Energy and Buildings xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Energy and Buildings

journal homepage: www.elsevier.com/locate/enbuild

Building energy simulation in real time through an open standard interface

Xiufeng Pang*, Thierry S. Nouidui, Michael Wetter, Daniel Fuller, Anna Liao, Philip Haves

Lawrence Berkeley National Laboratory, United States

ARTICLE INFO

Article history:
Received 22 July 2015
Received in revised form 10 October 2015
Accepted 12 October 2015
Available online xxx

Keywords:
Building energy simulation
Real-time
Functional Mockup Interface
BCVTB
EnergyPlus

ABSTRACT

Building energy models (BEMs) are typically used for design and code compliance for new buildings and in the renovation of existing buildings to predict energy use. The increasing adoption of BEM as standard practice in the building industry presents an opportunity to extend the use of BEMs into construction, commissioning and operation. In 2009, the authors developed a real-time simulation framework to execute an EnergyPlus model in real time to improve building operation. This paper reports an enhancement of that real-time energy simulation framework. The previous version only works with software tools that implement the custom co-simulation interface of the Building Controls Virtual Test Bed (BCVTB), such as EnergyPlus, Dymola and TRNSYS. The new version uses an open standard interface, the Functional Mockup Interface (FMI), to provide a generic interface to any application that supports the FMI protocol. In addition, the new version utilizes the Simple Measurement and Actuation Profile (sMAP) tool as the data acquisition system to acquire, store and present data. This paper introduces the updated architecture of the real-time simulation framework using FMI and presents proof-of-concept demonstration results which validate the new framework.

Published by Elsevier B.V.

1. Introduction

Energy modeling is commonly used to help engineers, architects and owners design and integrate different building elements, such as the envelope, mechanical and electrical systems. As the standard for testing simulation tools [1] and the methods used to develop an energy model [2] evolve, a carefully created model can predict actual performance more accurately, which makes it possible to utilize energy simulation throughout the life cycle of a building. Pang et al. [3] reported the development of a framework that allows a comparison of building actual performance and expected performance in real time, as shown in Fig. 1. The realization of the framework utilized EnergyPlus [4], the Building Controls Virtual Test Bed (BCVTB) [5,6], and a Building Automation System (BAS). EnergyPlus is used to model and simulate the building, the BCVTB acts as the middleware, and the BAS serves as the data acquisition system to collect the building and environment measurements. By real-time simulation, as opposed to off-line simulation, we mean the synchronization of the simulation time with real-time, as represented by the computer clock [3].

The BCVTB is a freely available open-source software environment that allows users to couple different simulation programs for co-simulation. The BCVTB also allows simulation programs to be coupled to actual hardware. The BCVTB is a special configuration of Ptolemy II, an open-source framework for actor-oriented design, with the addition of specific actors relevant for the building industry [5–7]. Ptolemy II provides the synchronization of simulation time to real-time.

EnergyPlus is a detailed whole building energy simulation program that calculates the building heating and cooling loads and outputs disaggregated energy end-use as well as many other simulation variables that are necessary for a detailed comparison of the simulated and actual building performance [8].

A few applications that use the previously published work have been reported [9,10]. However, there are two main limitations associated with the previous realization of the real-time simulation framework:

- It only works with the models generated from the supported simulation tools, such as EnergyPlus, Dymola and TRNSYS. The running of the models requires the installation of the simulation software or a runtime license in the computer that runs the real-time simulation platform.

* Corresponding author.
E-mail address: XPang@lbl.gov (X. Pang).

http://dx.doi.org/10.1016/j.enbuild.2015.10.025
0378-7788/Published by Elsevier B.V.

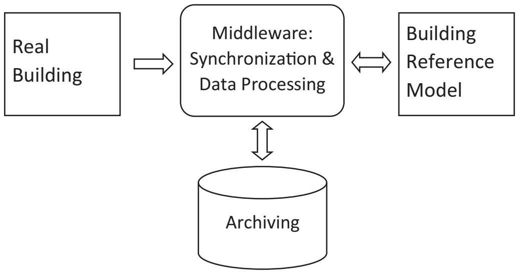

Fig. 1. Illustration of the framework.

- It can only communicate with hardware through BACnet [11] or an analog input/output (I/O) device [12].

This paper introduces a major new realization of the real-time simulation framework that addresses these and other limitations.

2. The new platform

The BCVTB has been further developed to support the import of simulation models or tools that implement versions 1.0 and 2.0 of the Functional Mockup Interface (FMI), which provides a standardized method of encapsulating models and simulation tools (see below). The resulting encapsulated models and tools are referred to as Functional Mockup Units (FMU). This extension allows the BCVTB to be used to couple simulation models and tools that implement the FMI interface with simulation models and tools that do not implement it. This new capability has enabled the development of the new real-time simulation platform. Fig. 2 shows the evolution of the realization of the real-time simulation framework. As can be seen in Fig. 2, the new platform uses the Simple Measurement and Actuation Profile (sMAP) and the FMI to address the two main limitations described above.

The BCVTB uses its Synchronous Data Flow (SDF) director to control the communication between different actors. Actors are BCVTB software components that execute concurrently and share data with each other by sending messages via ports [13]. With respect to FMI, the SDF director is the master algorithm that coordinates the data-exchange between FMU actors and other actors. In the new platform, the data-exchange occurs at a fixed time step, which can be specified by the user. The SDF director is responsible for advancing and synchronizing its local time to the computer clock time for real-time applications. If variable time-stepping is required, Ptolemy II supports co-simulation with adaptive time steps for example by using the Discrete Event director. However, EnergyPlus only supports fixed time step co-simulation, which is why we used the SDF director. The minimum time step which can be handled by the new platform is dictated by the minimum time step supported by the simulation tool that generates the FMU.

2.1. sMAP

sMAP is a protocol for publishing time-series data from a wide variety of sensors. The open-source Python-based sMAP library developed at the University of California, Berkeley includes software modules to interface to various meters and sensors, set up a backend database for data archiving and provide a front-end visualization plotter to display real-time and archived data. As shown in Fig. 3, the sMAP architecture consists of several components that may be used together to capture, transmit, store, and present time-series data [14]:

- The sMAP Sources communicate with existing instrumentation, and collect time-series data using a wide variety of underlying protocols.
- The sMAP Archiver is a high-performance database for storing large volumes of data. It provides a simple and powerful interface for making use of historical data and real-time data and it locates and cleans the raw data.
- The applications make use of historical and/or real-time data in order to achieve user objectives, for instance, providing attractive visualizations and computing optimal control strategies. For plotting data streams from the sMAP Archiver, sMAP uses the powerdb2 web-browser-based visualizer.

Fig. 4 shows how sMAP sources work. Each source contains one or more drivers, which contain code to communicate with the instruments. A list of available drivers can be found in [14], including Modbus, BACnet and HTTP/XML. Another feature of sMAP sources is that it is easy to write new drivers for any unsupported instruments. Description of other components in Fig. 4 can be found in [14].

The new real-time simulation platform invokes sMAP through the system command in the BCVTB using python scripts, which allows communication with a wide variety of instruments. In this way, the new real-time simulation platform successfully addresses the limitation regarding hardware communication noted above.

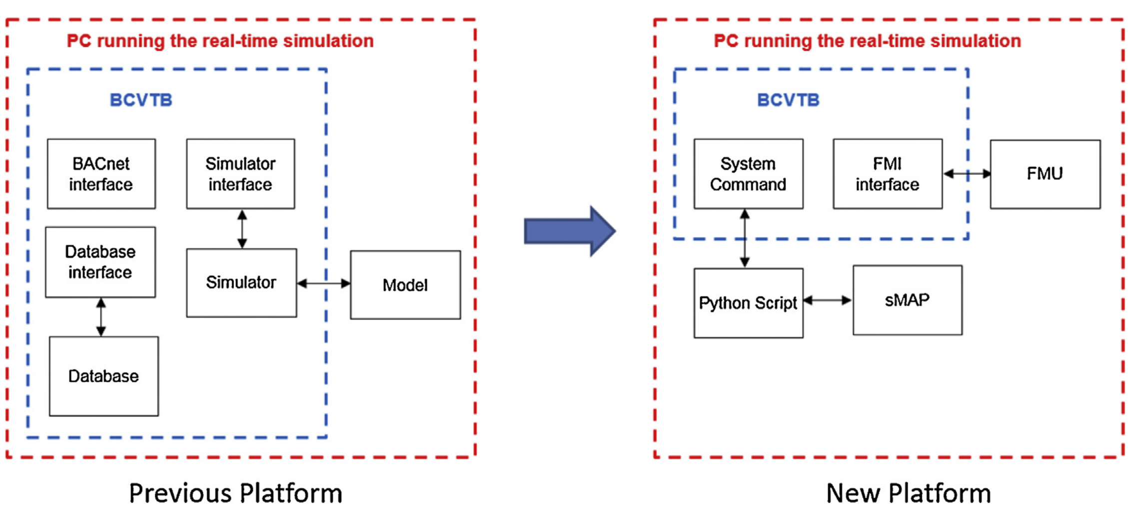

Fig. 2. Evolution of the real-time simulation platform.

Please cite this article in press as: X. Pang, et al., Building energy simulation in real time through an open standard interface, Energy Buildings (2015), http://dx.doi.org/10.1016/j.enbuild.2015.10.025