

PowerEnergy2018-7295

ENERGYPLUS INTEGRATION INTO CO-SIMULATION ENVIRONMENT TO IMPROVE HOME ENERGY SAVING THROUGH CYBER-PHYSICAL SYSTEMS DEVELOPMENT

Joe Singer*, Thomas Roth**, Chenli Wang*, Cuong Nguyen**, Hohyun Lee*

*Santa Clara University
Department of Mechanical Engineering
Santa Clara, California, USA, 95053

**National Institute of Standards and Technology,
Gaithersburg, MD, USA, 20899

ABSTRACT

This paper presents a co-simulation platform which combines a building simulation tool with a Cyber-Physical Systems (CPS) approach. Residential buildings have a great potential of energy reduction by controlling home equipment based on usage information. A CPS can eliminate unnecessary energy usage on a small, local scale by autonomously optimizing equipment activity, based on sensor measurements from the home. It can also allow peak shaving from the grid if a collection of homes are connected. However, lack of verification tools limits effective development of CPS products. The present work integrates EnergyPlus, which is a widely adopted building simulation tool, into an open-source development environment for CPS released by the National Institute of Standards and Technology (NIST). The NIST environment utilizes the IEEE High Level Architecture (HLA) standard for data exchange and logical timing control to integrate a suite of simulators into a common platform. A simple CPS model, which controls local HVAC temperature set-point based on environmental conditions, was tested with the developed co-simulation platform. The proposed platform can be expanded to integrate various simulation tools and various home simulations, thereby allowing for co-simulation of more intricate building energy systems.

INTRODUCTION

Heating, ventilation and cooling (HVAC) is the largest source of residential energy consumption in United States, encompassing about 25% of total residential energy usage. A significant portion of energy is wasted by unnecessary operation, such as overheating/overcooling or operation without occupants. Based on the recent paper by Nguyen and Aiello [1], energy conscious behaviors in residential homes can lead to 33% less energy consumption compared to the design point, and 50% less energy consumption compared to those demonstrating wasteful

behaviors. Incorporating intelligence into home appliances to make them Cyber-Physical Systems (CPS) can allow wasteful devices to exhibit energy conscious behaviors. CPS are systems involving interactions between computation and physical components that process information collected from sensors to control physical actuators [2]. For example, a smart HVAC system can consume less energy by replacing fixed operating set-points with dynamic operation based on occupancy sensor information. Assuming residential homes achieve 33% consumed energy reduction through CPS appliance operation, an upwards of 3.6 quadrillion BTUs of energy can be saved nationally in the United States' residential sector [3].

Several products currently manipulate appliance operation via CPS paradigm [4, 5]. Unfortunately, their functionality is limited to remote controllability because implementation of automated control algorithms is challenging. Physical building interactions required complex testing and validation that cannot be captured in a single CPS component. Such experimental CPS necessitates interdisciplinary knowledge and real-time data collection, which requires significant amounts of time and resources [6]. A co-simulation platform involving a building simulator that combines multiple CPS components can address the aforementioned challenges in developing and implementing automated building controls.

The National Institute of Standards and Technology (NIST) developed an open-source CPS experiment and testing environment called Universal CPS Environment for Federation (UCEF). Its graphical user interface is designed to make co-simulations and experiments for CPS product simple and available. UCEF integrates various simulation entities (federates) sourced in different development environments, which has traditionally been challenging to accomplish [7]. UCEF leverages the IEEE's High Level Architecture (HLA) standard [8] for its communication protocol, implemented by the

Portico Run-Time Infrastructure (RTI) [9], to achieve logical timing control and data transfer among a collection of federates (defined as a federation). UCEF remains under development and supports few simulation tools. Integrating a building specific simulation tool into UCEF as a new supported federate type can lead to simpler building automation experimentation and validation.

An open-source building simulation tool called EnergyPlus [10] can be integrated with UCEF to achieve data exchange in an CPS test environment. EnergyPlus, a widely used modeling tool created by the Department of Energy, can evaluate building energy consumption at sub-hourly time steps. The software calculation capabilities incorporate key building parameters such as user activity, HVAC systems, building composition, and more. In CPS testing, EnergyPlus can replace physically measured building information with a simulated model to accelerate CPS tool development.

The present work integrates EnergyPlus into UCEF. An EnergyPlus model will communicate building information to the RTI using a UCEF Java federate. Although this work can benefit any building control system, a simple model was tested to verify co-simulation capability. A HVAC set-point algorithm implemented in another Java federate will receive environment temperature from EnergyPlus and return HVAC set-points to EnergyPlus. Intelligent set-point control of an HVAC system can significantly reduce energy consumption in a residential building between 33%-50% [1, 11], providing a good use case for the developed platform. Further, other simulators integrated with UCEF can expand HVAC controllability to include pre-heating or pre-cooling a collection of homes to reduce excessive power draw during peak demand [12]. Enabling UCEF co-simulation with EnergyPlus creates a simple approach to reduce residential energy consumption by allowing automation and optimization of wasteful appliances.

ACRONYMS

BTU	British Thermal Unit
CPS	Cyber-Physical Systems
FMI	Functional Mockup Interface
FMU	Functional Mockup Unit
HLA	High Level Architecture
HVAC	Heating, Ventilation, and Air Conditioning
IDF	Input Data File
IEEE	Institute of Electrical and Electronic Engineers
NIST	National Institute of Standards and Technology
RTI	Run-Time Infrastructure
TCP/IP	Transmission Control Protocol/Internet Protocol
UCEF	Universal CPS Environment for Federation
XML	Extensible Markup Language

APPROACH

EnergyPlus currently has an existing co-simulation interface through the Functional Mock-up Interface (FMI) standard created by Modelisar [13]. The standard accomplishes

interoperability by connecting simulation platforms to an external model by use of a zip file (with extension *.fmu) known as a Functional Mock-up Unit (FMU). The zip file contains three elements: an Extensible Markup Language (XML) file, compiled C code binaries, and optional documentation for data exchange. The XML file establishes interfacing data, the C code orchestrates data exchange, and the documentation can define and specify operation. Unfortunately, the FMI and HLA standards have different notions of time management, and UCEF does not support data exchange using FMI. To bridge the two standards, we create an FMU with capabilities for bi-directional communication between an EnergyPlus model and a UCEF Java federate. The Java federate will be customized to wrap EnergyPlus models for data exchange to an RTI federation. This data communication, represented in Figure 1, is done through TCP/IP socket communication between our FMU and Java federate.

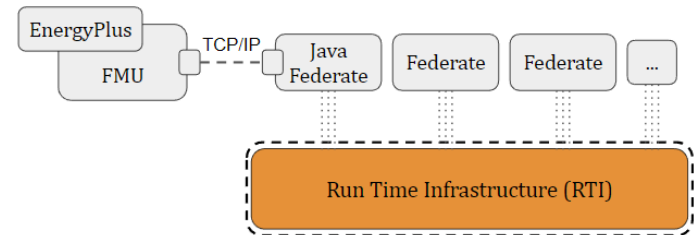


Figure 1: EnergyPlus has capability to interface with a FMU. Using TCP/IP socket communication inside a generic FMU allows for connectivity to a UCEF Java federate for the HLA RTI data exchange.

Connecting EnergyPlus to an FMU involves specific modifications of an EnergyPlus input data file (IDF). An IDF defines parameters to perform building energy simulations, such as building materials, components, and equipment. Using an IDF component called *FunctionalMockupUnitImport*, co-simulation is linked between EnergyPlus and the FMU. This component initializes the FMI master and slave architecture where slaves are coordinated and executed by the master program. EnergyPlus acts as the master in this configuration, which initializes the FMU as an executable slave instance.

Upon simulation start with the aforementioned component, EnergyPlus locates and unpacks the linked FMU to begin processes represented in Figure 2. Execution of the FMU's customized C binaries is controlled by EnergyPlus to run select FMI functions [14] that have been modified and implemented to exchange data with the HLA RTI. EnergyPlus first calls the *fmiInstantiateSlave* function to parse through the unpacked XML file to properly allocate memory for interface data. Next, the *fmiInitializeSlave* function uses TCP/IP sockets to establish connection to a server hosted in the UCEF Java federate. After TCP/IP connection is verified, EnergyPlus's time step calculations begin.

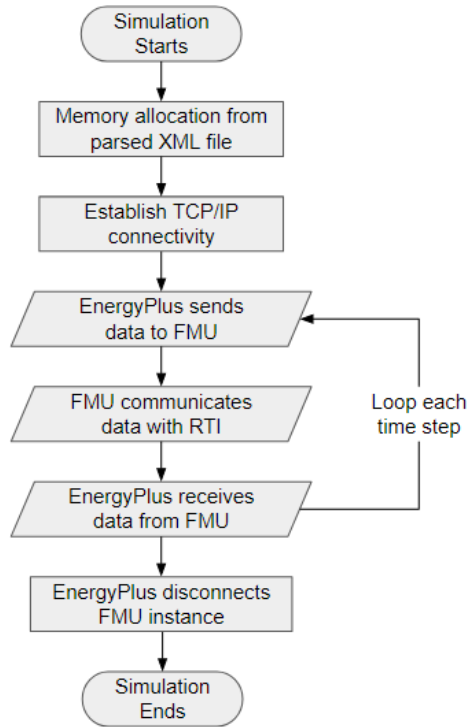


Figure 2: EnergyPlus as a master program for the Functional Mockup Interface (FMI) standard calls select functions throughout simulation to perform specific tasks. At each time step, three functions are called to transfer EnergyPlus data to a Functional Mockup Unit (FMU).

At each time step, EnergyPlus sends data to the FMU as a *real* data type using the function *fmiSetReal*. The FMU will then utilize socket connection in *fmiDoStep* to send the EnergyPlus data (as a concatenated string) to the designated Java federate. The format of this string is standardized and represented as follows:

```

HEADER\r\nTIMESTAMP\r\nNAME\r\nVALUE\r\n....
NAME\r\nVALUE\r\n\r\n
  
```

The “HEADER” defines handling procedures of the string. Data sent from FMU to the Java federate will either contain the header “UPDATE” or “TERMINATE”. An “UPDATE” header is used at each EnergyPlus time step to signify incoming information to the Java federate. A “TERMINATE” header informs the Java federate that EnergyPlus simulation has ended. Data received by the FMU from the Java federate will either contain “SET” or “NOUPDATE” headers. “SET” indicates federation interactions will change EnergyPlus variables, and “NOUPDATE” indicates no variables will change. After the header, the “TIMESTAMP” communicates simulation time (in seconds) for logical time management. Next, for each “UPDATE” and “SET” header, “NAME” and “VALUE” respectively represent each variable name and corresponding value of interfacing data defined through the XML file. Each piece of information is separated by a carriage return followed

by a line feed (“\r\n”). Two consecutive carriage returns and line feeds at the end signify the end of the string.

EnergyPlus will remain in the *fmiDoStep* function until the Java federate responds with a concatenated string. After a string is returned, the FMU will parse through the returned string in *fmiDoStep*. The *fmiGetReal* function passes received information back into the EnergyPlus model as a *real* data type. The described data exchange pipeline is represented in Figure 3. The master EnergyPlus program will exchange data with the Java federate at each time step. After the final time step, the FMU slave instance is disconnected, and the simulation ends.

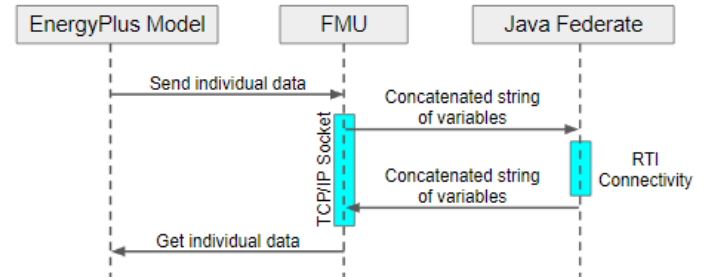


Figure 3: UML diagram representing data communication between the master EnergyPlus program and a UCEF Java federate via FMU slave instance.

The Java federate developed in UCEF communicates information between the FMU slave instance and the RTI. This federate begins by hosting a TCP/IP server for the FMU client connection. During simulation, the federate parses each received string from the FMU and passes its information to the RTI federation. The federate then waits for messages from the RTI that should be sent to EnergyPlus. A concatenated string containing the content of these messages, is then returned to the FMU client.

CONNECTIVITY VALIDATION

A series of simulations were executed to validate EnergyPlus communication with an HLA RTI federation. A simple three-room house model was created in an EnergyPlus IDF. The home was modeled as being located in San Francisco, CA, USA using weather information from June 2017. This location is chosen because hourly temperature change is high enough to potentially cause unnecessary heating or cooling. The home was equipped with a dual set-point HVAC system operating at a temperature range between 21° and 23° to simulate arbitrary occupant comfort range. The first simulation executed the standard EnergyPlus model without the implemented FMU external interface. Environmental temperature, zone temperature, and HVAC energy usage information were recorded at each time step. Resulting HVAC energy consumption using these “naive” set-points can resemble non-energy conscious behaviors.

The second simulation directly supplied environment temperature data through to a standalone thermostat controller algorithm. A simple algorithm in Java was created to adjust

heating and cooling temperature set-points based on environment temperature. When outside temperature is either high or low, the heating and cooling setpoints can adjust correspondingly to minimize HVAC loading and to preheat/precool the house. EnergyPlus and UCEF were not used in this second simulation. Rather, environment temperature recorded in the first simulation was directed through the thermostat controller algorithm to return dynamic dual set-points. The results of this second experiment act as a ground truth for the next experiment implementing EnergyPlus and UCEF connectivity.

The final simulation implemented the FMI external interface with the IDF described in the first simulation. The *FunctionalMockupUnitImport* class enforced data exchange with our developed FMU, linking EnergyPlus to our Java federate. The federation was created using UCEF, binding EnergyPlus and the secondary thermostat controller algorithm through RTI. Shown in Figure 4, environment temperature from EnergyPlus was sent through RTI to the thermostat controller at each time step. Before time step progression, the controller returned a heating and cooling set-point to the HVAC system in EnergyPlus. HVAC set-points and zone temperature were recorded.

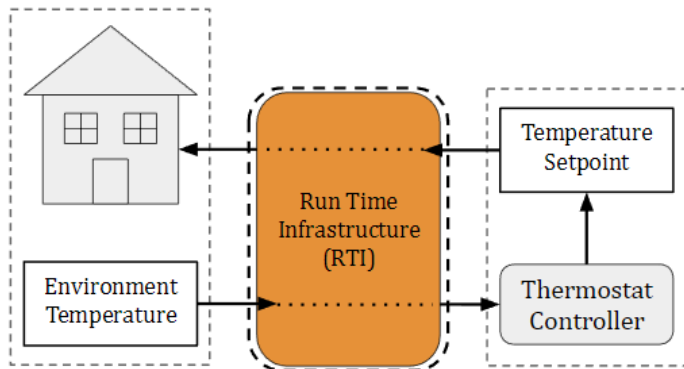


Figure 4: Representation of the data transfer using Run Time Infrastructure between the EnergyPlus Java Federate and the thermostat controller Java federate.

RESULTS & DISCUSSION

The first simulation recorded sub-hourly temperature and HVAC energy consumption data of a simple EnergyPlus model. Dual set-point of an HVAC system between 21 °C and 23 °C caused activation. Heating activated in the morning and evening, and cooling activated mid-day. Figure 5 shows the zone temperature fluctuation throughout the design day. Figure 6 represents respective heating and cooling energy consumed by the system. HVAC operation between the narrow temperature range represents excess consumed energy by a non-energy conscious user. The following simulations attempt to incorporate in intelligent CPS to control the model HVAC.

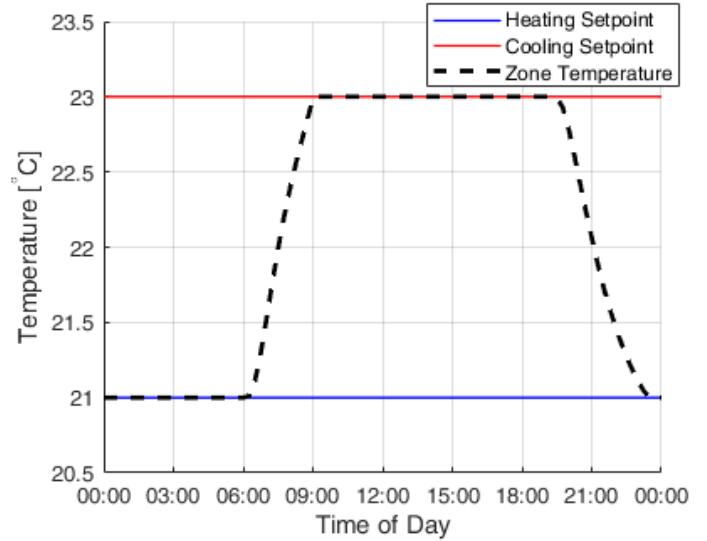


Figure 5: Simulation results of temperature fluctuation controlled by constant heating and cooling setpoints input.

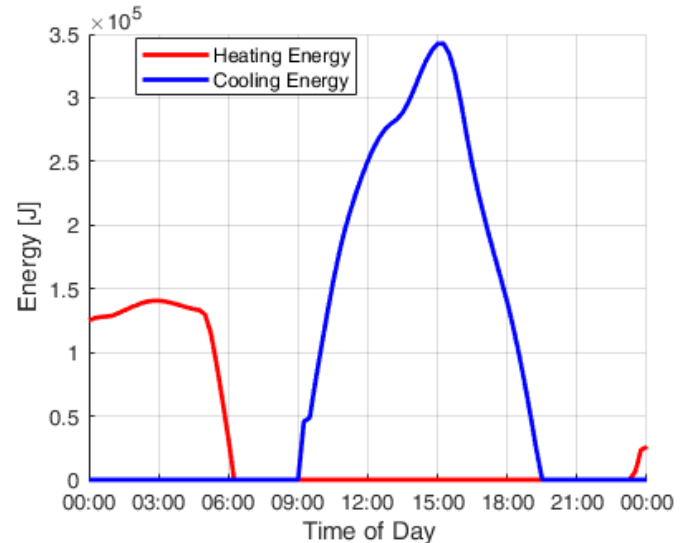


Figure 6: Heating and cooling power consumptions respective to HVAC operation controlled by setpoints (Figure 5).

A thermostat controller output dynamic set-points based on environment temperature. Direct input of the second simulation and EnergyPlus/RTI input of the third simulation yielded identical results, shown in Figure 7. Matching outputs of the two simulations validates continuous and accurate EnergyPlus integration with UCEF.

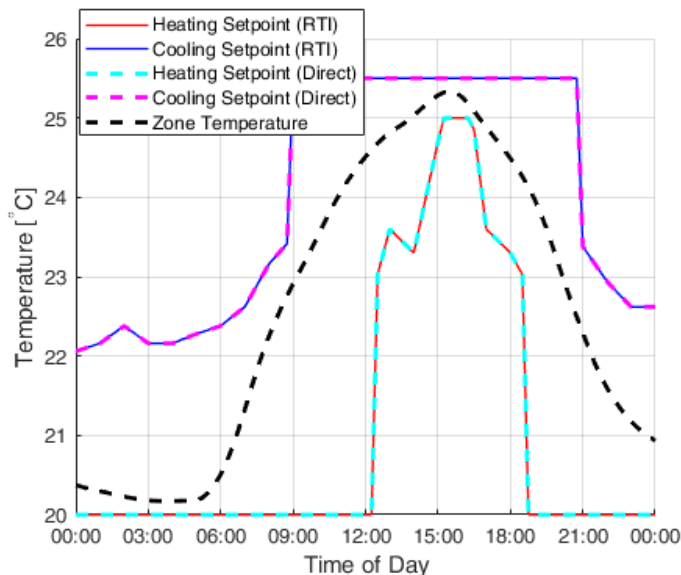


Figure 7: HVAC heating and cooling set-points based on external thermostat controller. Direct connection based and RTI connection yield consistent results.

Figure 7 also shows internal zone temperature of the EnergyPlus model. Dynamic thermostat controller outputs cause zero energy consumption for this simulation day. Compared to the naive set-points of the first simulation occupant, EnergyPlus co-simulation with the intelligent thermostat controller removed unnecessary energy consumption. Results verify UCEF integration does not impact simulated results.

CONCLUSION & FUTURE WORK

This paper developed an open-source integration of a building simulation software with UCEF for the design and validation of CPS. By developing a simple FMU with a TCP/IP connection to a Java federate, calculated data at each time step was communicated from an EnergyPlus model to an HLA federation. This successful integration allows co-simulation between EnergyPlus models and HLA federates. Simulated results validate UCEF-based federations can exchange data with EnergyPlus models without negative impact on results. More complex control algorithms and other simulation tools integrated into EnergyPlus creates an environment that can produce sophisticated CPS that reduce energy consumption in residential buildings. Integration of EnergyPlus into UCEF as a new federate type enhances the platform's capabilities through added support of building simulations.

Additional concepts can be further investigated for more robust development. Modifications of FMU configuration files may be necessary for different simulation designs requiring different building model information. Currently, the IDF and the XML file need to be created manually based on the desired interface data. UCEF has support for the automatic generation of configuration files based on the content of fields in its graphical user interface. A user should be able to enter desired EnergyPlus variable information directly into the UCEF interface to

automatically generate and update the IDF and XML file, rather than having to write the files themselves. Future work could address this usability feature through extensions to the UCEF graphical interface.

The presented approach using TCP/IP sockets could be further leveraged to integrate other FMI tools into UCEF. FMUs connected to other programs can utilize the TCP/IP concatenated string protocol to communicate with the Java federate in UCEF. Expanding co-simulation diversity to FMI connected tools can vastly improve UCEF simulator and emulator inventory. UCEF integration can increase development effectivity by allowing for improved logical timing control of these FMI tools.

ACKNOWLEDGMENTS

Portions of this publication and research efforts are made possible through the support of NIST via federal award #70NANB17H039.

Official contribution of the National Institute of Standards and Technology; not subject to copyright in the United States. Certain commercial products are identified in order to adequately specify the procedure; this does not imply endorsement or recommendation by NIST, nor does it imply that such products are necessarily the best available for the purpose.

REFERENCES

- [1] Nguyen, T. A., and Aiello, M., 2013, "Energy intelligent buildings based on user activity: A survey," *Energy and Buildings*, 56, pp. 244-257.
- [2] Griffor, E. R., Greer, C., Wollman, D. A., and Burns, M. J., June 2017, "Framework for Cyber-Physical Systems: Volume 1, Overview," <https://dx.doi.org/10.6028/NIST.SP.1500-201>.
- [3] Lawrence Livermore National Laboratory, 2016, "Energy Flow Charts: Charting the Complex Relationships among Energy, Water, and Carbon," <https://flowcharts.llnl.gov/>.
- [4] Alhafidh, B. M. H., and Allen, W. H., 2017, "High Level Design of a Home Autonomous System Based on Cyber Physical System Modeling," *IEEE*, p. 45.
- [5] Hong, T., Sun, H., Chen, Y., Taylor-Lange, S. C., and Yan, D., 2016, "An occupant behavior modeling tool for co-simulation," *Energy & Buildings*, 117, pp. 272-281.
- [6] Terpening, E. D., and Littleton, A., 2017, "The State of Internet of Things in the Home: PART II: OPPORTUNITIES AND CHALLENGES FOR BRANDS SELLING IOT PRODUCTS FOR THE HOME," *Altimeter Group - Research Reports*, p. 1.
- [7] Roth, T., Song, E., Burns, M., Neema, H., Emfinger, W., and Sztipanovits, J., 2017, "Cyber-Physical System Development Environment for Energy Applications," *ASME*

2017 11th International Conference on Energy Sustainability, Charlotte, North Carolina, USA.

[8] IEEE, 2010, "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-- Framework and Rules," IEEE Std 1516-2010, pp. 1-38.

[9] Pokorny, T., and Fraser, M., 2017, "The Portico Project," <http://timpokorny.github.io/public/index.html>.

[10] US Department of Energy's Building Technologies Office, and National Renewable Energy Laboratory (NREL), 2017, "EnergyPlus," <https://energyplus.net/>.

[11] Hong, T., and Lin, H.-W., 2013, "Occupant Behavior: Impact on Energy Use of Private Offices," Conference: ASim 2012 - 1st Asia conference of International Building Performance Simulation Association, Shanghai, China.

[12] Liu, Y., Qiu, B., Fan, X., Zhu, H., and Han, B., 2016, "Review of Smart Home Energy Management Systems," Energy Procedia, 104, pp. 504-508.

[13] Junghanns, A., 2017, "FMI Functional Mock-up Interface," <http://fmi-standard.org/>.

[14] Modelisar, 2010, "Functional Mock-up Interface for Co-Simulation v1.0 (Documentation)."