

Article

AppSimV: A Cyber–Physical Simulation and Verification Platform for Software Applications of Intelligent Buildings

Haining Jia ¹, Qiliang Yang ^{1,*} , Ziyang Jiang ², Wenjie Chen ¹  and Qizhen Zhou ¹ 

¹ College of Defense Engineering, Army Engineering University of PLA, Nanjing 210000, China; 15005169961@163.com (H.J.); cwj13505152123@163.com (W.C.); zhouqizhen2016@163.com (Q.Z.)

² Building Energy Research Center, Tsinghua University, Beijing 100084, China; jiangzy@tsinghua.edu.cn

* Correspondence: yql@893.com.cn

Abstract: Testing and verifying applications (Apps) are essential for a software-driven intelligent building system. Traditional methods connect App programs to hardware devices for debugging and testing on the engineering site. However, App bugs can hardly be found out before they are being deployed and thus always require an extended debugging cycle. To address this issue, we propose a cyber–physical simulation and verification platform named AppSimV, which enables the testing and verification of Apps in a mimic real scene. Taking swarm intelligence building as an example, this paper focuses on the cyber–physical architecture of AppSimV and its implementation mechanisms, including the standardized encapsulation of software components for the building physics model, a multitask scheduling simulation engine, a cyber–physical interaction interface, and the visual monitoring of the simulation process. The implementation mechanisms not only accurately simulate actual engineering scenarios but also facilitate the early detection and correction of issues that may arise during the App’s runtime, thus reducing the debugging time required for the App. With 1200 intelligent physical nodes connected in a swarm hardware system, AppSimV was validated by conducting the strict testing and verification of a set of Apps for an intelligent building. The results show that AppSimV is sound and reliable.

Keywords: intelligent building applications; real-time simulation and verification; cyber–physical simulation platform



Citation: Jia, H.; Yang, Q.; Jiang, Z.; Chen, W.; Zhou, Q. AppSimV: A Cyber–Physical Simulation and Verification Platform for Software Applications of Intelligent Buildings. *Buildings* **2023**, *13*, 2404. <https://doi.org/10.3390/buildings13102404>

Academic Editors: Yu Huang, Siwei Lou and Yukai Zou

Received: 16 August 2023

Revised: 18 September 2023

Accepted: 19 September 2023

Published: 22 September 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the continuous improvement of information technology, intelligent buildings have become the most important part of smart city construction, which provides important support for the green and low-carbon operation of buildings [1–7]. In recent years, intelligent buildings have gradually become software-intensive and software-defined [8]. Specifically, swarm intelligent building systems [9–14] that function based on Apps are typically representative of this type of technology, which attracts great concerns in the industry. In such cases, effectively testing and verifying the App is critical to guarantee its efficient deployment and the successful operation and maintenance of intelligent buildings. However, at the current stage, the development and debugging of intelligent building applications are generally performed at project sites [15,16] and affected by issues such as a long debugging cycle, the late discovery of problems, and the lack of systematic support. Some scholars have attempted to explore the verification of key algorithms used in intelligent building Apps in the form of simulation platforms before deployment to the project site [17–25]. However, the operating environment of application algorithms is generally a simulation software (such as MATLAB), and the hardware computing environment of an actual project site can hardly be appropriately considered [26–30]. Therefore, the effectiveness and reliability of simulation verification may easily be affected. Therefore, how to build intelligent building application simulations and verification platforms that mimic the real scene of an actual project is important.

With the swarm intelligent building system as an example, an open and scalable intelligent building application simulation and verification platform, named AppSimV, is proposed based on the integration of cyber–physical technology. In AppSimV, the intelligent building application runs with a real hardware controller (physical system), and the control objects in the App (such as the building chiller, water pump, etc.) exist as virtual simulation entities. Interactions among all objects and entities are considered in both directions through a physical network system. Additionally, a virtual entity encapsulation method for building unit objects is proposed to realize the dynamic expansion of the building simulation entity library in the platform. Thus, a new intelligent building application simulation and verification platform that can simulate the real engineering environment is formed. The main innovations of this work are summarized as follows:

- (1) An AppSimV architecture that integrates cyber–physical ideas is established. In this hardware-in-loop architecture, Apps run on real hardware controllers (physical systems) and interact with virtual simulation models of building units (cyber systems), which provide a platform that is close to the reality scenarios of buildings.
- (2) The universal model for real-time simulation of building units and the method of component encapsulation are proposed, which achieve the mirroring, residence, execution, and interaction of building elements and operational processes to the information world.
- (3) The AppSimV foundation software system is implemented. It includes a kernel engine, virtual real interaction middleware, and simulation process visualization monitoring software, which form a full scene real-time simulation support environment for testing and verifying the Apps of buildings.
- (4) A systematic test and an evaluation experiment for the AppSimV platform are conducted, including software testing and multi-application experiments. The experimental results show the effectiveness and availability of AppSimV.

The rest of this paper is organized as follows. Section 2 introduces background and related work. In Section 3, the framework of AppSimV is given. Section 4 introduces the key implementation mechanisms of AppSimV. In Section 5, the availability and reliability of AppSimV is verified. Finally, Section 6 concludes the paper.

2. Background and Related Work

2.1. App-Based Swarm Intelligent Building System

The swarm intelligent building system is proposed based on the spatial distribution of buildings. The buildings and their electromechanical systems in the building space are regarded as a decentralized network composed of building space units and large electromechanical equipment, which is [9–14,31–33] shown in Figure 1. Each building space unit and electromechanical equipment corresponds to a computing process node (CPN) [34–36]. App developers who are permitted or under granted can upload the developed swarm intelligent building application to the application store. The intelligent building system downloads an App from the App store to a CPN during construction debugging, and the CPN manages and controls the building through the running App, as shown in Figure 2. This process forms a new software-defined architecture. In the App-based swarm intelligent building systems, Apps play a direct role in the operation, management, and control of the swarm intelligent building, providing strategies and control algorithms for the operation of the intelligent building. However, in the open and multi-source software development model, in the actual engineering application stage, developers often find it difficult to enter the engineering site in advance, causing the App to encounter difficulties and challenges during development, testing, validation, and other aspects. Therefore, conducting rigorous testing and validating swarm intelligent building applications in near-real engineering scenarios before deploying them to actual engineering sites is very important.

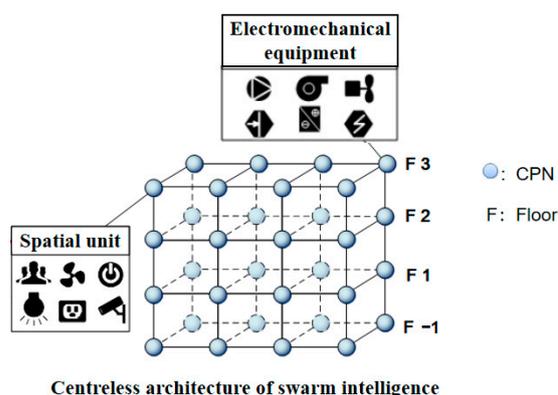


Figure 1. Schematic diagram of the swarm intelligent building system architecture.

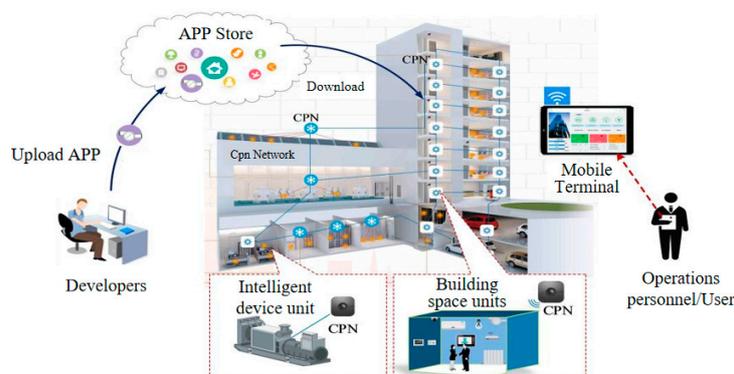


Figure 2. Development and deployment process of swarm intelligent building application.

2.2. Related Work

Research on and the construction of simulation platforms for buildings and the corresponding facilities and equipment are important for scholars around the world in the field of intelligent buildings. With the continuous development of building simulation technology, such technology has been widely used in building environments and other related fields. Researchers have mainly focused on the following two aspects of building simulation technology.

(1) Simulation software for the intelligent building design stage

An App in the architectural design stage, such as for the calculation of the building cooling/heating load, analyzing building energy consumption, and performing an economic analysis of the building, enables the designer to have a clear understanding of the economic cost of the design scheme. At present, the mature software simulation platforms mainly include TRNSYS [37] and HVACSIM [38] in the United States. The DeST platform series developed by academician Jiang Yi's team [39] is the most influential platform in China. The latest version of DeST3.0 (Yan et al., 2023) provides an efficient tool and comprehensive simulation platform for the analysis and optimization of building performance throughout the building life cycle [40]. Zhao et al. proposed a new three-dimensional simulation platform for prefabricated buildings based on BIM [41], which effectively solved the problems of low accuracy and poor analysis ability for traditional prefabricated building structures. Huang et al. [42] studied a crowd evacuation agent simulation platform considering the influence of building fire-related factors, identified the effects of fire on the evacuation decisions and behaviors of individuals, and established a crowd evacuation simulation platform considering the real-time impact of fire called FREEgress. This platform accurately simulated crowd evacuations during building fire disasters to optimize the performance-based fire protection design of buildings.

(2) Semi-physical building simulation platform

In recent years, the construction of building simulation systems based on semi-physical simulation has attracted increasing attention, and the control objects in building automation systems have been effectively simulated. Tao et al. [28] proposed a hardware-in-the-loop simulation platform based on a building air conditioning systems to optimize loop hardware, achieve an air balance operation state, optimize equipment operation, and reduce energy consumption. Guo developed a virtual simulation platform for wind power generation based on a programmable logic controller (PLC) [29]. The test platform L. Sadraddin, which is based on a low-delay distributed real-time hybrid simulation [30], combines physical testing with numerical simulation to evaluate seismic responses based on actual seismic data, which can meet the needs of modern buildings for complex structural systems.

In summary, simulations in the building field have focused on the design stage, and the simulation systems in the construction operation and maintenance stage are mainly focused on local entities. There has been a lack of simulation research on all the elements of building operation and maintenance, and there has been no research on semi-physical simulation technology and support platforms that can provide universal simulation and testing capabilities for building control and management Apps.

3. Framework of AppSimV

To provide an operation environment close to that in the field for the testing and verification of intelligent building Apps, cyber-physical technology is integrated to build a semi-physical simulation test and verification platform. A building simulation model is built for space units such as building spaces and electromechanical equipment, and it is run as a software virtual entity. The physical CPN hardware system of the App manages the building simulation model through special middleware operations, thus providing a support platform that closely mimics the actual engineering environment for the testing and verification of the Apps. The overall architecture of the AppSimV platform is shown in Figure 3.

AppSimV stores the simulation model data from various building spaces and electromechanical equipment in the simulation model libraries of intelligent building to perform managing, scheduling, execution, and monitoring tasks through a multisimulation scheduling execution engine. The data required in the process of Apps simulation tests and verification are stored, processed, transmitted, and integrated with a swarm intelligent simulation operation database, and the progress of specific simulation steps is monitored by the simulation data monitor. The CPN system running the intelligent building Apps communicates with the intelligent building simulation software through middleware. The multithreading scheduling of the simulation platform of the whole cyber-physical system in the simulation process is realized by the swarm intelligence simulation thread scheduling center module, and visualization and human-computer interactions are realized by the simulation display module. The specific functions of each component module are as follows.

(1) Simulation model library of swarm intelligent building

The library stores the relevant parameters, physical quantities, operation modes, and states of the building space model, and the model is encapsulated by a unified software component (DLL).

(2) Swarm intelligent building model execution engine

The intelligent building simulation model for management, scheduling, execution, and monitoring is the core of the simulation platform. This engine is composed of a model-parsing module and a modeling time module.

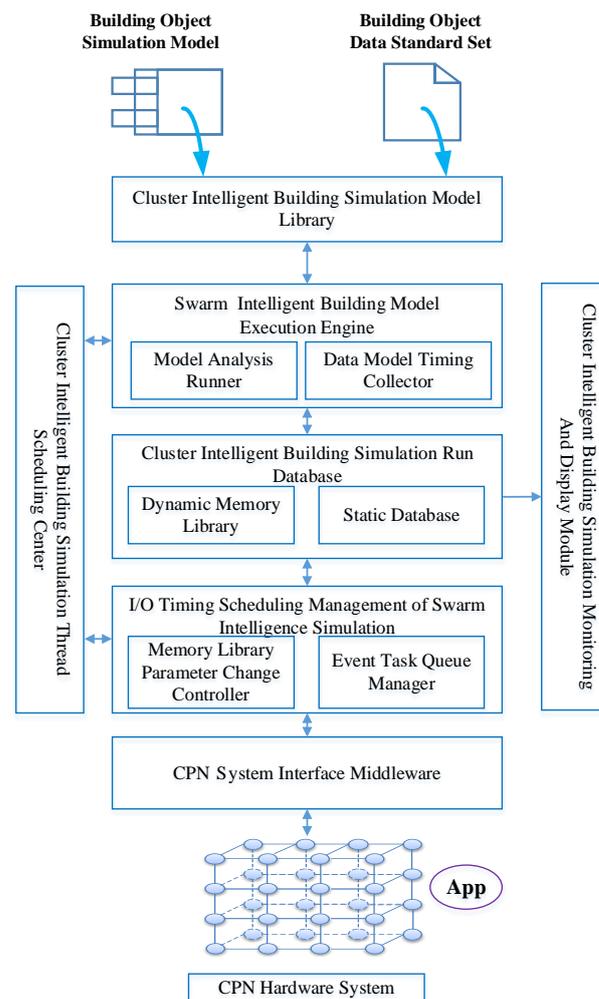


Figure 3. Overall architecture of AppSimV.

The former module is used to manage the execution process, interface, and memory of the simulation model for swarm intelligent building and is connected to the write parameter interface and callback interface of the simulation model. At runtime, the definitions of model parameters, interfaces, memory, permissions, and association relationships are established through a configuration process to initialize the model. Then, data are written to the simulation model through the write interface, and the scheduling simulation model is linked to the callback interface to realize the mutual calling of different model parameters.

The modeling time module is used to store and manage the trigger cycles of different models and is connected to the parameter reading interface of the swarm intelligent building simulation model to realize the starting and stopping of different submodels, record the modeling time, store data, and maintain and update the parameter memory.

(3) Simulation database for swarm intelligent building

The simulation database is used to realize the storage, processing, transmission, and integration of data in the process of App testing and verification and provides support for the basic data layers of the simulation platform. This database contains a memory library and data storage capability.

The memory library provides dynamic data support, dynamically stores the temporary data from the simulation model and CPN system, and supports the reading and writing of data. The database provides static data support, stores user data and simulation data, realizes the integrated management of simulation data, and provides support for the simulation display module.

(4) I/O timing and scheduling management for swarm intelligent building simulation

Swarm intelligent building simulation I/O timing scheduling management mainly involves monitoring the specific operation of the algorithm in the CPN system and providing support for the normal operation of the CPN system. This approach includes a parameter change monitor and an event task queue manager. The parameter change monitor records the simulation model parameters maintained by the modeling time module. When the monitor detects a certain difference, it will transfer this parameter to the CPN system, which performs a data reading operation. The event task queue manager monitors the CPN system, manages the change events output by the CPN system as event task queues, transfers these events individually to the simulation model, and completes the write operation for the simulation model driven by the change events.

(5) CPN system interface middleware

Interface middleware is used to realize data communication between the physical CPN hardware system and the swarm intelligent building simulation software and to establish data exchange between the CPN system and the swarm intelligent building simulation model.

(6) CPN system

The CPN system of the physical hardware controller running the intelligent building application is equivalent to the system deployed at an actual project site. The CPN system reflects the physical architecture of building space units and electromechanical equipment in a swarm intelligent building, and the operating environment of the swarm intelligent building App encompasses each node in the CPN system. The hardware system mainly includes the CPN control network system, the simulation and data storage server, and four App development and simulation workstations (the CPN network management station, App development management station, simulation configuration management station, and simulation process monitoring station).

The CPN network system is responsible for running the App, and its configuration is exactly the same as that at a project site. The simulation model and data storage server run the simulation model database management software to store, manage, and share dynamic simulation data. Among the four App development and simulation workstations, the CPN network management station is used for the debugging, configuration, and management of the CPN network. The App development management station supports the development and management of the App for swarm intelligent building. The simulation configuration management station realizes the operation and configuration of simulation engine software tools. The simulation process monitoring station runs the simulation process monitoring software, transparently monitors the dynamic changes in simulation model parameters in the form of real-time and historical trends, and modifies the simulation parameters to simulate on-site equipment start-up, shutdown, failure, etc.

(7) Clustering module for the intelligent building simulation thread scheduling center

This module provides multithread scheduling support for simulation platform software to ensure the stable and efficient operation of the platform.

(8) Simulation monitoring and display module for swarm intelligent building

The visual presentation of the simulation operation process is achieved, and the human-computer interaction interface is provided to support the intuitive and user-friendly monitoring of the simulation operation process and analyses of results, thus providing a direct tool for assessing the effectiveness of the App. During the simulation operation process, the changes in building parameters and events are recorded and maintained in real time by the simulation display module for the swarm intelligent building, and the more recent data are input into a real-time simulation data module. The real-time simulation data module only maintains one piece of data. As the changes in parameters and events are continuously written, the real-time simulation data module performs immediate data

updates. Then, the data are displayed through a human–machine interface, and values are updated dynamically in real time with the data updating scheme. This approach can be used to manage all the historical simulation data generated previously and display the curves of relevant parameters over time and the operation status of the simulation equipment.

4. Implementation

According to the overall architecture introduced in Section 3, this section focuses on the key implementation technologies of AppSimV, including the packaging method of the physical building simulation model, the AppSimV multisimulation task scheduling execution engine, the physical virtual reality interaction middleware and corresponding data, the simulation process visualization monitoring software, etc., to realize the dynamic management and scheduling of the simulation model, the definition of App simulation tasks, visualization monitoring, and the two-way interactions between the App and CPN hardware system. The specific implementation mechanism is as follows.

4.1. Virtual Encapsulation of Building Simulation Objects

4.1.1. Packaging Mechanism

To enhance the scalability of AppSimV, realize the efficient management of intelligent building simulation models (including building space units, electromechanical equipment units, etc.), and reduce the impacts of model modification, addition and deletion on the simulation platform, a component-based packaging and management method for building a simulation model, is established.

The component-based packaging concept used in this building simulation model is shown in Figure 4.

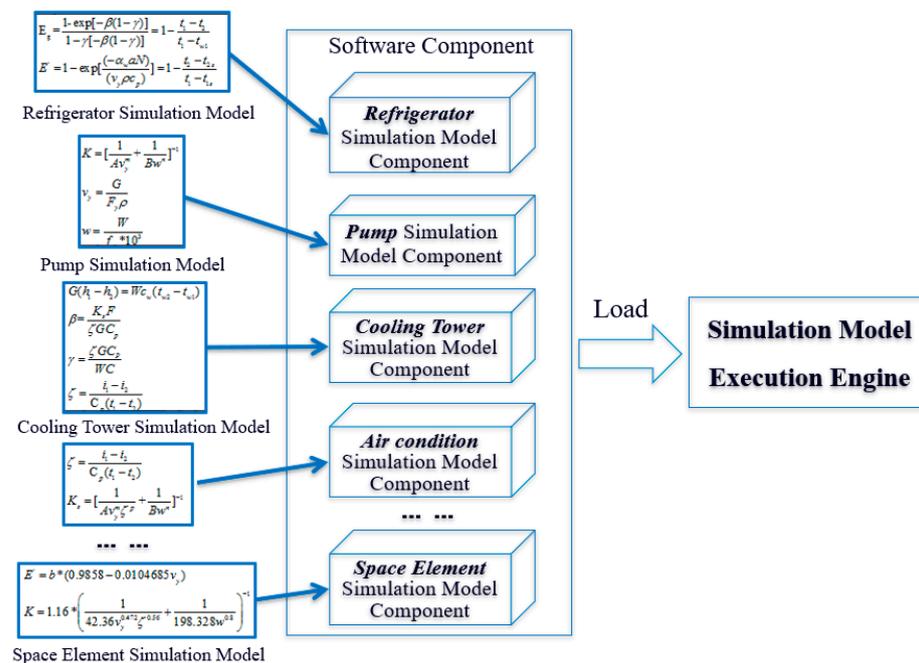


Figure 4. Component-based packaging method for the building simulation model.

Based on typical physical processes and equipment operation mechanisms in a building, a transfer function mathematical model is constructed, and an object simulation model for building space units and typical electromechanical equipment (fans, pumps, refrigeration units, cooling towers, fan coil units, etc.) is established. The model and interface, which includes the read interface, write interface, and callback interface, are constructed and encapsulated in the form of a dynamic link library (DLL; .dll format) so that the com-

ponents of the swarm intelligent building model are relatively independent, replaceable, and deployable, thereby reducing the effects of model modification, addition, and deletion on the simulation platform.

The specific implementation structure of the building simulation model components based on the DLL in the packaging process is shown in Figure 5.

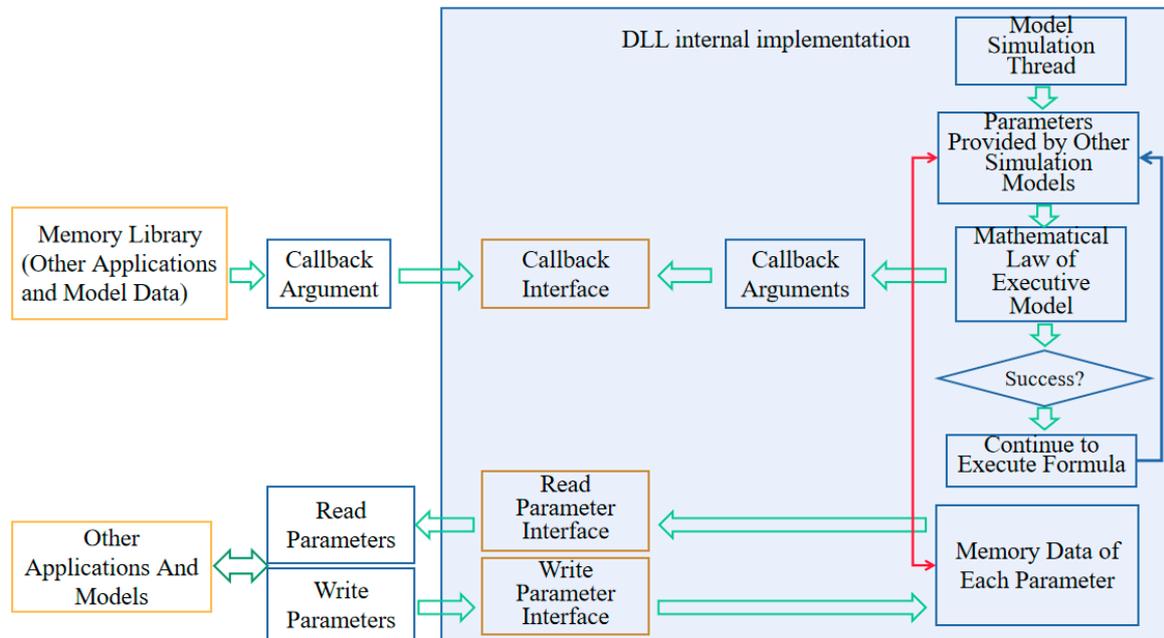


Figure 5. Implementation structure of the building simulation model components.

The mathematical laws of the building simulation model are encapsulated in the DLL and executed through a separate thread. The building simulation model needs to use relevant input and output parameters in the implementation process. Therefore, the simulation model component DLL is associated with three interfaces, namely, the App read parameter interface, App write parameter interface, and callback function interface, to consider the data interactions between the simulation model and App, as well as with other simulation models. The intelligent building App reads model data through the App read parameter interface and writes model data through the App write parameter interface, and the interactions among simulation model components are considered through the callback function interface.

At runtime, the interface is dynamically loaded into the process space where AppSimV runs and is driven by the multisimulation task scheduling execution engine of AppSimV. The simulation model is called and executed by the model analysis operator through time slice polling and interacts with the App in the CPN network through the defined interface. The call cycle of each simulation model is defined by the user.

4.1.2. General Development Template for Simulation Model Components

To provide unified support for the development of building simulation model components at the software implementation level, a general development template for the simulation model components of swarm intelligent building is developed; this template provides a standard and consistent development framework for simulation model developers to ensure the standardization of simulation model components while reducing the development difficulty and improving the development efficiency.

This paper provides a simulation model component development template based on the C++ language and defines four header files, namely, macro.h, fundeclaredef.h, uservar.h, and usersim.h. Users use the four files to achieve unified development and design. The functions of the four header files are as follows.

Macro.h defines the document with a memory database table, and all the macro definitions and user macro definitions involved in all tables are written into the current macro.h files. Fundeclaredef.h is used to write all function declarations in the user algorithm. Uservar.h is used to write the variables used in the user algorithm. Usersim.h is used to write the user algorithm, which mainly includes system functions and user functions. System functions are mainly used for user variable initialization, interface read parameter initialization, interface write parameter initialization, and periodic function calls. The user function stores the functions used by the user algorithm and writes the function declaration into fundeclaredef.h. The functions in the two files need to correspond. Notably, declarations need to be mapped individually.

The relationship among the four header files is presented in Figure A1 of Appendix A.

4.2. Multisimulation Task Scheduling Execution Engine

The multisimulation task scheduling execution engine aims to realize the real-time management and monitoring of the building simulation model. It is composed of basic model management, initial model management, timer management, and event monitoring, as shown in Figure 6. The basic model management component is responsible for the static and dynamic management of the DLL interface and the dynamic management of model memory to support the corresponding arrays of various data types and the reading of vector memory. The initial model management component initializes and personalizes the model through data. The timer management component is responsible for model timing and storage cycle management. The definition of a simulation task is realized through basic model management, initial model management, and timer management. The event monitor calls the interface to complete the reading and writing of events. Afterward, the corresponding operation is initiated in the CPN through the CPN system middleware, and necessary information is outputted.

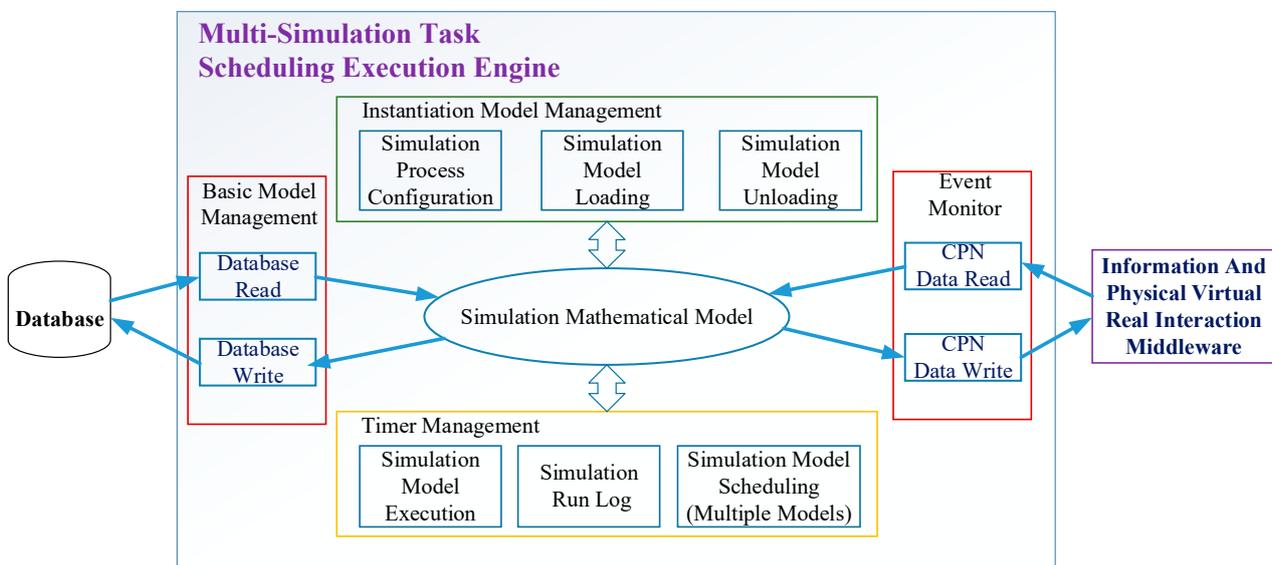


Figure 6. Internal structure of the multisimulation task scheduling execution engine.

The task execution process of the multisimulation task scheduling execution engine is presented in Figure A2 of Appendix A.

The specific algorithm implementation steps are presented in Algorithm 1.

Algorithm 1 Multisimulation Task Scheduling Execution Engine Implementation

- 1: Collect all current simulation tasks to be executed, obtain the task run-list from the event monitor and initialize it, label the task serial number, and prioritize the tasks.
- 2: Merge the tasks according to each task priority to build a mdl group, and build n model-type queues for the same-priority mdl list for each priority with an active trigger cycle.
- 3: If mdl list Size() < 1, the queue is empty, and the execution ends.
- 4: If mdl list Size() ≥ 1, obtain the highest-priority model-type queue pdl list from the mdl group get index for execution.
- 5: Create new threads for multiple tasks in the model-type queue pdl list and add them to the thread pool.
- 6: Execute tasks concurrently, adjust relevant parameters, and return corresponding data through the interface. The completed tasks are deleted from the mdl group, and the list size is adjusted.
- 7: Return to Step 1 and continue to execute the process until all tasks are completed.

4.3. Cyber–Physical Virtual Real Interaction Middleware

In the operation of AppSimV, the data associations among the simulation model, equipment, and physical field are particularly important. Information-based and physical–virtual interaction middleware can be used to support data interactions between the simulation model and physical entities. This middleware receives the real-time data and simulation configuration information generated by the multitask scheduling execution engine through the simulation model communication interface, configures the updated data in the simulation model through the data set management and data exchange execution components, and transmits the updated simulation model to the CPN hardware system through the CPN data exchange interface. The internal structure of this middleware is shown in Figure 7.

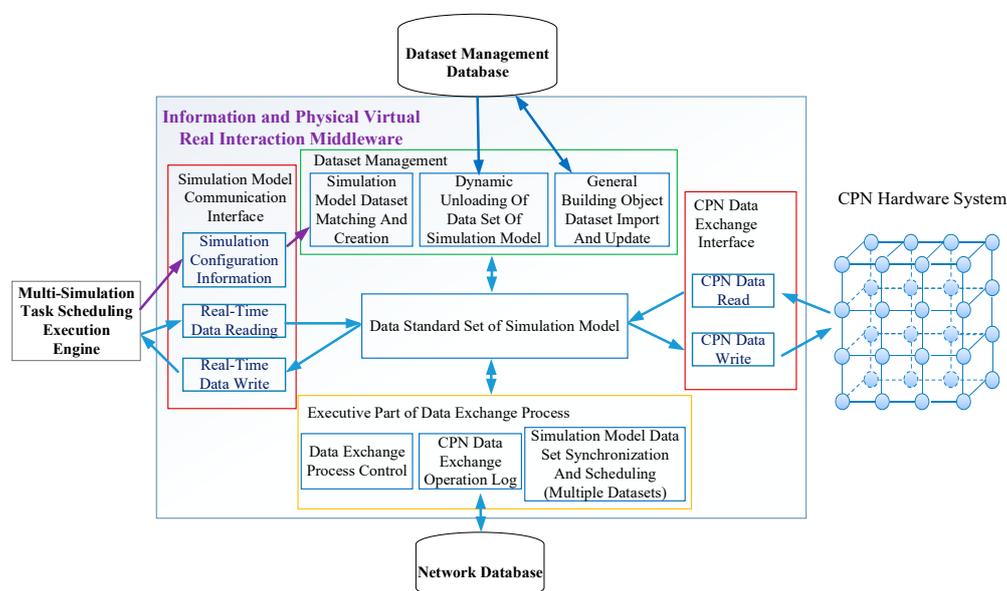


Figure 7. Internal structure of the middleware for cyber–physical–virtual interaction.

The specific operation flow chart of the middleware is presented in Figure A3 of Appendix A.

The specific algorithm implementation steps are presented in Algorithm 2.

Algorithm 2 Cyber-Physical Virtual Real Interaction Middleware Implementation

- 1: Load and call the basic interface, initialize the top interface, and set the data callback and status callback.
- 2: Execute the data callback. The data callback operation builds the varadd list and synchronizes the memory library to trigger data change events.
- 3: Add the change event to the cpnaddr list.
- 4: Set the offline status of the device, update the data according to the list, reconnect, and end the execution.

4.4. Visual Monitoring Module of the Simulation Process

To effectively realize the real-time management and scheduling of the simulation model, the corresponding data should be visually monitored during the simulation process. The visual monitoring module of the simulation process includes two parts: data monitoring and simulation display. Data monitoring is the monitoring of model input, output, and stored data. The simulation display is achieved with a UI. The swarm intelligent building simulation data monitoring component forms an event queue by monitoring the changes in model inputs and outputs and data storage parameters, as shown in Figure 8. The simulation display visually illustrates the parameter and event changes and shows historical simulation data and the real-time simulation data generated by calling the changes in parameters and events with the human-machine interface. The simulation display aids in monitoring the simulation process from a holistic perspective. The structure of the simulation display is shown in Figure 9.

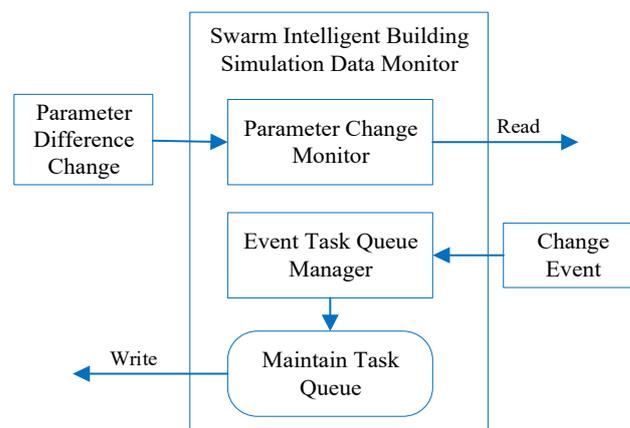


Figure 8. Internal structure block diagram of the simulation data monitor for the swarm intelligent building.

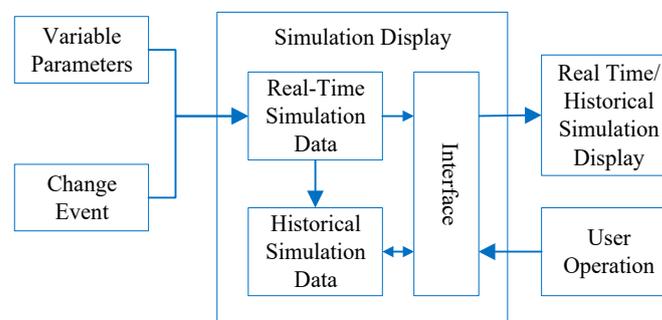


Figure 9. Internal structure block diagram of the swarm intelligent building simulation display.

The specific implementation flowchart of the simulation process visualization monitoring module is presented in Figures A4 and A5 of Appendix A.

The specific algorithm implementation steps are presented in Algorithm 3.

Algorithm 3 Visual Monitoring Module Implementation

- 1: Initialize the memory library (interface write operation list, interface read operation list, callback write operation list, and callback read operation list) and relevant databases (I/O port read operation list and I/O port write operation list).
 - 2: View the current task list, analyze the task execution process and modify the corresponding interface write operation list, interface read operation list, callback write operation list, callback read operation list, I/O port read operation list, and I/O port write operation list.
 - 3: Read the updated memory database and the database content for visualization. If data acquisition fails, proceed to Step 8.
 - 4: If the interface write operation list or callback read operation list > 1 , the model input is valid. In this case, modify the corresponding memory database according to the priority operation list, share it with other modules, delete the executed operation from the corresponding list, and return to Step 2 to continue execution.
 - 5: If the interface read operation list or callback write operation list > 1 , the model output is valid. In this case, modify the corresponding memory database according to the priority operation list, share it with other modules, delete the executed operation from the corresponding list, and return to Step 2 to continue execution.
 - 6: If the I/O port read operation list or I/O port write operation list > 1 , the database has available storage capacity. In this case, modify the corresponding memory database according to the priority operation list, share it with other modules, delete the executed operation from the corresponding list, and return to Step 2 to continue execution.
 - 7: If (all lists == empty), execute Step 8.
 - 8: End execution.
-

5. Experiment

5.1. Experimental Setting

AppSimV simulates the physical architecture of spatial units and electromechanical equipment within swarm intelligent buildings using collected local data and establishes connections with simulation models to simulate the operation of the App and achieve intelligent building control experiments. Additionally, AppSimV facilitates the testing, analysis, and validation of Apps. The test hardware platform uses 1200 CPN nodes, which are physically connected by CPNs through data lines, forming a grid topology. Each CPN has an independent processor, storage unit, six ports that can conduct bidirectional data communication with other adjacent nodes, and several I/O ports that can communicate with peripheral devices. The 1200 CPNs consist of 12 shelves, and 100 CPNs on each shelf form a grid-topology CPN network, as shown in Figures 10 and 11.

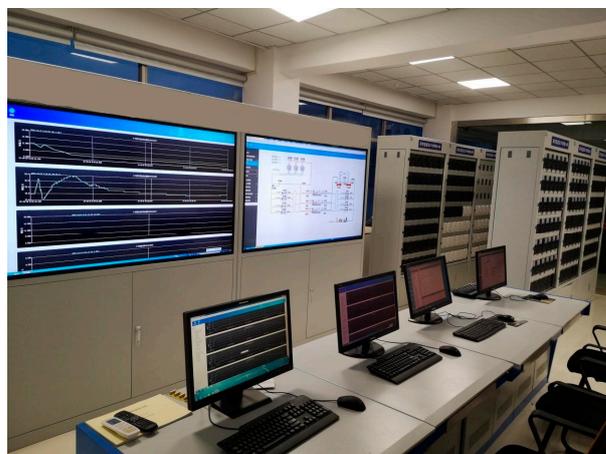


Figure 10. The physical site of AppSimV.



Figure 11. Large-scale CPN network.

The fundamental function simulation of AppSimV is accomplished by computing the quantity of individuals and altering the room temperature in predetermined real-world scenarios (such as during morning work hours in an office building). The test involves two types of four Apps, namely the room temperature summation App, the room temperature maximum value App, the room temperature minimum value App, and the App that keeps track of the total number of people in the room. These four Apps utilize parallel distributed algorithms and are centered on swarm intelligence technology, all providing a flat and centralized platform.

The specific testing process is as follows:

First, each App is encrypted and downloaded to the CPN network in the form of assigned frames. Check the basic information of the CPN network and confirm that all four Apps have been successfully downloaded.

Second, initialize the simulation platform model with the correct values, establish the initial number of individuals in each room to two, and assign an initial temperature of 24 to each room. Next, configure the initial values for the CPN network nodes and determine the physical variables for room temperature, room temperature summation, number of individuals in the room, and total number of individuals in the room, in accordance with the intelligent building object data standard set. By assigning physical variables to spatial nodes, it is possible to determine the initial number of people and temperature in the room for CPN network nodes. The initial values set by the CPN network and simulation platform models must be consistent. Confirm that the initial value of the variable was successfully set by querying the CPN network variable information. Simultaneously, assign a value of 1 to the App variable 0x1001 to the output nodes of the four App calculation results. Initiate and complete the simulation platform initialization.

Finally, we verify whether the App is running. While the simulation platform was successfully operating, the temperature value and number of people in the room were randomly assigned for the CPN network. During the simulation process, a program for generating random numbers was written to send the set values to six rooms every 10 s. The simulation platform will operate for an hour continuously to simulate the variations in the total number of individuals in the office building during morning work hours, as well as the temperature of the room.

5.2. Testing Room-Temperature Measurement App

To calculate the minimum, maximum, and average values of the room model temperature of the simulation platform, the room temperature is measured for the preset actual scene (such as the office building's morning working time) using the 1200 CPN node network. The anticipated experimental outcomes are as follows: initially, the output node App variable 0x1001 in the CPN network information should indicate a value of 1. Second, after the simulation platform is initialized, the log content of a successful connection to the CPN

network should be displayed. Afterward, the temperature App associated with the room model should accurately calculate the average, maximum, and minimum temperatures after operation. Finally, the results of the App data analysis must align with the changes in room temperature under actual scenario conditions. The experimental results are expected to be accurate, and the alterations in the temperature data of the room model during the simulation for an hour align with the predetermined actual scenario. The particular simulation operation data are depicted in Figures 12–14.

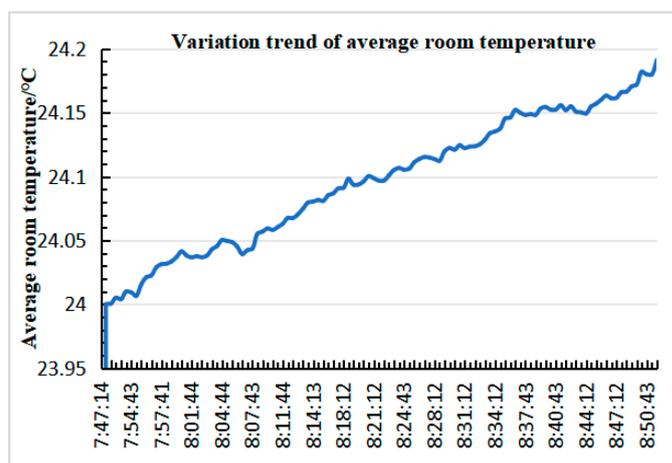


Figure 12. Trend of average room temperature over time.

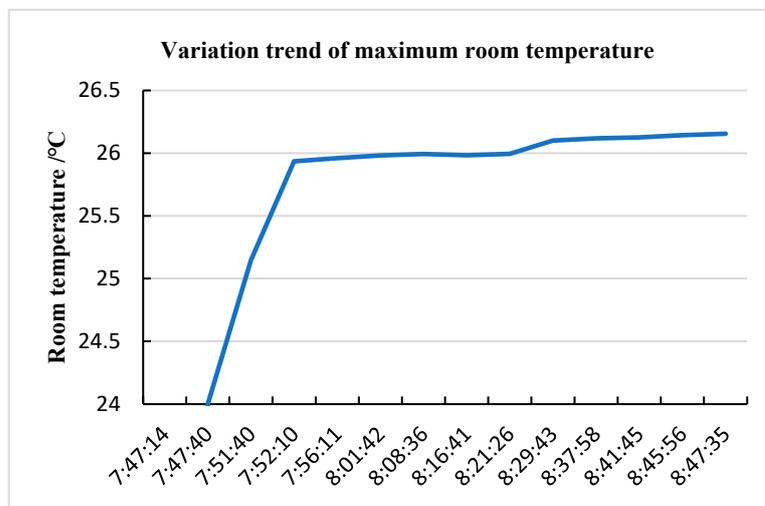


Figure 13. Trend of maximum room temperature over time.

In the predetermined actual scenario, it is scheduled to arrive at the work room at 7:47. As the working hours approach, the average temperature and the maximum room temperature both increase. The indoor air conditioning unit model available on the AppSimV platform detects the alteration in temperature by monitoring various sensors, subsequently adjusting the opening of the air valve to maintain a comfortable temperature range in the room. As illustrated in Figure 12, the average temperature of the room remains within a reasonable range of 24–24.2 degrees, with no notable increase. The maximum room temperature (Figure 13) remained consistently below 26.1 degrees, and the minimum room temperature (Figure 14) experienced a significant drop after the indoor air conditioner was switched on. During the App verification process, the AppSimV platform accurately simulates the actual engineering operation environment, thus realizing the rational utilization of resources. The App’s data are consistent with the actual situation, and the platform’s simulated data are accurate and trustworthy.

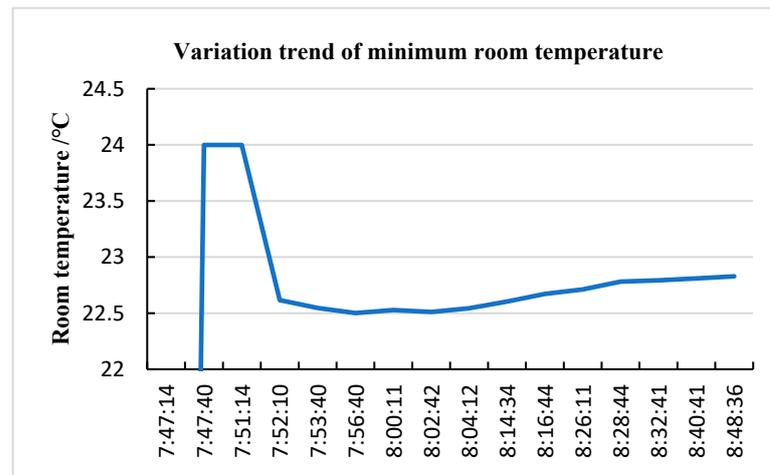


Figure 14. Trend of minimum room temperature over time.

5.3. Testing the App for Counting People in the Room

The simulation platform room's population is evaluated through testing against 1200 CPN node networks, which primarily involves counting the number of individuals in the simulation platform room model to ensure its ability to facilitate data interaction with the large-scale CPN network. The anticipated experimental outcomes are as follows: initially, the output node App variable 0x1001 in the CPN network information should indicate a value of 1. Second, after the simulation platform is initialized, the log content of a successful connection to the CPN network should be displayed. Afterward, the App should be able to accurately calculate the total number of individuals in all rooms. Finally, the results of the App data analysis must align with the changes in the total number of individuals in the room that occurred under the predetermined actual scenario. The experimental results are expected to be accurate, and the variation in the total number of individuals over an hour of simulation aligns with the predetermined real-world scenario. The particular simulation operation data are depicted in Figure 15.

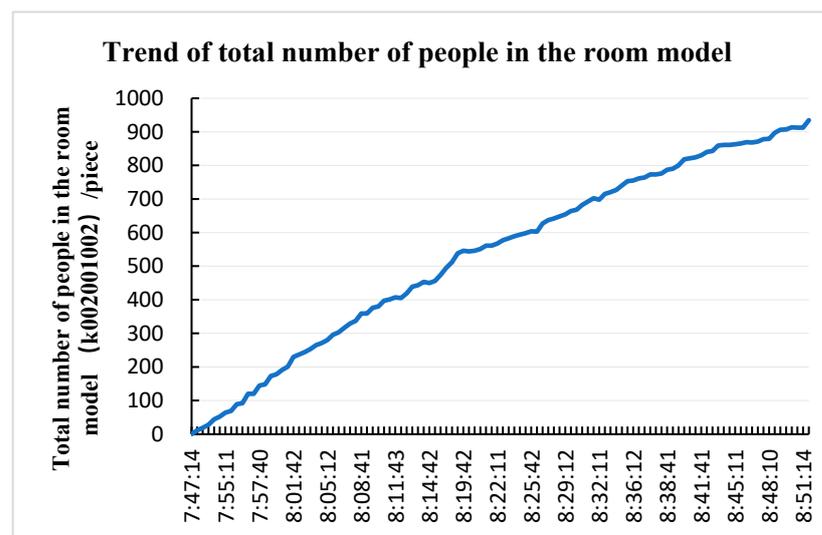


Figure 15. Trend of the total number of people in the room model over time.

As the working time approaches, the curve of the total number of people in the room approaches its peak, and the variation in the number of individuals is consistent with the current situation. As the number of individuals varies, the room temperature data correspondingly undergo changes. The platform possesses the data interaction capabil-

ity inherent in information physical equipment and the ability to simulate fundamental functions.

5.4. Discussion

Through the verification process from the APPSimV platform, the faults in Apps can be effectively found in advance before they are employed in the actual engineering site, which means the debugging time in the actual engineering site will be shortened. The results from many actual projects such as the Water Cube and China Resources Wujiang Vientiane show that the actual debugging time in the sites can be on average shortened from 4–5 months to 2–3 weeks.

6. Conclusions

To provide a comprehensive test environment for the operation of swarm intelligent Apps that closely reflects actual engineering scenarios, AppSimV, which is a cyber–physical simulation test and verification platform for swarm intelligent building Apps, is proposed. For the first time, the architecture of the semi-physical simulation of swarm intelligence building with cyber–physical fusion, dynamic expansion, and openness is established. Based on the idea of hardware-in-the-loop, the cyber–physical simulation architecture is formed. Apps run on the real CPN hardware, and basic building units run in the form of a simulation model within the simulation software. The App realizes the direct access to the building simulation unit through the middleware by transgressing the CPN physical system. A standardized and componentized encapsulation method of building simulation model based on software components is proposed, which enables the simulation model to be dynamically added, deleted, and modified in the simulation platform, and realizes the residence, execution, and interaction of the basic unit of building simulations in the information space, which ensures the strong dynamic scalability of the AppSimV platform. Based on this platform, experimental tests and applications of a typical engineering App are explored to verify the availability and reliability of the semi-physical simulation platform.

However, in general, the research work in this paper is preliminary, and there are still limitations in the interactive performance of the human–computer interface and the richness of the types of simulation models constructed. The author will carry out further research in these two aspects. First, the visual three-dimensional interactive human–computer interface of the platform will be enhanced, a three-dimensional display of engineering scenes will be built, and a better interactive experience will be provided for users. Second, we will further expand and enrich the internal simulation model, develop and build a simulation model for underground space units and facilities and equipment for underground engineering, expand the application scenarios of the platform, improve the reliability and stability of the platform, and accelerate its application process in multiscenario building engineering.

Author Contributions: All authors (H.J., Q.Y., Z.J., W.C. and Q.Z.) contributed to the study conception and design. Material preparation, data collection and analysis were performed by H.J. and Q.Y. The first draft of the manuscript was written by H.J. and all authors (H.J., Q.Y., Z.J., W.C. and Q.Z.) commented on previous versions of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was mainly funded by the National Natural Science Foundation of China, grant number 52178307, the Natural Science Foundation of Jiangsu Province, grant number BK20201335, the National Key Research and Development Project of China, grant number 2017YFC0704100, and was partially funded by the Natural Science of Jiangsu Province, grant number BK20210439, the Frontier Innovation Fund of PLA Army Engineering University, grant number KYFYJQZL2205, and the Natural Science Basic Research Plan in Shaanxi Province of China, grant number 2021JQ-374.

Data Availability Statement: The authors declare data is unavailable due to privacy.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

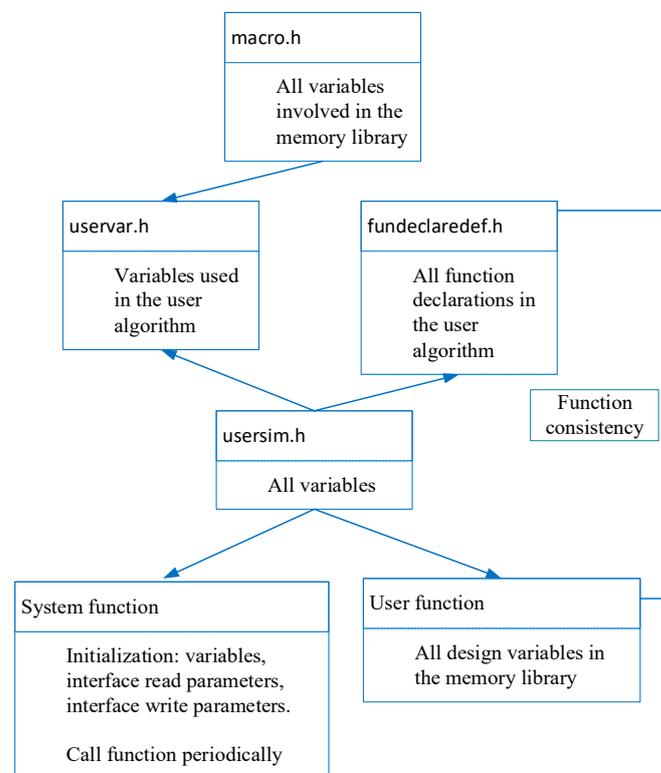


Figure A1. Relation diagram for the header files.

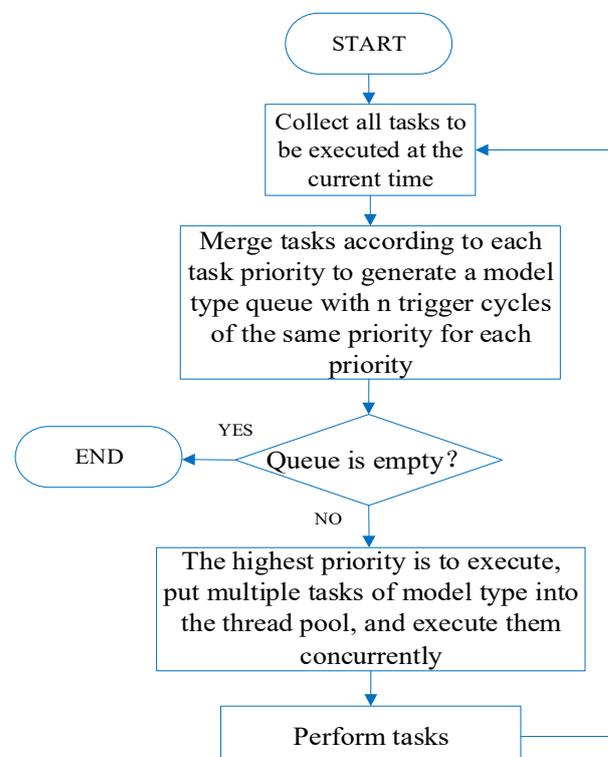


Figure A2. Algorithm flow chart of the multisimulation task scheduling execution engine.

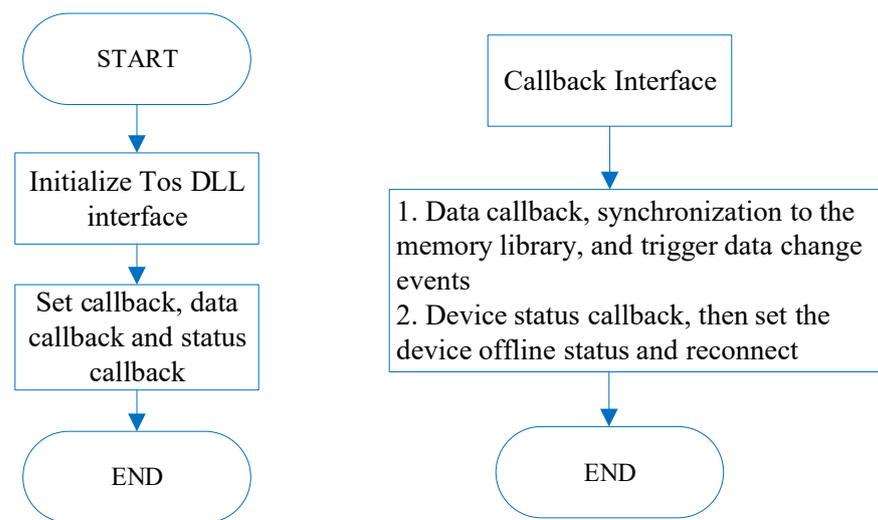


Figure A3. Flow chart of middleware implementation.

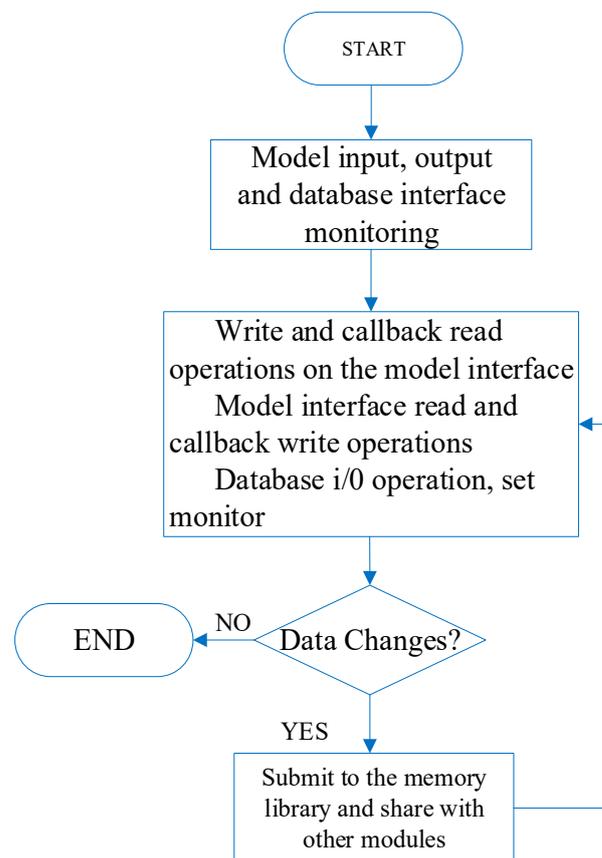


Figure A4. Algorithm flow chart of data monitoring implementation.

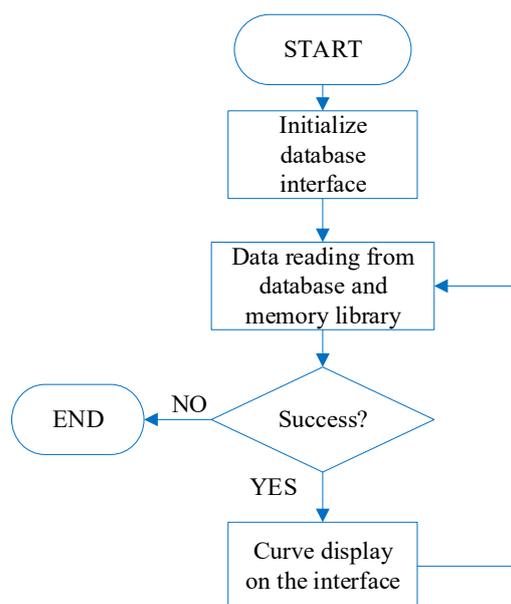


Figure A5. Algorithm flow chart of simulation display implementation.

References

1. Wang, W.W.; Yang, X.K.; Yu, D.; Sun, Y.; Deng, L.; Yang, J. Status quo of domestic intelligent building industry and development trend and opportunity of intelligent building technique. *Archit. Technol.* **2022**, *5*, 637–639.
2. Feng, L. Study on Life Cycle Quality Assessment of Intelligent Buildings. Master's Thesis, Xihua University, Chengdu, China, 2019. [CrossRef]
3. Ceglia, F.; Marrasso, E.; Roselli, C.; Sasso, M. Time-evolution and forecasting of environmental and energy performance of electricity production system at national and at bidding zone level. *Energy Convers. Manag.* **2022**, *265*, 115772. [CrossRef]
4. Ceglia, F.; Esposito, P.; Marrasso, E.; Sasso, M. From smart energy community to smart energy municipalities: Literature review, agendas and pathways. *J. Clean. Prod.* **2020**, *254*, 120118. [CrossRef]
5. Ceglia, F.; Esposito, P.; Farudello, A.; Marrasso, E.; Rossi, P.; Sasso, M. An energy, environmental, management and economic analysis of energy efficient system towards renewable energy community: The case study of multi-purpose energy community. *J. Clean. Prod.* **2022**, *369*, 133269. [CrossRef]
6. Loo, B.P.Y.; Tang, W.S.M. "Mapping" Smart Cities. *J. Urban Technol.* **2019**, *26*, 129–146. [CrossRef]
7. Kasznar, A.P.; Hammad, A.W.; Najjar, M.; Linhares, E.; Figueiredo, K.; Soares, C.A.; Haddad, A.N. Multiple dimensions of smart cities' infrastructure: A review. *Buildings* **2021**, *11*, 73. [CrossRef]
8. Froufe, M.M.; Chinelli, C.K.; Guedes, A.L.; Haddad, A.N.; Hammad, A.W.; Soares, C.A. Smart buildings: Systems and drivers. *Buildings* **2020**, *10*, 153. [CrossRef]
9. Kennedy, J. Swarm Intelligence. In *Handbook of Nature-Inspired and Innovative Computing*; Springer: Boston, MA, USA, 2006; pp. 187–219. [CrossRef]
10. Bonabeau, E.; Theraulaz, G.; Dorigo, M.; Marco, D. *Swarm Intelligence: From Natural to Artificial Systems*; Oxford University Press: Oxford, UK, 1999. [CrossRef]
11. Zhao, Q.C.; Jiang, Z.Y.; Chen, X. Swarm intelligent building intelligent system platform technology. *Intell. Build.* **2019**, *5*, 22–24.
12. Zhu, D.D. Swarm intelligent building control platform technology. *Build. Energy Effic.* **2018**, *11*, 7–10. [CrossRef]
13. Shen, Q. Studies on Architecture of Decentralized System in Intelligent Building. Ph.D. Thesis, Tsinghua University, Beijing, China, 2015.
14. Dai, Y.C.; Jiang, Z.Y.; Shen, Q.; Chen, P.Z.; Wang, S.Q.; Jiang, Y. A decentralized algorithm for optimal distribution in HVAC systems. *Build. Environ.* **2015**, *9*, 21–31. [CrossRef]
15. Wang, Z.G. Research on the design, construction and commissioning of high-efficiency computer room system in a high-end commercial building. *China Constr. Met. Struct.* **2022**, *5*, 63–65.
16. Tian, J.; Hao, C.C. Application of Energy and Environment Integrated Machine System. *Build. Energy Effic.* **2021**, *7*, 101–105. [CrossRef]
17. Su, B.; Wang, S.W. An agent-based distributed real-time optimal control strategy for building HVAC systems for applications in the context of future IoT-based smart sensor networks. *Appl. Energy* **2020**, *274*, 115322. [CrossRef]
18. Li, W.Z.; Wang, S.W.; Koo, C.G. A real-time optimal control strategy for multi-zone VAV air-conditioning systems adopting a multi-agent based distributed optimization method. *Appl. Energy* **2021**, *287*, 116605. [CrossRef]
19. Cihan, T.; Silvio, S.; Gulden, G.A. Evolvement of a personalized thermal comfort driven controller for HVAC systems. *Energy* **2021**, *237*, 121568. [CrossRef]

20. Li, M.Z.; Wang, Q.Q. Simulation of Online Energy Saving Control of Central Air Conditioning in Public Buildings. *Comput. Simul.* **2018**, *7*, 269–272.
21. Zhang, X.X. Development of Virtual Simulation Platform for Ground Source Heat Pump Air Conditioning System in a Low Energy Consumption Building. Master's Thesis, Shandong Jianzhu University, Jinan, China, 2019.
22. Yu, J.Q.; Liu, Q.T.; Zhao, A.J.; Zhang, R. Optimal chiller loading in HVAC System Using a Novel Algorithm Based on the distributed framework. *J. Build. Eng.* **2020**, *28*, 101044. [[CrossRef](#)]
23. Yu, J.Q.; Zhang, R.; Zhao, A.J.; Qian, X.G.; Liu, Q.T. Energy saving optimization insect intelligent control algorithm for parallel pumps in central air-conditioning system. *Control Theory Appl.* **2020**, *20*, 2155–2162. [[CrossRef](#)]
24. Li, S.S.; Wu, Y.Y.; Xu, C.Z. Compound Control Scheme Based on Repetitive Controller and Proportional Controller in Building Distribution Grid. *J. Phys. Conf. Ser.* **2020**, *1639*, 012025. [[CrossRef](#)]
25. Nassif, N.; Tahmasebi, M.; Ridwana, I.; Ebrahimi, P. New optimal supply air temperature and minimum zone air flow resetting strategies for VAV Systems. *Buildings* **2022**, *12*, 348. [[CrossRef](#)]
26. Lin, W.Y.; Ma, Z.J.; Li, K.H.; Tyagi, V.V.; Pandey, A.K. A dynamic simulation platform for fault modelling and characterisation of building integrated photovoltaics. *Renew. Energy* **2021**, *179*, 963–981. [[CrossRef](#)]
27. Lin, W.Y.; Green, A.; Gomis, L.L.; Kokogiannakis, G.; Cooper, P. Thermal effects of dew on cool roofs and conventional roofs in building performance simulations. *Energy Build.* **2023**, *279*, 112645. [[CrossRef](#)]
28. Tao, Y.X. Design and Implementation of Intelligent Building System Software Based on Semi-Physical Simulation Technology. Master's Thesis, Beijing University of Technology, Beijing, China, 2019. [[CrossRef](#)]
29. Guo, C.F. Construction of Virtual Simulation Platform of Wind Power Generation Based on PLC. Master's Thesis, Northeast Agricultural University, Harbin, China, 2019.
30. Sadraddin, H.L.; Cinar, M.; Shao, X.; Ahmed, M. Testing Platform and Delay Compensation Methods for Distributed Real-Time Hybrid Simulation. *Exp. Tech.* **2020**, *44*, 787–805. [[CrossRef](#)]
31. Zhao, Q.; Jiang, Z. Insect Intelligent Building (I2B): A New Architecture of Building Control Systems Based on Internet of Things (IoT). In Proceedings of the International Conference on Smart City and Intelligent Building, Hyderabad, India, 27–31 January 2018; Springer: Singapore, 2018; pp. 457–466. [[CrossRef](#)]
32. Wang, Z.; Wang, L.; Dounis, A.I.; Yang, R. Multi-agent control system with information fusion based comfort model for smart buildings. *Appl. Energy* **2012**, *99*, 247–254. [[CrossRef](#)]
33. Jiang, Z.; Dai, Y.C. A decentralized, flat-structured building automation system. *Energy Procedia* **2017**, *122*, 68–73. [[CrossRef](#)]
34. Zhao, T.Y.; Hua, P.M.; Zhang, J.L.; Jiang, Z.Y.; Jiang, A.; Wang, H.; Tang, W. Basic unit information model standard for swarm intelligent buildings. *Intell. Build.* **2019**, *5*, 19–21.
35. Hua, P.M. Application of Control and Information Model in Architectural Space Unit Based on I2B Platform. Master's Thesis, Dalian University of Technology, Dalian, China, 2019. [[CrossRef](#)]
36. Liu, X.G.; Xie, L.Q. Research on the Information Model Developing of Electrical Equipment of Insect Intelligent Building. *Build. Electr.* **2019**, *1*, 50–53. [[CrossRef](#)]
37. Klein, S.A.; Beckman, W.A.; Duffie, J.A. TRNSYS—A transient simulation program. *ASHRAE Trans.* **1976**, *82*, 623–634.
38. Park, C.; Clark, D.R.; Kelly, G.E. *HVACSIM+ Building Systems and Equipment Simulation Program: Building-Loads Calculation*; National Bureau of Standards—Building Equipment Division: Washington, DC, USA, 1986.
39. Yan, D.; Xie, X.N.; Song, F.T.; Jiang, Y. Building environment design simulation software DeST (1): An overview of developments and information of building simulation and DeST. *HV AC* **2004**, *7*, 48–56.
40. Yan, D.; Zhou, X.; An, J.; Kang, X.; Bu, F.; Chen, Y.; Pan, Y.; Gao, Y.; Zhang, Q.; Zhou, H.; et al. DeST 3.0: A new-generation building performance simulation platform. *Build Simul.* **2022**, *11*, 1849–1868. [[CrossRef](#)]
41. Zhao, D.G.; Yao, Y.H. Research on 3D Simulation of Fabricated Building Structure Based on BIM. *3D Res.* **2018**, *9*, 57. [[CrossRef](#)]
42. Huang, H. Research on Agent-Based Simulation Platform for Crowd Evacuation Considering Building Fire Impacts. Master's Thesis, Tsinghua University, Beijing, China, 2018. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.