Composability Modeling for the Use Case of Demand-controlled Ventilation and Heating System

Ali Behravan, Nadra Tabassam, Osama Al-Najjar, Roman Obermaisser
University of Siegen, Siegen, Germany
Email: ali.behravan@uni-siegen.de

***Abstract*—In recent engineering practice, the complexity of the systems is increasing. This complexity will keep on increasing if new components are added into the system or system configuration is changed. Numerous configurations of these components are repeated or their relationship is recognizable. On the other hand, simulation tools are commonly used to track the behavior of the modeled systems over time because of their advantages against experimental setups which are the abstraction of reality. However, the ability to integrate different components at different levels of the designed model along with the different techniques used for their inter-relationships based on users requirement is a challenge. The main contribution of this paper is to give an example of how to apply composability to handle this issue. One tangible example is to study the behavior of heating, ventilation, and air-conditioning systems in large buildings. Therefore, finding a solution to model this kind of complex system in a highly composable and scalable view is promising. Composability is a system design challenge that describes how different components can be selected and combined in different configurations and different levels to satisfy users requirement with a highly reduced development cost and time in the simulation, as its advantages. Finally, the practical steps for design and implementation of a composable demand-controlled ventilation and heating system as a useful and energy-efficient smart building's technology in MATLAB/Simulink (besides its constraints) is provided.**

***Index Terms*—Composability, Composable Model, Modeling and Simulation, HVAC, Simulink**

I. Introduction

The main goal of building's heating, ventilation, and air conditioning system (HVAC) is to provide the people with more comfort and also reduce energy consumption. Recent research in this domain shows that advanced Building Energy Management Systems (BEMS) such as demand-controlled ventilation system can reduce the consumption of energy up to 30% [1].

In order to optimize the building automation system, it is important to understand the overall behavior of the system in detail. The behavior of these systems can be studied in detail by analyzing the system model that has been used to create such a system. Lapusan et al. have designed a multi-room automation system using Simscape library from Matlab/Simulink. This system is based on 3R-2C network (3 resistors and 2 capacitors) [2]. Behravan et al. created a model of thermal dynamic that simulates a heating system for a multi-zone office building which is equipped with demand-controlled ventilation (DCV) in MATLAB/Simulink [3]. In that study, which is used as the main reference of this study, a model including all the thermodynamics and heat transfers between different rooms in both directions of horizontal and vertical was simulated.

The vision behind of this study is to create a composable model of the building equipped with DCV and heating system which has the capability of extension in vertical and horizontal directions to test a generic fault detection and diagnosis method that works for this kind of building's extension.

Composability is the ability to select and assemble components in various combinations (in an order) so that specific user requirements can be fulfilled [4]. It is a vital quality of the Modeling and Simulation discipline which is difficult to achieve [5], [6]. The overall process is difficult because the characteristics of individual components are different from the behavior of components in the combined form. In order to implement flexibility during the design of system design, composability is studied as a configuration problem by authors in [7]. The quality of the underlying system is completely dependent on the composition of its subsystems and their quality. In order to fulfill the objective of fast application and adequate development of simulation model resources on the platform of cloud simulation, a framework based on hierarchical modeling is proposed by authors in [8]. The correctness of the proposed hierarchical simulation model is verified by simple examples. Different case studies in this context show that the key technologies of composable modeling can productively enhance the resource modeling simulation over the platform of cloud simulation.

The objective of this work is to give an example of how to apply composability to handle the issue of the ability to combine and recombine different components in different model levels and in different ways considering inter-relationships based on the user requirement. This study focuses on the method in MATLAB/Simulink to generate a composable version of that DCV and heating system model of a smart building with six number of rooms and one corridor in one floor which as the reference [3]. In other words, the goal is to create a novel method for automatic generation of an extended building model with N number of rooms and K number of floors (instead of only 6 number of rooms and 1 floor) from the original one available in this reference. Also, the constraints of the developed model are discussed. The novelty of this work resides in the programming a new $.m$ (MATLAB) file which connects different Simulink system blocks together based on the user requirement for desired architecture by a new graphical user interface and the result will be automatically established in Simulink. Therefore the challenge of this work is not only limited to the Simulink environment but also has a focus on the inter-relationship of MATLAB programming and Simulink. The other possibility of this work is the fault injection in the automatically developed composable model based on a reference of previous work [9] and is another part of our vision discussed earlier in this section. The remainder of this study is organized as follows: Section II introduces the composability and section III describes composable model generation. In Section IV, composable model implementation in MATLAB/Simulink is discussed.

II. Composability

Modeling and simulation are effective solutions from cost, time, and risk points of view to design and to monitor the system and to test the performance instead of experimental setups [10]. Composability is a system design challenge that describes how different components can be selected and combined in different configurations and different levels that satisfies user requirement with a highly reduced development cost and time in the simulation. It is beneficiary to design composable models in an automated way to be manipulated for a larger use case. On the other hand, scalability should be considered when designing the model, scalability can be defined as the characteristic of a model which describes its ability to scale while maintaining an acceptable level of performance and efficiency when working in a larger operational environment [11], [12]. The state of the art in composable modeling today stems from the advancement of the Modeling and simulation sciences and the rapid growth in simulations tools such as MATLAB/Simulink or LabVIEW. Currently, users only identify their model requirements and then the developer constructs the model usually with an iterative process with functionally being added at each iteration, and later the model is delivered with an instruction manual on how to use and maintain [13]. As the complexity of systems is on the rise, the main difficulties and problems in system analysis that affect model composability are stemming from these aspects [14].

1) Integration of models in different disciplines and fields. 2) Abstraction of the model description. 3) Restrictions and constraints of the development tools. 4) Level of composability of the model components.

Figure 1 is an example of a composable model, the original model is designed to be a composition of multiple components, those components are the main components of interests that are needed to be manipulated further to create a new variety of the original model. The new model can be a new iteration of the original model where components are added or deleted like model A or manipulated by adding duplicates of the same components as model B or by modifying the same components internally (i.e. changing the component gain, math relationship or functionality) to create a new system like model C.

III. Composable Model Generation

In this section, the prerequisites for the implementation of a composable model are discussed.

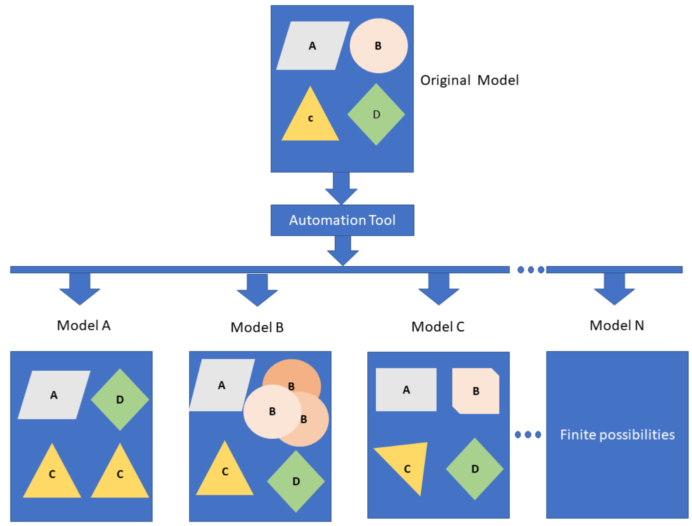

Fig. 1. Example of a Composable Model.

A. System Components Inter-relationships Identification

The first step for a composable model generation in this study is to identify the system components inter-relationships for the original model which is based on a demand-controlled ventilation and heating system model for an office building that has six rooms, one corridor, and one stairway [3], [9]. For this purpose, the authors examined the interconnection between rooms and the other existing components. These interconnections are strongly correlated by the thermodynamic interaction between the different components of model. The model is created in the hierarchical order which means that it is composed of multiple sub-models and this configuration is based on the relationship between sub-models' inputs and outputs data. Figure 2 shows this hierarchical view of composable model at different levels of abstraction. The main three upper layers are part of the high-level model and the bottom layer is the low-level model of abstraction. Simulink is the tool that connects different Simscape library blocks e.g. thermal convection block, thermal conduction, etc. together to create model subsystems e.g. heater subsystem, thermal subsystem, damper subsystem, etc. But to create a composable building model, the MATLAB code which is created by authors in this study is required. The hierarchical nature helps to create a composable model in a straightforward manner.

B. Possibilities and Limitations in Composability Modeling

The integrity of the model during the creation of a new configuration for a composable model must be maintained. After system components inter-relationships identification based on mathematical and physical perspectives, analyzing constraints or limitation besides existing possibilities is very important. Therefore, the authors identified three main types of rooms (with different colors of blue, red, and green in figure 4) and one type of expandable corridor based on the existed symmetry in this study. Mathematical and physical specifications especially heat and mass transfer make these rooms' type

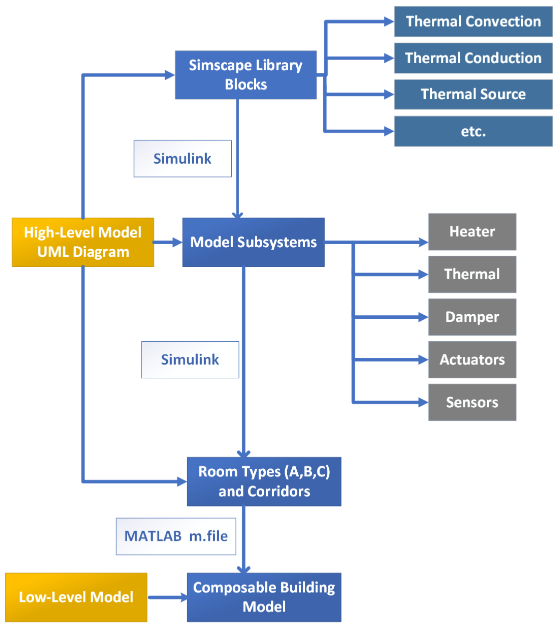

Fig. 2. Hierarchical View of Composable Model.

unique. Figure 3 describes these interactions between rooms and environment. This means they depend on the desired architecture of the target composable model, these types of rooms can be extended or contracted. Figure 4 describes the ability of the implemented composable model in two directions of vertical and horizontal expansion to N number of rooms and K number of floors as the possibility and limitation of this composable model. Table 1 illustrates the room indexing structure which is used to address each specific room in the designed composable model with N number of rooms per floor and K number of floors.

C. *Unified Modeling Language of Composable Model*

Unified Modeling Language (UML) is a standardized modeling language that addresses a wide spectrum of application areas. UML is created and defined using object modeling and consist of many diagram types [15]. A class diagram is one of the types of UML diagrams which describes the functionality

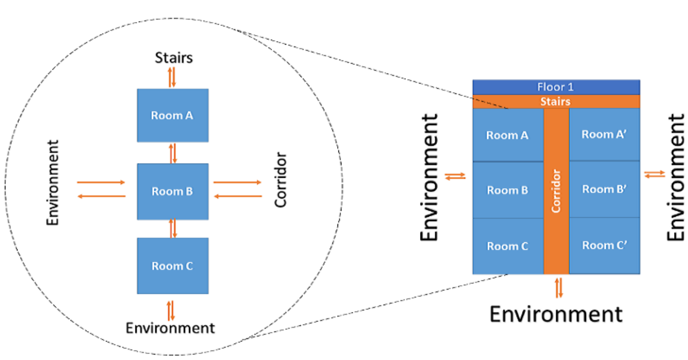

Fig. 3. Interactions among Rooms.

TABLE I
ROOMS INDEXING STRUCTURE.

Floor 1		Floor 2		...	Floor K	
1	(N ÷ 2) +1	(N + 1)	(N+(N ÷2)) +1	...	(N * (K-1)) +1	(N * (K -1)) + (N ÷2) +1
2	(N ÷ 2) +2	(N + 2)	(N+(N÷2)) +2	...	(N * (K -1)) +2	(N * (K -1)) + (N ÷2) +2
3	(N ÷ 2) +3	(N + 3)	(N+(N÷2)) +3	...	(N *(K -1)) +3	(N * (K -1)) + (N ÷2) +3
...	...	...	...	...	...	...
N÷2	N	N + (N ÷ 2)	(N x 2)	...	(N * (K -1)) + (N ÷2)	(N * K)
N: The total number of rooms per floor, K: The total number of floors in the model						

and interaction between the modeled components. UML illustrates the attributes, operations, relationships and associations which are present in the proposed model of the system. The goal of the proposed model is to implement the scalable heating control system equipped with demand control ventilation for smart building management. Therefore, to achieve the complete insight of the accurate building model, a class diagram is designed. So that all the technical requirements and synergy between different systems components can be easily understood by the end designer. The main parts are:

a) Building: The building is the major input class which provides the user with the functionality of creating a number of rooms and corridors in different floors at run-time. Each room class contains the attributes including room number and floor on which room is located. Each corridor has the attributes including a number of rooms and floor numbers. The user can add three types of rooms named as Type A, Type B, and Type C. Along with that user can also add custom corridor block. The other classes are as follows:

b) DCV and Heating System: This class presents the information about the DCV and heating system and is a compositional part of a building class. DCV and heating system class has two more classes which have the compositional links with it. These two classes are process interface and ventilation and heating control system.

c) Process interface: This class represents the processing modules of the DCV and heating system. It has a compositional link with two more classes known as actuators and sensors.

- Actuators: Actuator class inherits its properties to two types of actuator classes known as heater actuator and damper actuator. This class includes heater actuator (Thermostat) which is associated with the heater class of ventilation and heating control system and damper actuator which is associated with the damper class of ventilation and heating control system.
- Sensors: Sensor class inherits its properties to two types of sensor classes known as CO_2 sensor and temperature sensor. CO_2 sensors sense the amount of CO_2 present in the room. The temperature sensor is a class which gives an input value of room and corridor temperature to the thermal class. Whereas thermal class manages to calculate the heat amount within the room. So temperature sensor

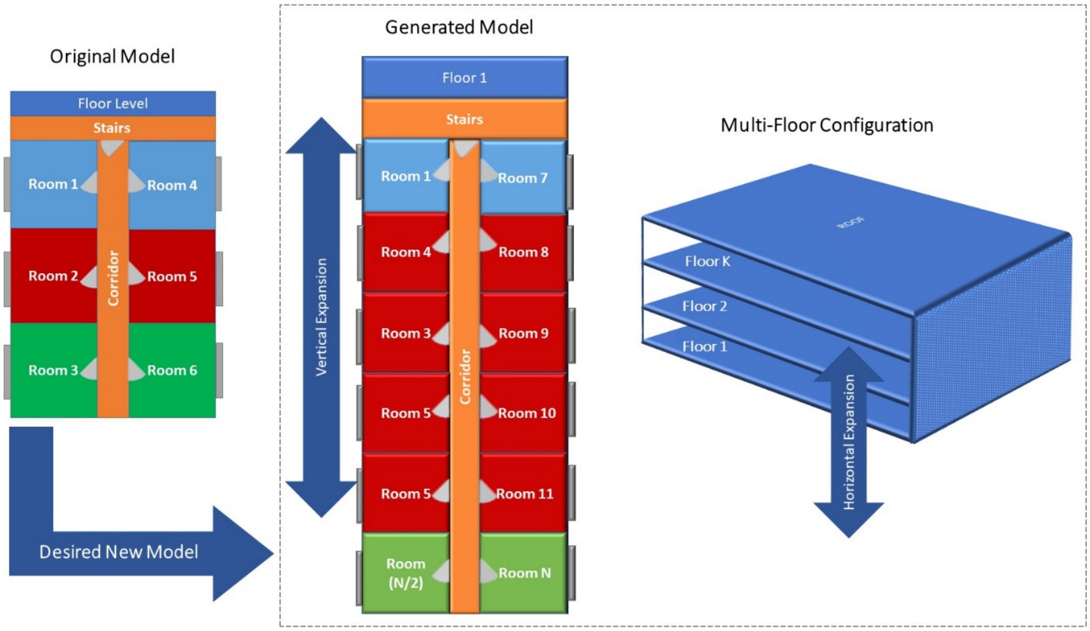

Fig. 4. Interactions among Rooms.

class has an association with the thermal class.

d) Ventilation and Heating Control System: For ventilation control system our DCV and heating system has three compositional classes known as heater, thermal and damper classes.

- Heater: Heater class deals with the amount of heat generates within the room.
- Thermal: Thermal class gets input temperature from the outside environment and adjacent rooms and calculate the temperature of a particular zone which is required.
- Damper: Damper is the class which deals with the air flow calculations of the ventilation system. It has the attribute of CO_2 concentration which actually deals with the amount of CO_2 in the room. Overall CO_2 control signals give the value of CO_2 within the room.

Figure 5 shows the UML diagram respective to the designed composable demand-controlled ventilation and heating system model.

IV. Composable Model Implementation in MATLAB/Simulink

In this section, a complete step-by-step description of composability implementation in MATLAB/Simulink is provided and validated.

A. Composable Model Implementation

The code that was written by authors to generate a composable model consists of 10 main sections written in MATLAB language. Each section is mentioned and described briefly below:

a) System & Variables Initialization: In this section, the model name is given and initialized, also the in variables (i.e. index variables) are initialized and set to zero.

b) User Parameters: This section is made for the user input, where the user can enter the desired number of rooms in each floor based on the section III.B.

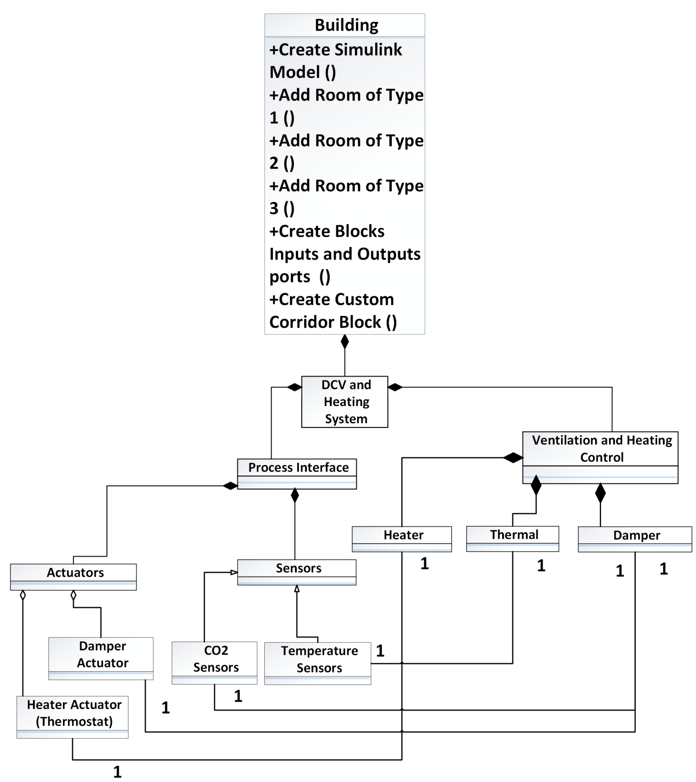

Fig. 5. The UML Class Diagram of Designed Composable Model.

c) System Inputs: In this section, the system inputs are imported from a predefined library, which resides in the same folder of the project. These inputs are the outside temperature block (sine wave Simulink block), the upper floor temperature constant which acts as the roof temperature in the model and finally a stairway temperature constant which is connected to the office zones through an input port labeled $Stairs$. The code will import these blocks and place them on the created Simulink page using the $add\_block()$ command.

d) Rooms Aligning Logic: As discussed in section III.B, the established composable model in this study is scalable and the expansion in horizontal and vertical directions considers the thermodynamics of modeling of the room blocks which enables the room block to have heat transfer with its surrounding in both directions [3]. If any other building architecture was desired by the user, a new modification for the codes is necessary.

e) Main Components & Inter-Connections: After the creation of the desired building structure, the main model components of section III.C are added to the model in this section and the inter-connection among them are established. Also, an output index is assigned to each output port of each model component that is used to capture data for monitoring and diagnostic.

f) Corridor Block Generator: This section of the code is responsible for a customized corridor block for the composable model which suits to the new model. The new customized corridor block includes all the scaled or expanded mathematical and physical interactions among different rooms with the composable corridor block.

g) Assembling of the Sub-Functions: This section of the code deals with arranging the blocks in the simulation environment and layouts.

h) Adding Monitoring Scopes and Labels: Preparing suitable and flexible scope blocks that can automatically adapt to the new system design was another challenge in this study. This means the designed model must be able to show the behavior of the system in form of output signals from sensors and actuators in every scales. The authors solved this challenge therefore based on the desired system architecture, the suitable output figures will be available to the user for the aims of monitoring and diagnostic.

i) Graphical User Interface (GUI): Graphical user interface eliminates the need to use programming languages in order to execute a command in an application [16]. The GUIs help the user to navigate and control the generated model easily by setting the parameters in a graphical and simple view. MATLAB offers the ability to design and run a GUI for various range of applications, the GUI tool can be accessed by typing $> GUIDE$ in the MATLAB command window. This tool will allow users to drag and drop elements to the users GUI window or alternatively the user can add each component programmatically using the corresponding command for each desired element in the GUI i.e. for a push button the user has to use the following command in his/her code and the button will be created each time the user compiles this script [17]. The user can also control the GUI elements behavior using function callbacks later. The designed GUI for the present study includes the following main parts:

- In component and fault parameters selector section the user can specify Room or Corridor and the type of the component that is desired for monitoring or fault injection. As discussed, this model can be used to study the system behavior (such as temperature signal) in case of different types of faults. Also, the user can specify the component index, Fault value, and the delay which is representative for the time of fault injection.
- In the Fault type section, the user can pick the wanted fault type and fault category, i.e. $Heater\_Actuator\ FI$ from the first fault category: $Hardware\ Fault$.
- In the Monitoring and Scopes section, the user can view the system behavior signals in the desired room by simply entering the room index.
- In the Fault Signal Controller (Temperature Sensor & Co2 Sensor) the user can apply a fault signal that has a mean and variance to the Temperature and CO2 Sensor value.
- In the Simulation Controls section, the user can specify the intended duration of simulation, also there is an ability to start/stop/pause the simulation.

Figure 6 describes the designed GUI.

B. Validation

To make sure that the simulation results of the composable model are reliable, the results of the established composable model for different parameters signal such as temperature of different zones were compared with the original model that is available in reference [3]. For the validation process of the recent model, the number of rooms was selected six and the number of the floor was selected one to create an exactly the same building with the reference [3]. Figure 7 certifies a complete match between the captured data for temperature signal from the reference model and the recently developed composable model for example room number one.

Conclusion

This paper describes a step-by-step composability modeling of the heating and demand-controlled ventilation system approach for smart buildings whose implementation has been done successfully in MATLAB/Simulink. The reason of this study is to develop a suitable composable model for the huge buildings for system behavior monitoring in building management systems and to study and test the diagnostic techniques in further studies. Also, a graphical user interface for this purpose is developed by authors to create a composable building model with its unique specifications considering the users requirements.

Acknowledgment

This work was supported by the DFG research grant OB384/5-1.

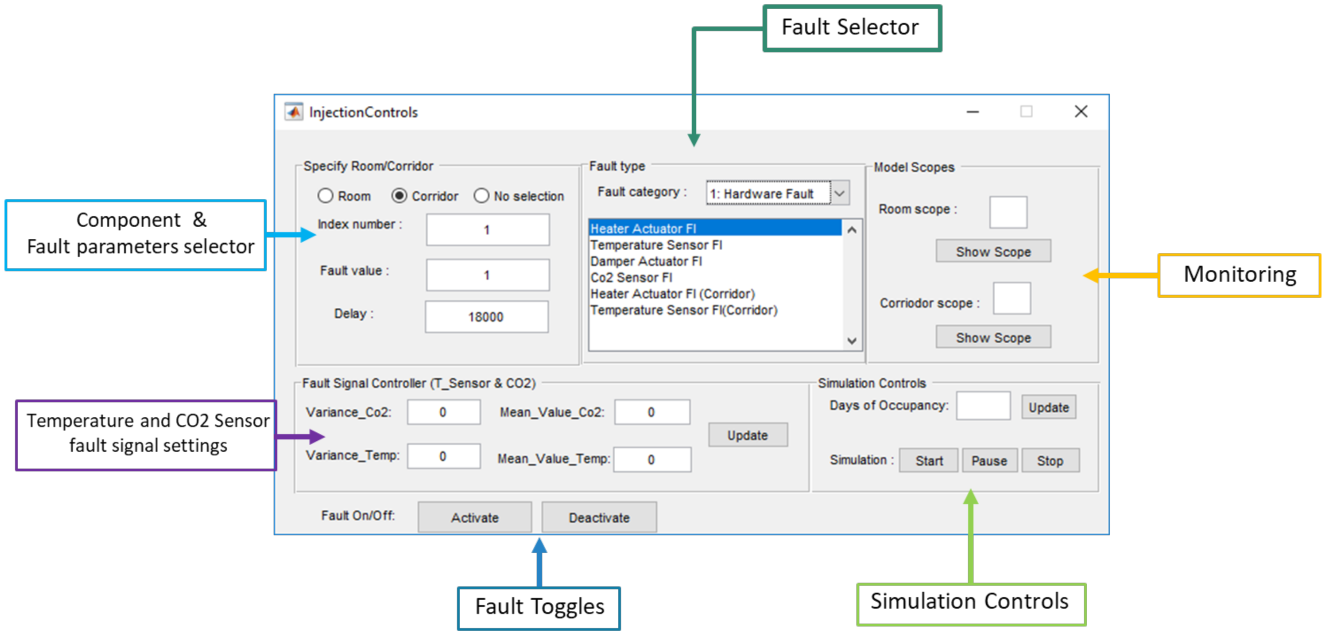

Fig. 6. The Designed GUI.

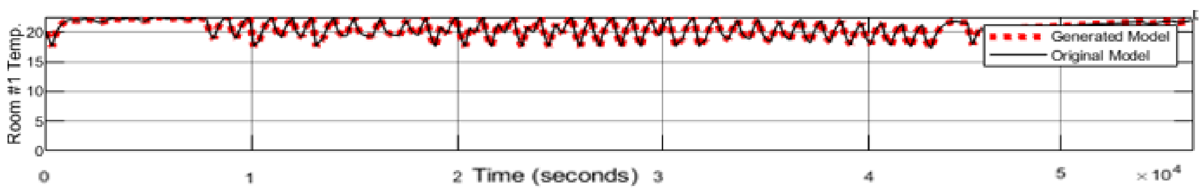

Fig. 7. Model Validation Result.

REFERENCES

[1] P. Shaikh, N. Nor, P. Nallagownden, I. Elamvazuthi, T. Ibrahim, A review on optimized control systems for building energy and comfort management of smart sustainable buildings, Journal renewable and sustainable energy reviews, Vol. 31, 2014, pp. 409-429.

[2] C. Lapusan, R. Balan, O. Hancu, A. Plesa, Development of a multiroom building thermodynamic model using Simscape library, EENVIRO-YRC, November 2015, Bucharest, Romania.

[3] A. Behravan, R. Obermaisser, A. Nasari. "Thermal dynamic modeling and simulation of a heating system for a multi-zone office building equipped with demand controlled ventilation using MATLAB/Simulink." In Circuits, System and Simulation (ICCSS), 2017 International Conference on, pp. 103-108. IEEE, 2017.

[4] M. D. Petty, E. W. Eric, "A theory of simulation composability," Virginia Modeling Analysis & Simulation Center, Old Dominion University, Norfolk, Virginia, 2004.

[5] O. Balci, J. D. Arthur, W. F. Ormsby, "Achieving reusability and composability with a simulation conceptual model," Taylor & Francis Journal of Simulatio, Vol. 5, Number 3, pp. 157-165, 2011.

[6] K. Davis Paul, H. Anderson Robert, "Improving the composability of department of defense models and simulations," Santa Monica CA: RAND Corporation, 2003.

[7] J. F. Briones, M. de Miguel, J. P. Silvana, A. Alonso, "On the requirements for quality composability modeling and analysis," 13th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORCW), pp. 123-129, 2010.

[8] D. Zhu, P. Liu, Q. Su, "Research on Hierarchical Modeling of Simulation Model of Cloud Simulation Platform," IEEE International Conference on Virtual Reality and Visualization (ICVRV), pp. 455-458, 2014.

[9] A. Behravan, A. Mallak, R. Obermaisser, D. H. Basavegowda, C. Weber, and M. Fathi. "Fault injection framework for fault diagnosis based on machine learning in heating and demand-controlled ventilation systems." In 2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI), pp. 0273-0279. IEEE, 2017.

[10] A. Behravan, R. Obermaisser, M. Abboush, Fault Injection Framework for Demand-Controlled Ventilation and Heating Systems Based on Wireless Sensor and Actuator Networks, The 9th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON 2018) "BEST PAPER AWARD", University of British Columbia, Vancouver, Canada, 2018.

[11] Bondi, Andr B. (2000). Characteristics of scalability and their impact on performance. Proceedings of the second international workshop on Software and performance WOSP '00. p. 195.

[12] Hill, Mark D. (1990). "What is scalability?". ACM SIGARCH Computer Architecture News.

[13] S. Kasputis, C. Ng. Henry, "COMPOSABLE SIMULATIONS," Proceedings of the Winter Simulation Conference, 2000.

[14] Yifan Zhu, Weiping Wang, and Dongxiang Zhou,"Conceptual Framework of Composable Simulation Using Multilevel Model Specification for Complex Systems," Asia Simulation Conference, 2008.

[15] C. Kobryn, Will UML 2.0 Be Agile or Awkward?, COMMUNICATIONS OF THE ACM, Vol. 45, January 2002.

[16] Create Apps with Graphical User Interfaces in MATLAB, MathWcorks. Retrieved October 19, 2018.

[17] Create a Simple App Programmatically, MathWcorks. Retrieved October 19, 2018.