

Title	Development of a smart community simulator with individual emulation modules for community facilities and houses
Author(s)	Makino, Yoshiki; Fujita, Hiroshi; Lim, Yuto; Tan, Yasuo
Citation	2015 IEEE 4th Global Conference on Consumer Electronics (GCCE): 12-15
Issue Date	2015
Type	Conference Paper
Text version	author
URL	http://hdl.handle.net/10119/14273
Rights	This is the author's version of the work. Copyright (C) 2015 IEEE. 2015 IEEE 4th Global Conference on Consumer Electronics (GCCE), 2015, 12-15. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Description	

Development of a Smart Community Simulator with Individual Emulation Modules for Community Facilities and Houses

Yoshiki Makino^{*†}, Hiroshi Fujita[‡], Yuto Lim[†], Yasuo Tan[†]

^{*}Hokuriku StarBED Technology Center

National Institute of Information and Communications Technology, Ishikawa, Japan

Email: ymakino@nict.go.jp

[†]School of Information Science

Japan Advanced Institute of Science and Technology, Ishikawa, Japan

Email: {m-yoshi,ylim,ytan}@jaist.ac.jp

[‡]Fujitsu Hokuriku Systems, Ishikawa, Japan

Email: hrs-fujita@jp.fujitsu.com

Abstract—A smart community simulator that simulates electricity consumption and thermal energy consumption is being developed by the authors. The simulator is designed to be able to work with modules implemented as individual external emulators of many kinds of facilities and homes. With this simulator, we can change aspects and features of the community and estimate the long-term effects of these changes on the community as a whole. In this paper we introduced the design and implementation of the simulator.

I. INTRODUCTION

With the introduction of renewable energy sources in recent years, supplying stable electrical energy has become a complex task. On the other hand, the idea of "smart communities" has been introduced. Such a smart community enforces power consumption policies on the community as a whole. The smart community encompasses a specific geographical region and now becomes an electrical power consumer in and of itself, able to send and receive energy as well as being able to handle cases of instability regarding energy demand. By including features such energy regulation for each house of the community and being able to distribute energy in the most appropriate way possible, smart communities are expected to achieve better energy control overall.

In this research we introduce an emulation-based simulator for smart communities, which simulates the demand and supply of electrical and thermal energy of the community.

II. SIMULATOR STRUCTURE

In the proposed simulator, various elements of the community are implemented as software modules, suitable for execution on a distributed environment. A depiction of all the simulator's modules can be seen in Fig.1. The simulator is comprised of a management PC node and many PC nodes used for simulation. By using multiple simulation nodes, the simulation proceeds in a distributed manner. We will now discuss the contents and the implementation state of the various modules.

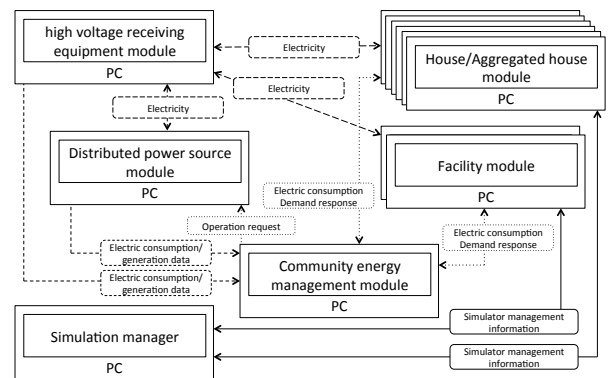


Fig. 1. Simulator structure

Simulation management module

The overall management of the simulation is managed by the management module. This module loads a community profile and sets the correct simulation parameters for all other modules. Furthermore, it sends the first message that signals the start of the simulation and implements further time synchronization functionality in the form of message passing, used by the other modules.

The power consumption results calculated from the other simulator modules are stored to their corresponding internal databases. Currently, SQLite is used as a database stored on a single file. After the end of the simulation, the file containing the results is automatically transferred to a computer node where the management module is operating. The users of the simulator can now confirm the obtained data.

Community energy management module

This module implements the functionality of a Community Energy Management System (CEMS) that manages energy over an entire area, such as a small city or town. This module receives information regarding the consumption of electrical and thermal energy from other modules and, based on this information, it proceeds to issue Demand Response (DR) commands to other modules, thus controlling the total energy power consumption of the simulated community. By changing the settings of this module it is possible to control the overall power consumption trends of the community.

House module

This module is dedicated to house emulation. It calculates the current electrical and thermal power consumption as well as any excess power, and relays this information to community energy module, the high voltage receiving equipment module and the thermal energy module. Moreover, it implements HEMS functionality, receiving the DR commands of the community energy management module and controlling the devices in the house accordingly thus performing energy management in the home. The messaging protocol currently used in the simulation follows a unique format, based on OpenADR[1]. However, if it is necessary, other protocols that are actually being deployed in the real world may be used, assuming that all other modules can use it. In such a case, the simulator can operate in real time, and use a real system as a module in the simulation.

Aggregated house module

This module is used to emulate a group of houses. Its functionality is similar to the house module explained earlier, only this time the computations extend to a given group of houses.

Distributed power source module

This module emulates the various types of power sources available in a community, such as solar panels, cogeneration systems and storage batteries. This module's state is monitored and controlled by many other modules, and its information is sent to the CEMS module. If it is necessary, the CEMS module can request a change in the operation of this module.

III. COMMUNICATION OF MESSAGES

The communication of messages among the modules of the simulator is explained below. The sequence diagram for messaging sending and receiving can be seen in Fig.2. At the beginning, the slave management module operating on the simulation nodes registers itself with the master management module. Next, the master management module requests the start of simulation for each slave module and the slave modules begin operation. At this time, if there are multiple registered simulation nodes, the computational tasks are distributed to the nodes and begin execution. After that, the modules proceed with their processing while keeping their time synchronized. When their simulation is ended, a notification message that states the end of the simulation is sent to the slave management module. When all the modules have finished execution, the

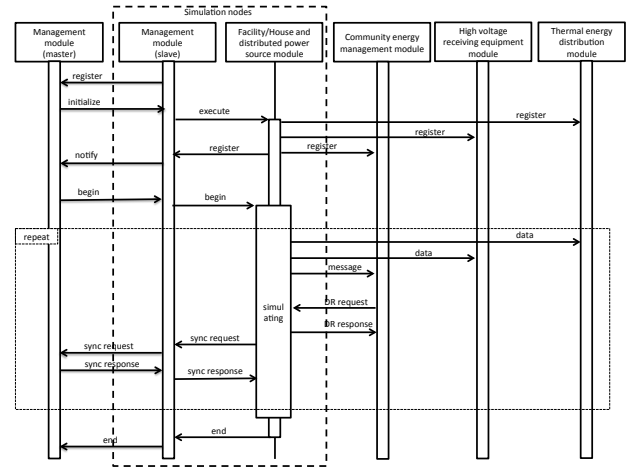


Fig. 2. Communication of messages

slave management modules notify the master management module that the simulation is over.

As it was explained with the house module, the facilities module and the distributed power source module communicate with the community energy module via a simple protocol based on OpenADR. It is possible to change the communication protocol among the modules by altering their implementation. An example message that shows the power consumption can be seen below.

```
<oadr:oadrUpdateReport>
  <pyld:requestID>String</pyld:requestID>
  <ei:venID>vmcent66_16705</ei:venID>
  <oadr:oadrReport>
    <xcal:dtstart>
      <xcal:date-time>
        2014-07-24T00:05:00Z
      </xcal:date-time>
    </xcal:dtstart>
    <xcal:duration>
      <xcal:duration>P5M</xcal:duration>
    </xcal:duration>
    <ei:eiReportID>
      vmcent66_16705_0xe000_HousePlayer_0_0
    </ei:eiReportID>
    <ei:reportRequestID>
      vmcent66_16705_0xe000_HousePlayer_0_0
    </ei:reportRequestID>
    <ei:reportSpecifierID>
      vmcent66_16705_0xe000_HousePlayer_0_0
    </ei:reportSpecifierID>
    <ei:reportName>Wattage</ei:reportName>
    <ei:createdDateTime>
      2015-02-03T20:36:57Z
    </ei:createdDateTime>
    <strm:intervals>
      <ei:interval>
        <xcal:duration>
          <xcal:duration>P5M</xcal:duration>
        </xcal:duration>
        <xcal:uid>
```

```

    <xcal:text>0</xcal:text>
  </xcal:uid>
  <oadr:oadrReportPayload>
    <ei:rID>
      vmcent66_16705_0xe000
    </ei:rID>
    <ei:confidence>95</ei:confidence>
    <ei:accuracy>0.2</ei:accuracy>
    <ei:payloadFloat>
      <ei:value>0.17808219</ei:value>
    </ei:payloadFloat>
    <oadr:oadrDataQuality>
      Good
    </oadr:oadrDataQuality>
  </oadr:oadrReportPayload>
</ei:interval>
</strm:intervals>
</oadr:oadrReport>
</oadr:oadrUpdateReport>

```

This message shows the aggregated 5 minute power consumption was 0.17808219 at a given point in time.

IV. SMART COMMUNITY DEFINITION

In order to use the simulator, it is necessary to install and setup the simulator software on the management and simulation PC nodes. Furthermore, the user must provide a set of configuration files, collectively called the "community profile", that contain information regarding details of the simulated community. After the setup is complete, the user can start the simulation using the management node. The smart community profile is defined using XML and contains 5 types of definitions

Overall community definitions

These definitions pertain to the overall community as a whole. By defining the number and type of elements such as core facilities, shopping malls, houses, groups of houses and distributed power sources of the simulated community, it is possible to describe the community in a simple way. The detailed settings and definitions for each module are stored in separate profile modules.

The following example is an example file showing the settings for a community. According to these setting, the community consists of one school, one shopping mall, 400 houses, 10 aggregated houses, and two types of distributed power sources. Their individual settings are associated with unique xml attributes and their more specific settings are set in other configuration files. With this scheme, it is possible to define various communities with specific characteristics.

```

<?xml version="1.0" encoding="UTF-8"?>
<community>
  <facility>
    <facilitytype type="school1">
      <num>1</num>
    </facilitytype>
  </facility>
  <commercialfacility>
    <commercialfacilitytype type="market1">

```

```

      <num>1</num>
    </commercialfacilitytype>
  </commercialfacility>
  <house>
    <housetype type="house1">
      <num>200</num>
    </housetype>
    <housetype type="house2">
      <num>200</num>
    </housetype>
  </house>
  <condominium>
    <condominiumtype type="cond1">
      <num>10</num>
    </condominiumtype>
  </condominium>
  <communitypowersource>
    <powersourcetype type="powerC1">
      <num>1</num>
    </powersourcetype>
    <powersourcetype type="battC1">
      <num>1</num>
    </powersourcetype>
  </communitypowersource>
</community>

```

Simulation time definition

Here, information regarding the start and end of the simulation time.

Facility module definitions

Each facility has its own profile. The profiles of, for example, a school, a shopping mall, or a house capture different type of information. For example, the information of a house has the following parts: type of house, electrical energy profile, thermal energy profile, contract for electrical information, distributed power sources, various energy management equipment.

An example settings file for a unique house can be seen below. With the ?simprogram? element, the program which will be used in the simulation of the unique house can be selected. As operation settings for this program, the elements ?demandprofile? and ?thermalprofile? can be used. Using these elements it is possible to specify minute settings for energy power consumption and and thermal energy consumption. In this example, the program to be used in this simulation is the program ?houseSimulator?. This program outputs the energy power consumption and thermal energy consumption data specified in the files houseDemandProfile1.csv and houseThermalProfile1.csv respectively. Furthermore with the use of the "facilitypowersource" element it is possible to specify the settings for the distributed power source installed in the house, if present. Finally with the "xems" element, it is possible to define the details regarding the HEMS operation.

```

<?xml version="1.0" encoding="UTF-8"?>
<housespec>
  <housetype>house1</housetype>
  <simprogram>houseSimulator</simprogram>
  <demandprofile>

```

```

    houseDemandProfile1.csv
</demandprofile>
<thermalprofile>
    houseThermalProfile1.csv
</thermalprofile>
<contractdemand>4</contractdemand>
<contractthermal>0</contractthermal>
<facilitypowersource>
    <powersourcetype type="powerH1">
        <num>1</num>
    </powersourcetype>
    <powersourcetype type="battH1">
        <num>1</num>
    </powersourcetype>
</facilitypowersource>
<xems type="hems1"></xems>
</housespec>

```

Community distributed power sources definitions

These are definitions for the distributed power sources that exist in the community. Currently, this includes information regarding the generated and stored electrical power, stored thermal power and their change over time.

Facility equipment definitions

These definitions relate to the equipment presented in the various facilities of the community.

V. IMPLEMENTATION

The core infrastructure of the simulator was designed to be utilized by modules written in various types of languages and was built using C++. Moreover, the message communication part of the simulator is designed as a reusable library, ready to be used by new modules. Finally, a part of the simulation is written in python, a feature that allows for easy modifications and extensions to the simulator.

At this point, the current state of implementation of the facility module will be explained. Currently, the facility module provided by default uses the energy power consumption and thermal energy consumption data stored in a csv file. This default facility module is a program written in python, called by the slave management module.

This module consists of many submodules, with the submodules being reusable and ready to be used in new modules without any changes. The details of the facility module described above can be seen in Fig.3

VI. SIMULATION

The current simulation environment is based on Linux. Any distribution can be used, as long as the necessary libraries and tools needed for compiling the simulator are present. The specifications of the PC nodes are as follows.

- CPU: Intel(R) Xeon(R) CPU E5-2603 v3 1.80GHz
- RAM: 16GB

The total configuration of the simulator is as follows.

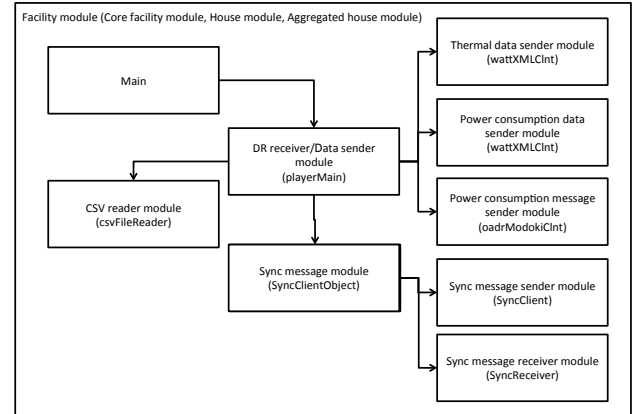


Fig. 3. Implementation of facility module

- Total number of nodes: 6
- 1 node: Management, high voltage receiving equipment, CEMS, thermal energy module
- 5 nodes: Simulation nodes

The simulator was operated with the following settings: 1 real minute corresponds to 4 hours of simulated time, simulation time: 24 hours, core facilities: 1 school, shopping mall: 1 mall, unique houses: 2 types of houses x 400, aggregated houses: 1 aggregation of 200 house and distributed power sources: 2 installations.

VII. CONCLUSION

In this paper we introduce the details of a smart community simulator. The different parts of the simulator are implemented as modules. This leads to a freedom of choices and flexibility during development. We have already implemented various modules, such as thermal simulation module[2][3] and distributed power source module[4]. Furthermore, it is possible to use real communication protocols in the simulator. The smart community is defined using an XML-based community profile. Using this profile we can change the characteristics and behavior of the community easily.

本研究の一部は「先進的ICT国際標準化推進事業(スマートコミュニティにおけるエネルギーマネジメント通信技術)」の委託研究の成果である。

REFERENCES

- [1] <http://www.openadr.org>
- [2] H. Nguyen, Y. Makino, A. O. Lim, Y. Tan, Y. Shinoda, "Implementation and Evaluation of Thermal Simulator for Houses", IEICE technical report. IN, vol. 112, no. 4, pp. 31-36, 2012.
- [3] Y. Makino, H. Nguyen, J. Kim, A. O. Lim, Y. Tan, "Design and Implementation of Home Simulator and Experimental House", "Production Facility Control", IEE, 2014.
- [4] Y. Liu, H. Fujita, Y. Makino, Y. Lim, Y. Tan, "Real Time Simulation of Hybrid Distributed Energy Resource Using Solar PV and Wind Power", 2015 IEICE GENERAL conference, 2015