

# Demo Abstract: BUSICO 3D - Building Simulation and Control in Unity 3D

Jonathan Fürst  
IT University of Copenhagen  
jonf@itu.dk

Philippe Bonnet  
IT University of Copenhagen  
phbo@itu.dk

Gabe Fierro  
UC Berkeley  
gt.fierro@berkeley.edu

David E. Culler  
UC Berkeley  
culler@berkeley.edu

## Abstract

In this demonstration, we present a novel system of building control and simulation focused on the integration of the physical and virtual worlds. Actuations and schedules can be manifested either in a physical space or in a virtualization of that space, allowing for more natural interactions with simulations and easier transferring of schedules and configurations from the simulated virtual environment to a real-world deployment. We provide an implementation using a widely used game engine (Unity 3D) and sMAP (Simple Measurement and Actuation Profile), a developed time series database and metadata store.

## 1 Introduction

Buildings are increasingly turning into systems of human-computer-building interaction via software. People are beginning to use mobile apps to control their devices like lights or thermostats. The ‘Internet of Things’ has spawned many new devices that people can interact with, such as the Philips Hue LED bulb and Nest thermostat. At the same time, there have been research efforts towards making existing building instrumentation accessible in a uniform way and integrating it with this new generation of smart devices, e.g., [1], [2]. Many of these projects provide relatively mature solutions for abstracting physical sensors and actuators to a uniform interface, but the question of how inhabitants or building managers access this physical information is still not very well developed. Interfacing with a building often requires programming or technical experience specific to that system. Furthermore, access to a Building Automation System (BAS) is often restricted because running applications or experiments on a building or changing its configuration can be harmful.

With BUSICO 3D, we present a system that provides: (i) a virtual representation of a building as a more natural interface for human interaction, and (ii) an integration of simulation and control that allows the testing of configurations without any harm on the physical building or personal environment (see Figure 1). BUSICO 3D uses sMAP as an abstraction layer for accessing and describing various sensors and actuators of the physical building. The structure of the physical building is derived from an imported BIM (Building information modeling) model. The system can be run in either simulation or control mode. In simulation mode, people can change settings, define rules or schedules, and simulate the effect of these on the building. Feedback on the building’s state, including temperature and energy consumption, is provided through the virtual experience. In control mode, feedback of the simulation engine is replaced by feedback of the actual building. In the following, we describe our prototype design and implementation using a widely used game engine (Unity 3D) and existing platforms for modeling the physical structure of a building (Autodesk Revit).



Figure 1. BUSICO 3D from birds-eye perspective.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.

Copyright is held by the owner/author(s).

SenSys’14, November 3–5, 2014, Memphis, TN, USA.  
ACM 978-1-4503-3143-2/14/11.  
<http://dx.doi.org/10.1145/2668332.2668380>

## 2 Design and Implementation

Here we explain briefly the different components and working of our system. We finish by describing a typical interaction with our system. We have built a physical testbed for our system that resembles two rooms of a building (see Figure 2).

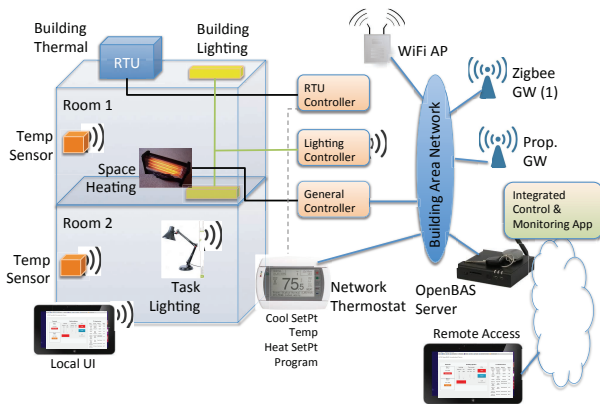


Figure 2. Schematic view of our testbed.

## 2.1 Components

BUSICO 3D uses Unity 3D as an execution environment. Unity is a game engine that allows a fast implementation of games and other virtualizations by providing a common set of features like graphics, lighting or physics engine. Unity allows for scripting of the underlying engine in several high level languages (C#, JavaScript and Boo). BUSICO 3D is entirely written in C#. It allows the import of a CAD architectural model of the building that is created in a modeling program of choice (Autodesk Revit, Graphisoft ArchiCAD, Sketchup). sMAP acts as a hardware abstraction layer on top of various physical devices. sMAP drivers abstract away the particulars of sensors and actuators and expose a uniform REST interface [1]. BUSICO 3D reads this REST interface of sensors and actuators and maps it by means of sMAP metadata to the CAD model, creating automatically objects for each found sensor or actuator including GUI, programming logic and renderer.

## 2.2 Simulation and Control Engine

BUSICO 3D is built around an integrated simulation and control engine that follows the working of discrete event simulation. Every object (e.g., an object representing a thermostat) has a `Start()` method and an `Update()` method. `Start()` gets called once when the object is created, `Update()` gets called every frame by the Unity engine. Our system adds to this the state-effect pattern ([3]) by dividing calls to `Update()` in an ‘effect’ and ‘state’ phase. In the ‘effect’ phase, all objects are calculating effect values based on the state values of themselves and related objects. These effect values are then saved for the ‘state’ phase. Here, objects are accessing effect values to calculate their new states.

## 2.3 Schedules and Rules

BUSICO 3D includes an event-driven rule engine. A user may specify simple rules in the form of:

```
IF (condition) THEN (consequence);
```

An example could for instance be:

```
IF (window1==open)
  THEN (thermostat1_mode=OFF);
```

BUSICO 3D also includes a scheduler in which users can set different schedules for their lights or temperature in a building. Scheduling can also be event driven, in which case events are points in time.

```
IF (time==2014-07-27T07:48:42)
  THEN (thermostat1_temp=42);
```

## 2.4 User Interaction

A sMAP configuration file describes the components of a building; BUSICO 3D automaps the various devices therein to their correct locations. A user has two view options. She can observe each floor from a birds-eye-perspective, or she can switch to a first person view and walk through the building. Actuation can be done in two ways, either in pure simulation in which the drivers act as virtual drivers that include a thermal model and a model for power consumption, or it can be done by actually controlling the physical building. In a real scenario, a building manager first tries out different schedules and rules, runs the simulation and optimizes it over time. When she is sure about the result, the building configuration is applied to the real building. At any point, it is possible to walk through the building to have a visualization of different aspects (e.g light, heating, energy consumption). When discovering something to change there is no need to switch to a different view, the user can just naturally switch devices or set new setpoints like she would in a real building.

## 3 Conclusion and Future Work

BUSICO 3D provides a natural interface for building control and simulation that allows non-experts to interact with a building and configure it. Using a game engine for implementation allowed fast prototyping. We are currently exploring how BUSICO 3D can be tailored for various types of what-if analysis, and more generally as a means of experimenting with new forms of building operations. We are also designing serious games to gain insight on the social practices in a given building and the interactions that define how the building is actually used. We want to implement a more natural experience to the building by using a virtual-reality headset like Oculus Rift. On the simulation side we are currently investigating integration with a more sophisticated simulation engine like Energy Plus.

## 4 References

- [1] S. Dawson-Haggerty, X. Jiang, G. Tolle, J. Ortiz, and D. Culler. sMAP: A Simple Measurement and Actuation Profile for Physical Information. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, SenSys '10*, pages 197–210, New York, NY, USA, 2010. ACM.
- [2] C. Dixon, R. Mahajan, S. Agarwal, A. B. Brush, B. Lee, S. Saroiu, and P. Bahl. An Operating System for the Home. In *NSDI*, volume 12, pages 337–352, 2012.
- [3] G. Wang, M. V. Salles, B. Sowell, X. Wang, T. Cao, A. Demers, J. Gehrke, and W. White. Behavioral Simulations in MapReduce. *Proc. VLDB Endow.*, 3(1-2):952–963, Sept. 2010.