:timeOffset>0</p:timeOffset>
 </p:eventInfoValues>
 <p:eventInfoValues>
 <p:value>0.0</p:value>
```

```
<p:timeOffset>120</p:timeOffset>
</p:eventInfoValues>
</p:eventInfoInstances>
<p:eventInfoInstances>
<p:eventInfoTypeID>PRICE_MULTIPLE</p:eventInfoTypeID>
<p:eventInfoName>cpp_price</p:eventInfoName>
<p:eventInfoValues>
<p:value>0.0</p:value>
<p:timeOffset>0</p:timeOffset>
</p:eventInfoValues>
<p:eventInfoValues>
<p:value>0.0</p:value>
<p:timeOffset>120</p:timeOffset>
</p:eventInfoValues>
</p:eventInfoInstances>
</p:drEventData>
</p:eventStates>
</p:listOfEventState>
```


b) DR Event in JSON

Notification:


```
{
"p:listOfEventState": {
"@xmlns:p": "http://www.openadr.org/DRAS/EventState",
"p:eventStates": {
"@programName": "LADWP Demo [03.08.2015]",
"@eventModNumber": "0",
"@eventIdentifier": "Client10_Test_345678_906745",
"@drasClientID": "dras_client",
"@eventStateID": "7112038466",
"@schemaVersion": "1.0",
"@drasName": "UCLA-DRAS 2.1.8",
"@testEvent": "false",
"@offLine": "false",
"p:simpleDRModeData": {
"p:EventStatus": "NONE",
"p:OperationModeValue": "NORMAL",
"p:currentTime": "0.0"
}
}
}
}
```


Near State:


```
{
"p:listOfEventState": {
"@xmlns:p": "http://www.openadr.org/DRAS/EventState",
"p:eventStates": {
"@programName": "LADWP Demo [03.08.2015]",
"@eventModNumber": "0",
"@eventIdentifier": "Client10_Test_345678_906745",
"@drasClientID": "dras_client",
"@eventStateID": "7112038466",
"@schemaVersion": "1.0",
"@drasName": "UCLA-DRAS 2.1.8",
"@testEvent": "false",
"@offLine": "false",
"p:simpleDRModeData": {
"p:EventStatus": "NEAR",
"p:OperationModeValue": "NORMAL",
"p:currentTime": "-10",
"p:operationModeSchedule": {
"p:modeSlot": {
```

"p:OperationModeValue": "HIGH",
"p:modeTimeSlot": "0"
}
}
},
"p:drEventData": {
"p:notificationTime": "2015-03-08T21:42:00-07:00",
"p:startTime": "2015-03-08T21:42:30-07:00",
"p:endTime": "2015-03-08T21:44:30-07:00",
"p:eventInfoInstances": [
{
"p:eventInfoTypeID": "LOAD_AMOUNT",
"p:eventInfoName": "bid",
"p:eventInfoValues": [
{
"p:value": "0.0",
"p:timeOffset": "0"
},
{
"p:value": "0.0",
"p:timeOffset": "120"
}
]
},
{
"p:eventInfoTypeID": "PRICE_ABSOLUTE",
"p:eventInfoName": "price",
"p:eventInfoValues": [
{
"p:value": "0.0",
"p:timeOffset": "0"
},
{
"p:value": "0.0",
"p:timeOffset": "120"
}
]
},
{
"p:eventInfoTypeID": "PRICE_MULTIPLE",
"p:eventInfoName": "cpp_price",
"p:eventInfoValues": [
{
"p:value": "0.0",
"p:timeOffset": "0"
},
{
"p:value": "0.0",
"p:timeOffset": "120"
}
]
}
]
}
}
}
}

Active State:

{
"p:listOfEventState": {
"@xmlns:p": "http://www.openadr.org/DRAS/EventState",
"p:eventStates": {
"@programName": "LADWP Demo [03.08.2015]",
"@eventModNumber": "0",
"@eventIdentifier": "Client10_Test_345678_906745",
"@drasClientID": "dras_client",

"@eventStateID": "7112038466",
"@schemaVersion": "1.0",
"@drasName": "UCLA-DRAS 2.1.8",
"@testEvent": "false",
"@offLine": "false",
"p:simpleDRModeData": {
"p:EventStatus": "ACTIVE",
"p:OperationModeValue": "HIGH",
"p:currentTime": "8",
"p:operationModeSchedule": {
"p:modeSlot": {
"p:OperationModeValue": "HIGH",
"p:modeTimeSlot": "0"
}
}
},
"p:drEventData": {
"p:notificationTime": "2015-03-08T21:42:00-07:00",
"p:startTime": "2015-03-08T21:42:30-07:00",
"p:endTime": "2015-03-08T21:44:30-07:00",
"p:eventInfoInstances": [
{
"p:eventInfoTypeID": "LOAD_AMOUNT",
"p:eventInfoName": "bid",
"p:eventInfoValues": [
{
"p:value": "0.0",
"p:timeOffset": "0"
},
{
"p:value": "0.0",
"p:timeOffset": "120"
}
]
},
{
"p:eventInfoTypeID": "PRICE_ABSOLUTE",
"p:eventInfoName": "price",
"p:eventInfoValues": [
{
"p:value": "0.0",
"p:timeOffset": "0"
},
{
"p:value": "0.0",
"p:timeOffset": "120"
}
]
},
{
"p:eventInfoTypeID": "PRICE_MULTIPLE",
"p:eventInfoName": "cpp_price",
"p:eventInfoValues": [
{
"p:value": "0.0",
"p:timeOffset": "0"
},
{
"p:value": "0.0",
"p:timeOffset": "120"
}
]
}
]
}
}
}
}

APPENDIX B:
Mitsubishi CITY-MULTI HVAC System Specification

Specifications of the HVAC system installed at Hilgard Av., graduate apartments

720 Hilgard Av. / 54 Apartments				
Two Systems				
14-ton heat recovery system				
Specifications		System	Module 1	Module 2
Unit Type		PURY-P168TSHMU-A	PURY-P96THMU-A 8-Ton	PURY-P72THMU-A 6-Ton
Nom. Cooling Cap.	Btu/h	168,000	96,000	72,000
Nom.Heating Cap.	Btu/h	188,000	108,000	80,000
Electrical Power Requirements			208/230V, 3-PH, 60 Hz	
Cooling Power Input	kW	14.19	8.35	5.55
Heating Power Input	kW	15.14	8.66	6.04
Cooling Current (208/230 V)	A	43.7 / 39.5	25.7 / 23.2	17.1 / 15.4
Heating Current (208/230 V)	A	46.6 / 42.2	26.7 / 24.1	18.6 / 16.8
16-ton heat recovery system				
Specifications		System	Module 1	Module 2
Unit Type		PURY-P192TSHMU-A	PURY-P96THMU-A 8-Ton	PURY-P96THMU-A 8-Ton
Nom. Cooling Cap.	Btu/h	192,000	96,000	96,000
Nom.Heating Cap.	Btu/h	200,00	108,000	108,000
Electrical Power Requirements			208/230V, 3-PH, 60 Hz	
Cooling Power Input	kW	17.04	8.35	8.35
Heating Power Input	kW	15.85	8.66	8.66
Cooling Current (208/230 V)	A	52.5 / 47.5	25.7 / 23.2	25.7 / 23.2
Heating Current (208/230 V)	A	48.8 / 44.2	26.7 / 24.1	26.7 / 24.1

824 Hilgard Av. / 29 Apartments				
One System				
12-ton heat recovery system				
Specifications		System	Module 1	Module 2
Unit Type		PURY-P144YSHMU-A	PURY-P72YHMU-A 6-Ton	PURY-P72YHMU-A 6-Ton
Nom. Cooling Cap.	Btu/h	144,000	72,000	72,000
Nom.Heating Cap.	Btu/h	160,000	80,000	80,000
Electrical Power Requirements			460V, 3-ph, 60 Hz	
Cooling Power Input	kW	11.32	5.55	5.55
Heating Power Input	kW	12.44	6.04	6.04
Cooling Current (208/230 V)	A	15.7	7.7	7.7
Heating Current (208/230 V)	A	17.3	8.4	8.4

APPENDIX C:
BACnet Commands for HVAC Load Curtailment Tests

The following tables show sample BACnet commands generated by the bridge OpenADR client for each of the proposed tests in response to a DR event.

Test 1: Reducing compressor frequency (speed)			
Operate the compressor at reduced capacity to maintain minimally effective environmental conditions			
Step	Description		Result (Success/Failure)
1.	*Frame the command packet* Control item: Fan Speed Control Type: Multistate Input Instance Number: 08		
2.	*Setup property values* Possible values: High, Mid2, Mid1, Low		
3.	Application Protocol Data Units (APDU)		Individual test packet frames developed
	HEX value	*Description*	
	00	PDU Type=0 (BACnet-Confirmed-Request-PDU, SEG=0, MOR=0, SA=0)	
	04	Maximum APDU Size Accepted=1024 octets	
	59	Invoke ID=89	
	0F	Service Choice=15 (WriteProperty-Request)	
	0C	SD Context Tag 0 (Object Identifier, L=4)	
	00801000	Multiple Input, Instance Number=8	
	19	SD Context Tag 1 (Property Identifier, L=1)	
	01	Present Value [High]	
	3E	PD Opening Tag 3 (Property Value)	
	44	Application Tag 4 (Real, L=4)	
	03	New Value [Mid1]	
	3F	PD Closing Tag 3 (Property Value)	

Test 2: Zonal on/off sequencing			
Sequentially switch on/off zones to minimize discomfort by minimizing temperature changes			
Step	Description		Result (Success/Failure)
1.	*Frame the command packet* Control item: On Off Control Type: Binary Input Instance Number: 02		
2.	*Setup property values* Possible values: On, Off		
3.	Application Protocol Data Units (APDU)		Individual test packet frames developed
	HEX Value	*Description*	
	00	PDU Type=0 (BACnet-Confirmed-Request-PDU, SEG=0, MOR=0, SA=0)	
	04	Maximum APDU Size Accepted=1024 octets	
	59	Invoke ID=89	
	0F	Service Choice=15 (WriteProperty-Request)	
	0C	SD Context Tag 0 (Object Identifier, L=4)	
	00800001	Binary Output, Instance Number=2, Group Number to be provided [Zone]	
	19	SD Context Tag 1 (Property Identifier, L=1)	
	01/00	Present Value [On/Off]	
	3E	PD Opening Tag 3 (Property Value)	
	44	Application Tag 4 (Real, L=4)	
	00/01	New Value [Off/On]	
	3F	PD Closing Tag 3 (Property Value)	

Test 3: Turning off indoor units selectively			
Turn off one or more indoor units and allow conditioning (temperature/humidity) to drift from adjacent zones			
Step	Description		Result (Success/Failure)
1.	*Frame the command packet* Control item: On Off Control Type: Binary Input Instance Number: 02		
2.	*Setup property values* Possible values: On. Off		
3.	Application Protocol Data Units (APDU)		Individual test packet frames developed
	HEX Value	*Description*	
	00	PDU Type=0 (BACnet-Confirmed-Request-PDU, SEG=0, MOR=0, SA=0)	
	04	Maximum APDU Size Accepted=1024 octets	
	59	Invoke ID=89	
	0F	Service Choice=15 (WriteProperty-Request)	
	0C	SD Context Tag 0 (Object Identifier, L=4)	
	00800001	Binary Output, Instance Number=2	
	19	SD Context Tag 1 (Property Identifier, L=1)	
	01/00	Present Value [On/Off]	
	3E	PD Opening Tag 3 (Property Value)	
	44	Application Tag 4 (Real, L=4)	
	00/01	New Value [Off/On]	
	3F	PD Closing Tag 3 (Property Value)	

APPENDIX D:
Battery Specification

Lithium Iron Phosphate (LiFePO4) chemistry-based battery technology was selected for its high power and energy density, excellent safety performance, stability under high temperature, better life cycling, light weight, cost-of-ownership over lifetime, and better environment friendliness. The specification of the cells of the battery:

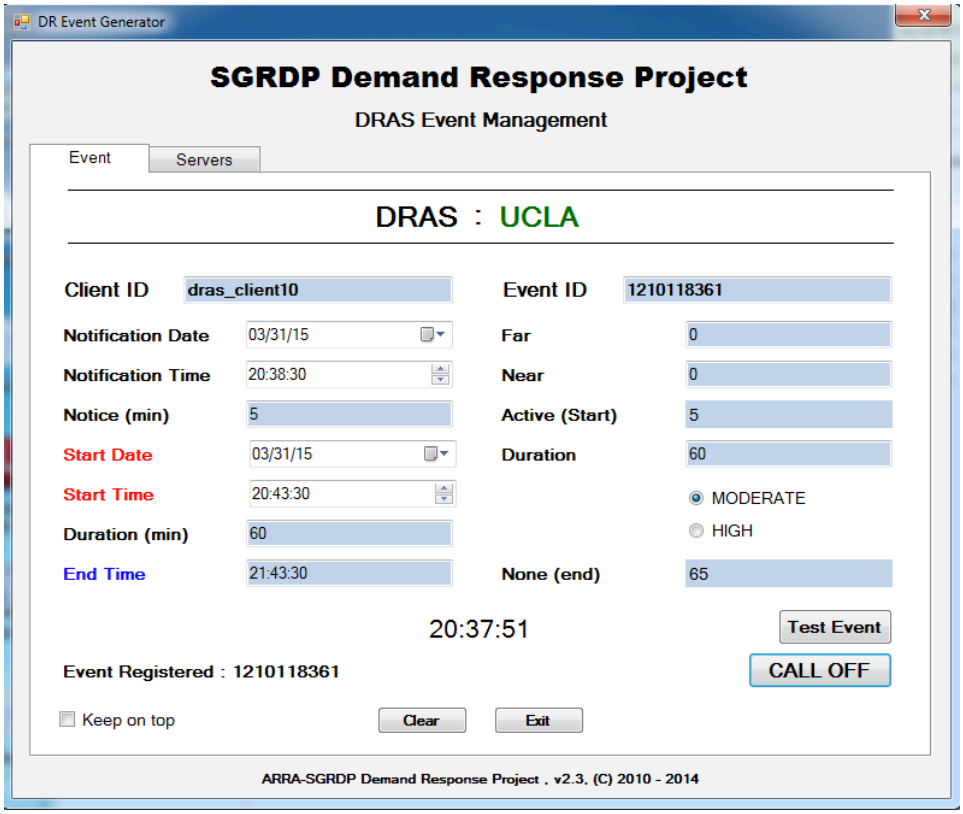
Battery Specification: LiFePO4 Prismatic Module: 3.2V, 40 Ah, Rate (128 wh)			
Nominal Capacity	40 Ah	**Working voltage**	Single cell charging : 3.8V
			Battery pack charging : 3.65V
			Single cell discharging: 2.5V
			Battery pack discharging: 2.8V
Max. Charging Current	≤3C	**Max. discharging Current**	Continuous current: ≤ 3C
			Impulse current: ≤10C
Std. Charging Current	03~0,8C	**Best charging current**	0.5C
Internal Resistance	≤2.5 mΩ	**Cycle life**	Single cell ≥ 2000 times (80% DOD)
			Battery pack ≥ 1500 times (80% DOD)
Self-discharge rate (month)	≤ 3%	**Cell weight**	1.4kg+-100g
Energy density	85 – 100 Wh/kg	**Power Density**	> 800 w/kg
Cell dimensions	126 mm x 46 mm x 180 mm		

Source: Specification of LiFePO4 Prismatic Module: 3.2V 40 Ah, 10C Rate (128 wh), AA portable Power Corp, Richmond CA 94804

APPENDIX E:
Backup Power for Critical Load during DR Event

The battery rig was setup to support critical load during a DR event. The DR event was scheduled as shown in Figure 26. It was a 'moderate' event for a period of 60 min. During the active period the battery controller switched supply to the designated critical load (6 lamp bank).

Figure 26: Demand Response Event Scheduling



However, for this test there was no specific preference set on charge level, or duration of supply to critical load. Under this setup the battery pack operated in default mode and is controlled by the battery management system (BMS) with built-in protections for cell voltage, cell temperature, battery capacity, and battery charge and discharge rates. As seen in Figure 27, the voltage of cell 21 dropped below 2.5V, which is the voltage threshold, within 16 min approximately and the battery controller opted out of the DR event and stopped supplying the critical load. The battery was switched to charging. Figure 28 shows the cell temperatures which is one of the parameters monitored by the BMS.

Figure 27: Cell Voltage and Amperes During the 60 Minute DR Event

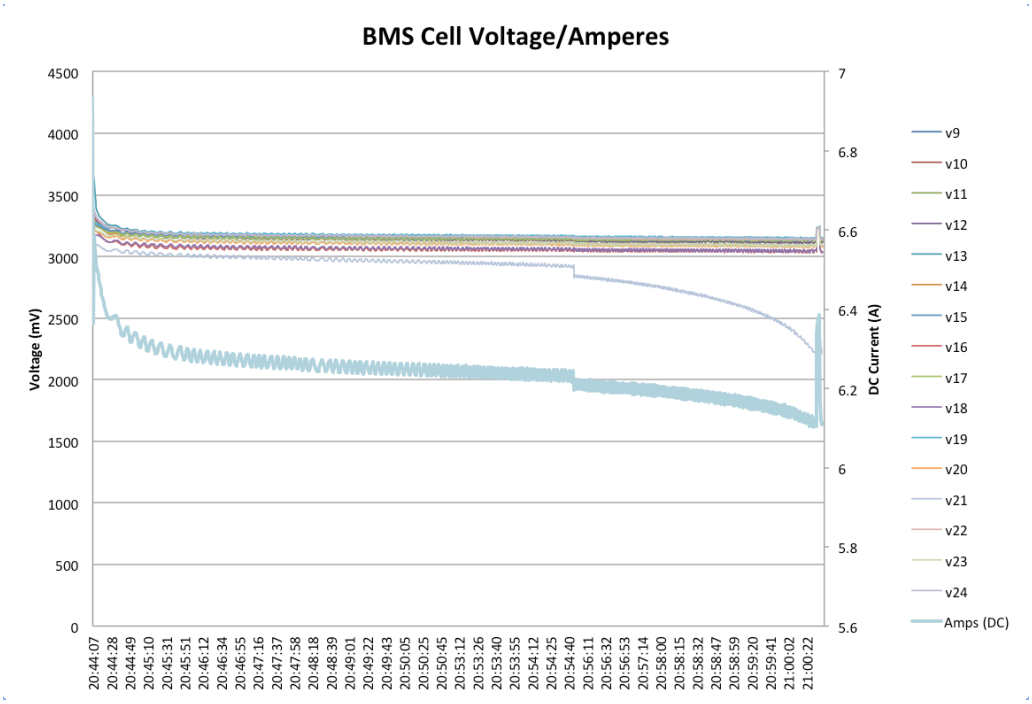
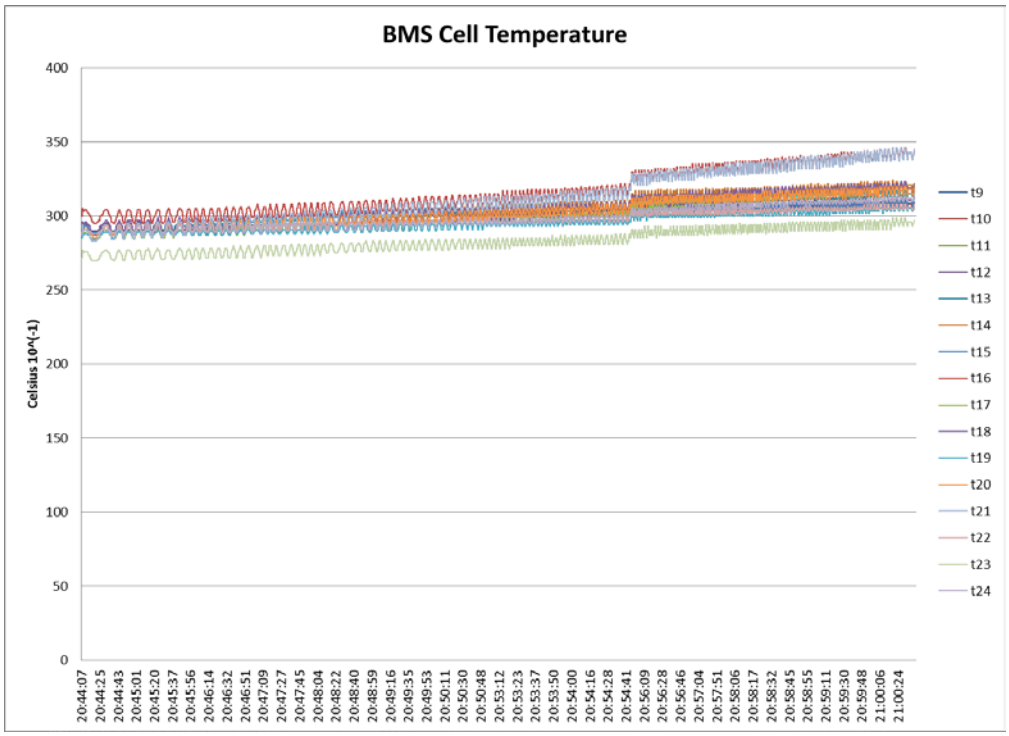


Figure 28: Cell Temperature During 60 Minute DR Event



APPENDIX F:
Electrical Service Model

An XML-based data model to define a residential unit in terms of hierarchical electrical services like the full service of the unit, the circuits, sub-circuits, appliances, gadgetsplug-loads, and so forth. Sample dictionary of the model is as shown in the following table.

Tag	Description
MODEL XML	
<RESIDENTIAL_UNIT> </RESIDENTIAL_UNIT>	Tag for ESM Model – All the ESM Information
<SERVICE_SIZE> </SERVICE_SIZE>	Info of service size: value, subpanels and total amperage
<VALUE> </VALUE>	Value of the service size.
<FIVE_AMP_SUBPANELS> </FIVE_AMP_SUBPANELS>	Details of Five ampere subpanels
<FIVE_AMP_SUBPANELS> <NUMBER> </NUMBER> </FIVE_AMP_SUBPANELS>	Number of Five ampere subpanels
<SUBPANEL> </SUBPANEL>	Subpanel information.
<DEVICE> </DEVICE>	Identity of the device in the subpanel: AC, TC, FL, Fridge.
<AMPERAGE> </AMPERAGE>	Amperage information
<TWENTY_AMP_SUBPANELS> </TWENTY_AMP_SUBPANELS>	Information for twenty ampere subpanel.
<TOTAL_AMPERAGE> </TOTAL_AMPERAGE>	Total amperage of all subpanels.
<PARAMS> </PARAMS>	Parameters of individual device.

Sample ESM of an apartment

```
<?xml version="1.0" encoding="utf-8"?>
<residential_unit>
 <service_size>
 <five_amp_subpanels>
 <number>2</number>
 <subpanel>
        <device>TV</device>
 <device>Fan</device>
 <device>Fluorescent Light</device>
 <amperage1>3.2</amperage1>
 </subpanel>
 <subpanel>
 <device>Incandescent Light</device>
 <device>Incandescent Light</device>
 <device>Incandescent Light</device>
 <device>TV</device>
 <device>Fan</device>
 <amperage2>4.5</amperage2>
 </subpanel>
 </five_amp_subpanels>
```

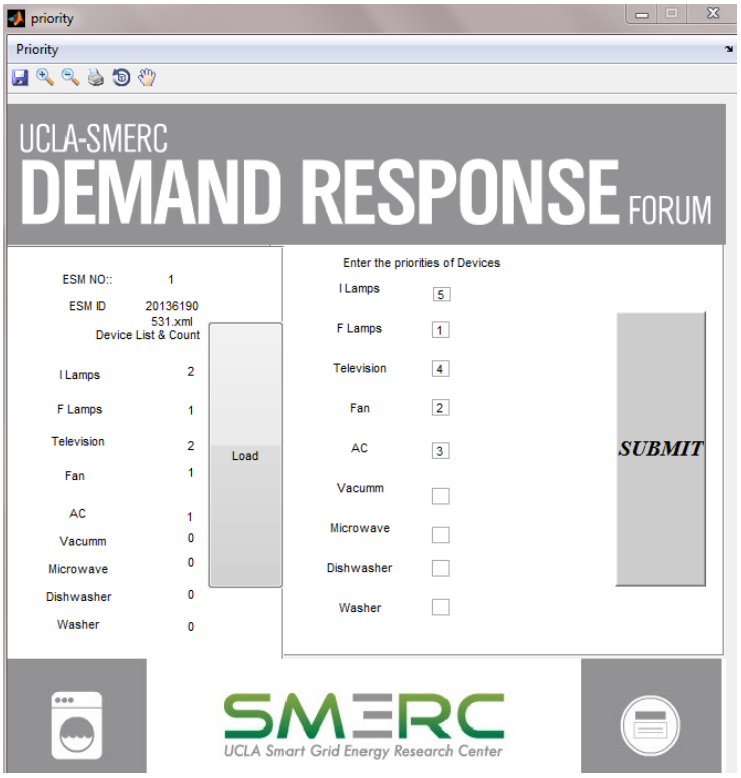
```
<Count>
<tv>3</tv>
<fan>2</fan>
<washer>1</washer>
<dishwasher>1</dishwasher>
<microwave>0</microwave>
<vacuum>0</vacuum>
<AC>1</AC>
<flamp>1</flamp>
<llamp>3</llamp>
</Count>
<Priority>
<tv>6</tv>
<fan>1</fan>
<washer>2</washer>
<dishwasher>7</dishwasher>
<microwave>0</microwave>
<vacuum>0</vacuum>
<AC>4</AC>
<flamp>5</flamp>
<llamp>3</llamp>
</Priority>
<twenty_amp_subpanels>
<number>2</number>
<subpanel>
<device>AC</device>
<params>
<set_point>71</set_point>
<load_calc>336</load_calc>
<demand_response>1</demand_response>
</params>
<device>TV</device>
<amperage3>19.5</amperage3>
</subpanel>
<subpanel>
<device>Dishwasher</device>
<device>Washer</device>
<amperage4>20.0</amperage4>
</subpanel>
</twenty_amp_subpanels>
<total_amperage>47.2</total_amperage>
</service_size>
</residential_unit>
```

APPENDIX G:
Residential Demand Response Simulation Results

Demand response simulation was run on a batch of five Electrical Services Models (ESM). For each of the individual ESM, priorites and preferences were established. The data and screen shots of each of the stages of the simulation run are depicted below. Samples are shown for only 2 models.

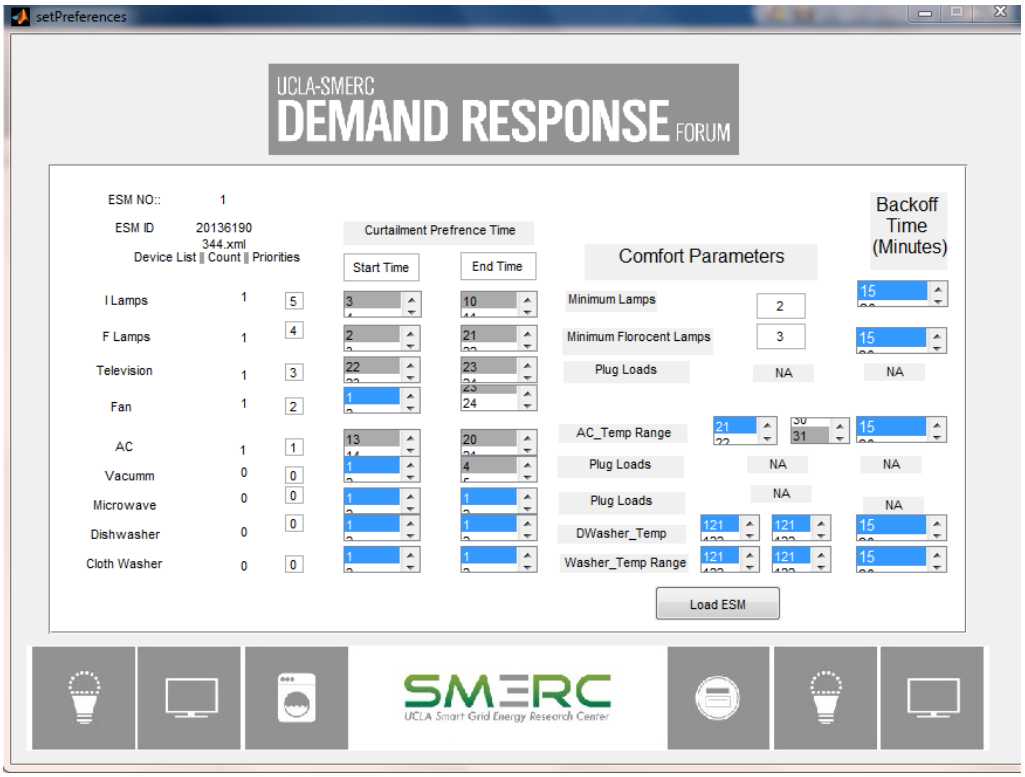
1. *Setting Priorities*
   This is the UI via which the ESM model is read and then the ESM model is updated with priorities provided by the user as shown.

   ESM ID: 20136190531

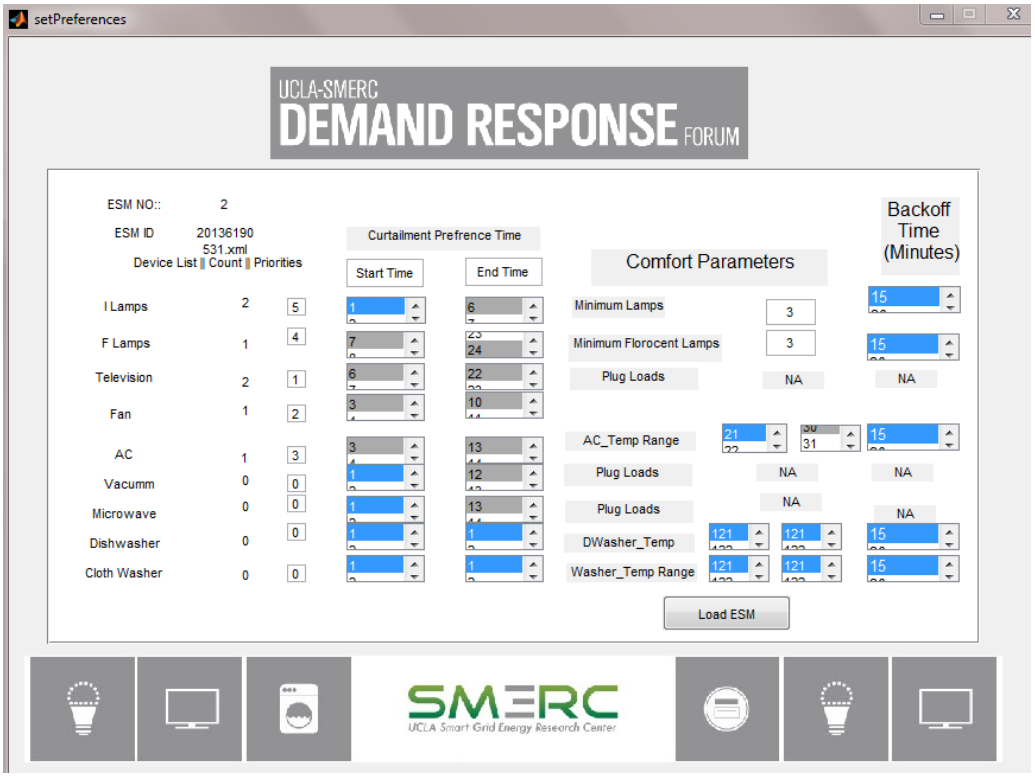


2. *Setting preferences*
   This is the UI via which the ESM model is read along with priorities and then the timestamp and comfort zone preferences are set.

   ESM ID: 20136190344



ESM ID: 20136190351



3. *ESM Models with priorites assigned to the loads*

This is the Priority-count XML model for the ESMs that were selected during the curtailment. These values are accumulated after the priorities are provided by the user

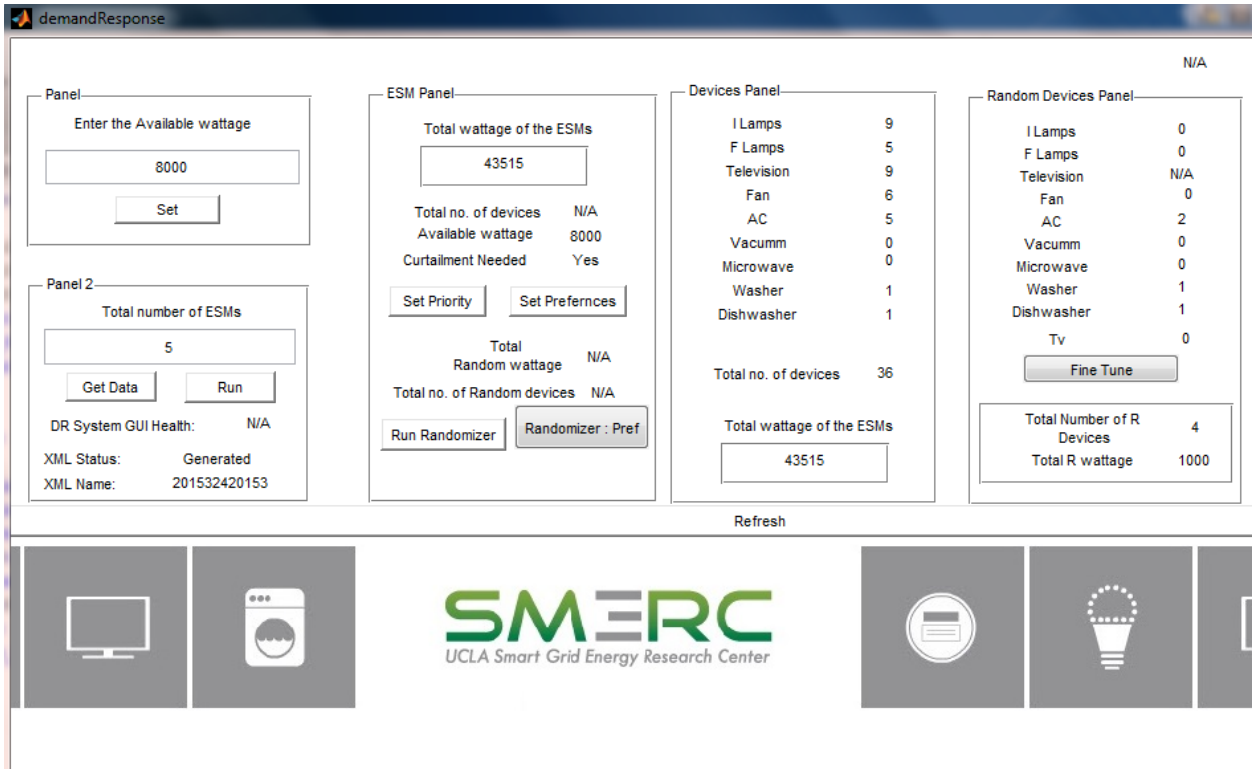
ESM ID: 20136190344


```xml
<?xml version="1.0" encoding="utf-8"?>
<ESM_SIMULATION>
   <TimeStamp>24-Mar-2015 19:54:26</TimeStamp>
   <ESM_FILE_NAME>20136190344.xml</ESM_FILE_NAME>
   <Total_Amperage>3865</Total_Amperage>
   <device_Count>
      <tv>1</tv>
      <fan>1</fan>
      <ac>1</ac>
      <llamp>1</llamp>
      <flamp>1</flamp>
      <vaccum>0</vaccum>
      <microwave>0</microwave>
      <washer>0</washer>
      <dishwasher>0</dishwasher>
   </device_Count>
   <device_priority>
      <tv>3</tv>
      <fan>2</fan>
      <llamp>5</llamp>
      <flamp>4</flamp>
      <ac>1</ac>
      <vaccum>0</vaccum>
      <microwave>0</microwave>
      <dishwasher>0</dishwasher>
      <washer>0</washer>
   </device_priority>
```

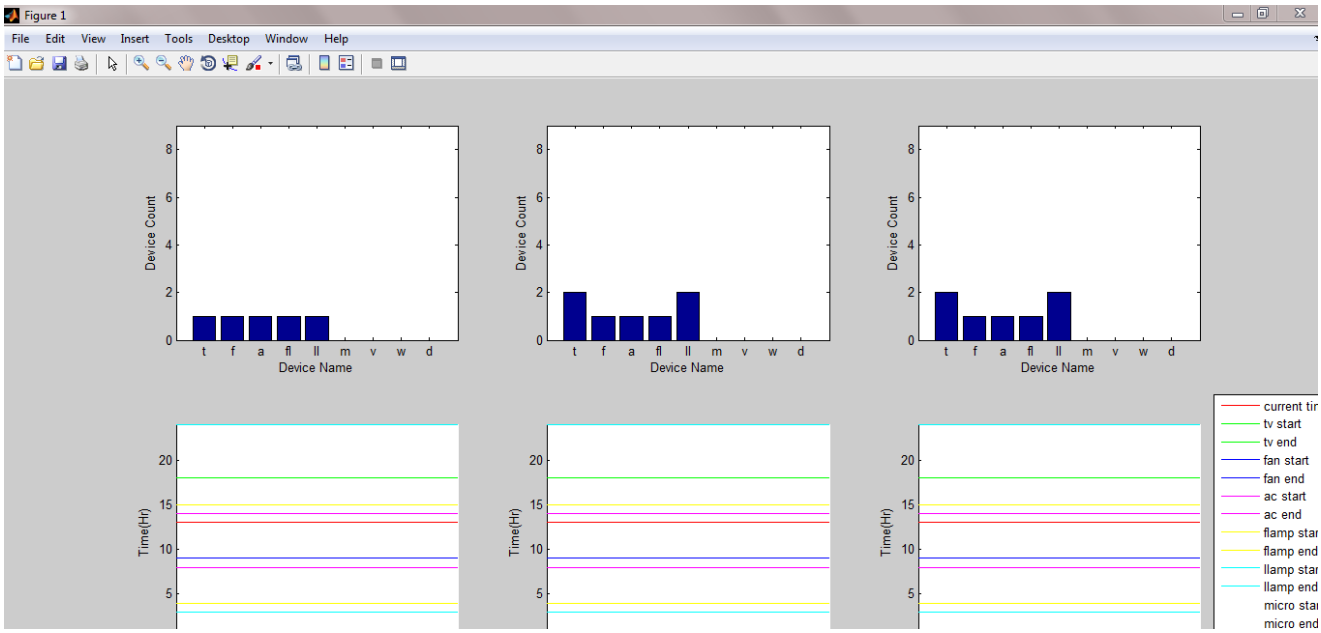

ESM ID: 20136190351


```xml
demandResponse.... × 201538133736.xml × 201538133741.xml × 20153813374
<?xml version="1.0" encoding="utf-8"?>
<ESM_SIMULATION>
   <TimeStamp>08-Mar-2015 13:37:41</TimeStamp>
   <ESM_FILE_NAME>20136190531.xml</ESM_FILE_NAME>
   <Total_Amperage>4200</Total_Amperage>
   <device_Count>
      <tv>2</tv>
      <fan>1</fan>
      <ac>1</ac>
      <llamp>2</llamp>
      <flamp>1</flamp>
      <vaccum>0</vaccum>
      <microwave>0</microwave>
      <washer>0</washer>
      <dishwasher>0</dishwasher>
   </device_Count>
   <device_priority>
      <tv>1</tv>
      <fan>2</fan>
      <llamp>5</llamp>
      <flamp>4</flamp>
      <ac>3</ac>
      <vaccum>0</vaccum>
      <microwave>0</microwave>
      <dishwasher>0</dishwasher>
      <washer>0</washer>
   </device_priority>
```

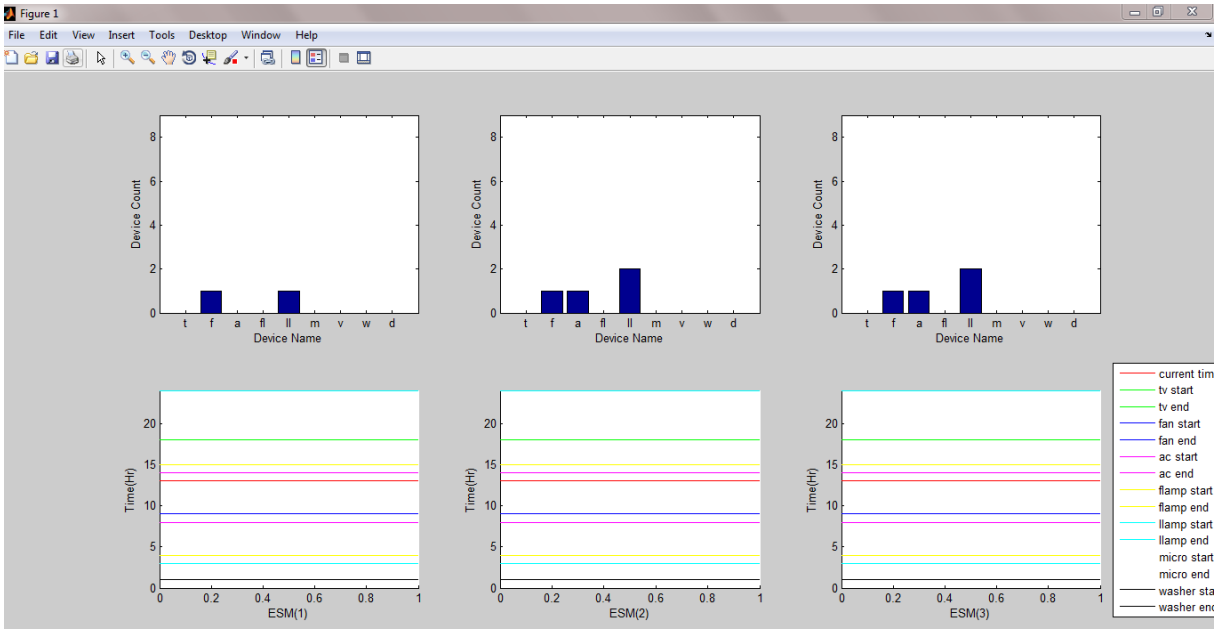
4. *Curtailment request*
   UI where demand (in Wattage), number of ESMs(Counts) and type of AutoDR to simulate is selected



5. Potential curtailment for the selected ESMs with priority and preferences
   Here the three windows show the selections for each of the selected ESMs. The windows depict the number of devices of each device type currently active in the house. The bottom windows show the timestamps of every device set by user along with current time line. Results change after the curtailment run over the devices and their timestamp-priorities.



The following screen shot shows the resulting windows after the end of the curtailment event. All the devices are shut down based on the curtailment control. Loads left (available for curtailment) meet the priority, preference and timestamp selected by the user, thus ensuring maximum consumer satisfaction and minimum discomfort.



6. *Potential curtailment data files*

Data files generated by the simulation are posted back to local ESM hub after the curtailment event is over. Based on the data in this output file the information can be used to curtail devices in the selected homes during a DR event.

ESM ID: 20136190344

demandResponse.... × 201538133736.xml × 201538133741.xml × 201538133745.xml ×


```xml
<?xml version="1.0" encoding="utf-8"?>
<ESM_SIMULATION>
   <TimeStamp>08-Mar-2015 13:40:32</TimeStamp>
   <ESM_FILE_NAME>20136190344.xml</ESM_FILE_NAME>
   <Total_Amperage>3865</Total_Amperage>
   <Total_Loads>5</Total_Loads>
   <Randomize_Engine_Results>
      <Randomized_Load>2</Randomized_Load>
      <device_Count>
         <tv>0</tv>
         <fan>1</fan>
         <ac>0</ac>
         <llamp>1</llamp>
         <flamp>0</flamp>
         <vaccum>0</vaccum>
         <microwave>0</microwave>
         <washer>0</washer>
         <dishwasher>0</dishwasher>
      </device_Count>
   </Randomize_Engine_Results>
</ESM_SIMULATION>
```


ESM ID: 20136190531

demandResponse.... × 201538133736.xml × 201538133741.xml × 201538133745.xml × 2015


```xml
<?xml version="1.0" encoding="utf-8"?>
<ESM_SIMULATION>
   <TimeStamp>08-Mar-2015 13:41:14</TimeStamp>
   <ESM_FILE_NAME>20136190531.xml</ESM_FILE_NAME>
   <Total_Amperage>4200</Total_Amperage>
   <Total_Loads>7</Total_Loads>
   <Randomize_Engine_Results>
      <Randomized_Load>4</Randomized_Load>
      <device_Count>
         <tv>0</tv>
         <fan>1</fan>
         <ac>1</ac>
         <llamp>2</llamp>
         <flamp>0</flamp>
         <vaccum>0</vaccum>
         <microwave>0</microwave>
         <washer>0</washer>
         <dishwasher>0</dishwasher>
      </device_Count>
   </Randomize_Engine_Results>
</ESM_SIMULATION>
```