buildings

MDPI

Article

A Foundation Model for Building Digital Twins: A Case Study of a Chiller

Suliang Li, Qiliang Yang *, Jianchun Xing, Wenjie Chen and Rongwei Zou

College of Defense Engineering, Army Engineering University of PLA, Nanjing 210000, China; suliang_li@126.com (S.L.); xjc@893.com.cn (J.X.); cwj13505152123@163.com (W.C.); zrwlyf@163.com (R.Z.)
* Correspondence: yql@893.com.cn

Abstract: Due to the high-fidelity mapping of the physical buildings and the intelligent performance shown in their lifecycle, digital twins (DTs) have gained increasing attention in the building sector. Although digital twins based on building information modeling (BIM) have become a hot research topic, existing works emphasize the digitization of building static and dynamic information and lack a unified consideration of the inherent physical mechanisms and interactive behaviors of buildings. To this end, this paper proposes a foundation model for building digital twins which realizes the unification of building static information, physical mechanisms and interaction patterns. The conceptual framework of the model is given first and then formal modeling and verification with time automata theory are performed to demonstrate the plausibility of the model. Finally, a practical digital twin of a chiller is developed based on the proposed foundation model as an example, thus, indicating its effectiveness and credibility.

Keywords: digital twins; foundation model; physical mechanisms; interaction patterns

check for updates

Citation: Li, S.; Yang, Q.; Xing, J.; Chen, W.; Zou, R. A Foundation Model for Building Digital Twins: A Case Study of a Chiller. *Buildings* **2022**, *12*, 1079. https://doi.org/10.3390/buildings12081079

Academic Editors: Vicente A. Gonzalez and Heap-Yih Chong

Received: 4 June 2022
Accepted: 22 July 2022
Published: 24 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 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 increasing requirements of buildings in terms of safety, comfort, convenience and energy savings, the prevalence of research and applications related to intelligent buildings is continuously increasing. Intelligent buildings call for higher flexibility in energy consumption and real-time interaction among the environment, personnel and various facilities. Research on the interweaving of virtual and physical buildings through various project stages, such as building information modeling (BIM) [1–3] and building automation systems (BASs) [4,5], has undergone significant development in recent years. As the virtual representation of the physical and functional characteristics of a facility, BIM provides a convenient repository for collaboration to share information during a facility's lifecycle [6]. Open BIM standards such as industry foundation classes (IFC) guarantee the generality and completeness of the data used in the design and construction process. However, the modeling capability of BIM in terms of dynamic behavior and building physical mechanisms is still lacking [7,8]. BIM aims to build an efficient digital model encompassing the entire lifecycle of a building while ignoring cyber–physical interactions [9]. As part of the shift from traditional approaches toward automation, BASs perform well in terms of energy quality [10], response efficiency and real-time interaction. By combining hardware, such as sensors and actuators, with software, such as operating logic, such a control system utilizes dynamic information to monitor and control the mechanical and electrical equipment of a building [11]. However, a BAS focuses primarily on facility management (FM) and participates less in other stages of the building lifecycle, thus, breaking the overall structure of a building information model [12,13]. Accordingly, it is more like an embedded system, which makes the information sharing between different building systems inefficient and prone to error [11].

Therefore, exploring a new technology that can establish bi-directional information flow between the physical building and digital model which integrates building information and intelligence to assist stakeholders in monitoring and controlling has become a concern in the building sector.

Digital twins (DTs) are considered an appropriate solution for the aforementioned issues. A DT is a living and evolving model endowed with intelligence, being the virtual counterpart of a corresponding asset [14]. DTs were first proposed with the intent of resolving thorny problems such as dramatic failure or exporting incorrect results in a complex system [15]. By virtue of a real-time digital representation that reflects the laws governing the internal and interactive behavior of the target asset as well as the integration of heterogeneous data and intelligent applications, a DT is able to represent the current state of the asset and predict future states to assist in the development of operation strategies [16], decision making [17], fault forecasting [18], etc.

The development of DTs in the building sector has also been explored in many papers [19]. Through digital twin technologies, the real state of physical buildings is reflected in the cyber world, and the integration of data across multiple dimensions presents a comprehensive view of building lifecycle management. DTs in the design phase present the design information and simulation results of buildings to stakeholders in advance, and iterative optimization based on DTs effectively reduces rework costs [20]. Sensors attached to construction projects transmit data to DTs, which support the arrangement of materials and workers in a reliable and timely way [8]. With the adoption of big data, artificial intelligence, machine learning, etc., DTs realize intelligent feedback for physical buildings, which has a significant impact on building operation and maintenance. Verification of operation and maintenance solutions with the help of DTs can be performed without disrupting the normal operation of the buildings [21]. The value of data integration and mining in DTs is also evidenced by efficient troubleshooting and predictive maintenance. Using real-time performance data, DTs can identify problems in physical buildings in a timely manner and realize continuous building health monitoring [22]. Traditional, preventive maintenance activities become predictive maintenance, and intelligent forecasting and scheduling through DTs save time and costs and minimize the impact of unexpected breakdowns [23].

Nevertheless, these studies usually cover only one phase of the building lifecycle, and there is a lack of systematic discussion on the continuous evolution of building digital twins (BDTs). Researchers have established a number of BDTs for special cases but have less often discussed the fundamental components of a generic BDT. Although BIM-based building digital models are widely used in the building sector, the BDTs derived by extending BIM still cannot solve the problems of dynamic behavior description and physical mechanism expression.

This study aims to explore a model that supports the mapping of building information to the virtual world and takes into consideration the integration of building physical mechanisms and dynamic behaviors. Based on this model, a domain-specific modeling method is used to implement the research of modeling mechanisms for BDTs and to provide a dedicated, effective and credible description method for BDT models.

Domain-specific modeling restricts the problem to a smaller scope and abstracts concepts and rules in the problem domain. These concepts and rules are used in the description of model information to obtain a metamodel, which defines modeling elements and constraints. A standardized and consistent model representation enables the model to support deconstruction and reuse, improving the generality and portability of the model.

However, the plausibility of a BDT metamodel which is positively constructed by domain analysis needs further rigorous verification. Formal verification is a mathematical proof method for checking whether the relevant properties of a system are satisfied. Errors such as inconsistencies, ambiguities and incompleteness are detected as much as possible.

BDTs evolve along with the corresponding physical buildings in the lifecycle. The real-time interaction between the virtual world and reality indicates that the BDT system is a time-constrained system. Therefore, a time automata approach is applied in this study to verify that the foundation model satisfies the domain properties of the BDTs.

In summary, the main contributions of our work are as follows:

(1) We establish a foundation model for BDTs. This model provides a framework for BDT construction. This framework consists of three submodels: a static information model, a physical model and an interaction model. A BDT constructed in this way can serve as a high-fidelity virtual replica of a physical building and coevolve with it in the lifecycle;
(2) We use object-oriented modeling ideas to refine the foundation model into a collection of entities, parameters, behaviors and pipes, whereby a metamodel of BDTs is proposed. Furthermore, the deadlock-free, reachability and consistency characteristics of the foundation model are verified using a time automata network to indicate the credibility of the model;
(3) We validate the proposed foundation model by constructing a chiller DT with the aim of highlighting the effectiveness of the foundation model in constructing multiphysics and multiscale BDTs for advanced management.

The rest of the paper is structured as follows: Section 2 discusses related studies on the use of DTs in the building sector and identifies the problems that they have solved. Section 3 introduces the foundation model, which allows the characteristics of dynamic interactions to be described in detail by virtue of the coupling of the submodels. By using a time automata network, the foundation model is formally validated. Section 4 presents a real case study of a chiller DT to validate the efficiency and instructiveness of the proposed foundation mode. Section 5 discusses the contributions and continuing work of this paper. Section 6 concludes the paper.

2. Literature Review

According to a statement by the research firm Gartner, Inc. in 2018, DTs are expected to be one of the top ten emerging technologies in the next decade [24]. Indeed, the potential of accelerating digitalization and integrating physical and virtual assets has already been explored, mainly in aerospace and manufacturing [25]. To improve the performance of a vehicle in a mission, NASA and the US Air Force use a DT for ultrahigh-fidelity simulations in the processes of certification, health management and mission monitoring. The structural behavior of the vehicle can be analyzed and predicted based on the digital replica through the integration of the maintenance history and real-time data [26]. The role of DTs in intelligent manufacturing is also crucial. In the era of Industry 4.0, manufacturers are pursuing a greater level of technical innovation with enhanced flexibility, adaptability and predictability. DT-driven production control systems can respond more sensitively to customer demands and environmental changes and help to optimize plant operations during the production process [27].

Due to the promising achievements enabled by DTs in manufacturing, DT technology is also attracting increasing interest in the building sector. Because BIM can provide an information-rich and visualized 3D model that spans the lifecycle of a building, from design to operation and maintenance, a BIM model is considered a starting point for a corresponding DT model. For instance, Boje et al. reviewed the literature on BIM applications and proposed the concept of construction digital twins (CDTs). This concept completes the semantic representation of BIM in control systems and has the potential to support the management of value-added data and dynamic processes [28]. Lu et al. automated the generation of geometric DT models of existing buildings from images and computer-aided design (CAD) drawings to achieve accurate and convenient model construction [29]. Pan et al. emphasized the importance of data processing and analysis in improving the effectiveness of DT models; accordingly, they incorporated the Internet of Things (IoT) and data mining techniques into BIM with the aim of discovering the full value of the collected data [27].

The aforementioned efforts show the promising potential of DTs in the building sector, but it is also notable that more attention is given to developing DTs by attaching extra data to BIM. Although there is a relationship between BIM and BDTs, BIM lacks dynamic semantics, which makes it difficult to achieve consistency in the expression of building static information and dynamic interactive behavior, and it cannot support the portrayal of physical processes [30].

In addition, some practical applications of DTs in the construction industry that support smarter construction services have also been proposed. Lu et al. asserted that more attention should be given to effective operations and maintenance because of the immense value potential in this context. Specifically, a system architecture with five layers was proposed for integrating heterogeneous data and constructing a dynamic DT model of the corresponding assets at the building level, thereby enabling intelligent management and enhancing interaction with facility managers [19]. Angjeliu et al. developed a procedure for creating detailed DT models of historical masonry buildings for structural analysis. In this procedure, the real geometry, material properties and internal structure of a building were digitalized with high precision. When such a model is combined with a nonlinear finite element model, operations such as assessing structural response, simulating maintenance actions and predicting mechanical damage are possible [31]. Likewise, Lu et al. proposed a DT-based anomaly detection method for asset monitoring. An extension of IFC was used as a data structure to realize data integration and operation. In addition, considering the changing loads of assets during operation, a Bayesian methodology for identifying anomalies in context was integrated into the framework. This method improved the efficiency and automation of real-time monitoring [22]. Kaewunruen et al. focused on information sharing throughout the lifecycle of a railway turnout system and established an innovative 6D BIM model spanning from the design phase to the demolition phase based on a 3D model established with Revit software, with the time schedule, cost data and carbon emission calculation composing the other three dimensions. Such integration promotes design and construction efficiency as well as maintenance predictability and reduces cost and construction error. Moreover, evaluation of the carbon footprint is significant for environmental protection [32].

BDTs first need to realize the mapping of physical building information to the virtual world, which includes the static property information and dynamic information. Pan et al. proposed a novel framework to enrich the geometric, semantic and text information of BDTs. Laser scanning and object detection in images are fused to realize nearly fully automated information mapping [33]. Zhou et al. introduced DT technology to improve the construction management efficiency of prefabricated buildings. The actual situation, such as displacement and rotation torque of the building components, was effectively monitored. Great benefits were brought by the DT-oriented management control system [34]. The mapping of static and dynamic information enables the virtual world to mirror the physical building, but this is only the first stage of the construction of BDTs. The second stage is realizing the feedback of information from the virtual world to the physical world. Zhao et al. established a DT framework for intelligent building operation with the integration of a DT model and machine learning algorithm and validated the data-driven intelligent prediction and diagnosis in real-time interactions at the theoretical level [35]. Zhao et al. adopted DTs for facility management activities in an illustrative case study. Four representative use cases were investigated to show the benefits of DTs for real-time data collection and monitoring, decision making, predictive maintenance and cost reduction [23]. Peng et al. conducted a continuous lifecycle integration study on the DT of a hospital building. The DT project was performed over one year. Facility diagnosis and operation suggestions supported by dynamic, integrated data resulted in increased management satisfaction, while energy consumption and facility failures were reduced [36]. Liu et al. integrated DTs and a global navigation satellite system (GNSS) to establish intelligent, closed-loop control. The information captured by GNSS in real time realized the precise positioning of operation and maintenance elements and provided a data basis for analysis management and maintenance decisions [37].

These papers have made meaningful efforts in continuous data integration and virtual–real interaction in BDTs. In terms of addressing specific problems, there has been less systematic discussion of the essential components and properties of BDTs. In particular, a unified conceptual modeling framework and domain-specific modeling approach are absent when implementing abstract descriptions for BDTs. This leads to two problems in the development of BDTs. The first is inefficient information communication. The expressions and semantics of different description methods vary, making it difficult for developers to communicate on the basis of the same cognitive context. The second is the shortage of model reusability. Non-uniform modeling approaches bind each BDT model and case closely, and the model deconstruction and reuse are hard to execute. Although there are comparatively few studies on the construction paradigm of BDT models, a specialized and standardized method for describing BDT models is significant for improving the efficiency of model construction and promoting the development and application of BDTs.

3. Methodology

A model is an abstraction of a particular aspect of the system, and the model created varies with the focus of the research. To achieve the goals of optimized operation, assisted decision making and fault prediction in building activities, one form of BDTs is as software entities with intelligent computing capabilities. As the intermediate form between the design and realization of the BDTs, the model is critical for expressing design requirements and driving the development process. By analyzing the domain characteristics and refining the essential components and internal relationships of BDTs, the foundation model is formed as the conceptual framework of BDTs to guide the construction of BDT models. Utilizing the metamodeling method to organize the components of the foundation model with consistent definition rules, the transformation of the foundation model components into the BDT modeling elements is accomplished so that the BDT metamodel is obtained. The metamodel offers a constraint and consensus for BDT domain modeling, which not only ensures the accuracy of the information conveyed by the BDT models and improves communication efficiency, but also correlates more closely to the programming language and contributes to the model-driven software implementation.

3.1. Components of the Foundation Model

From the summary and analysis of the research on BDTs in the literature review, it was found that the virtual–real symbiosis between the BDTs and the physical buildings relies on comprehensive information perception and continuous interactive feedback. In the process of interaction, to achieve more accurate predictions and more optimal decision making, simulations of the physical mechanisms of the corresponding physical building are also essential. Maintaining the same physical properties as the physical buildings provides these abilities for BDTs without interfering with the normal operation of the physical building. BDTs achieve a high degree of consistency with physical buildings in both appearance and substance. The essential features of BDTs are shown in Figure 1.

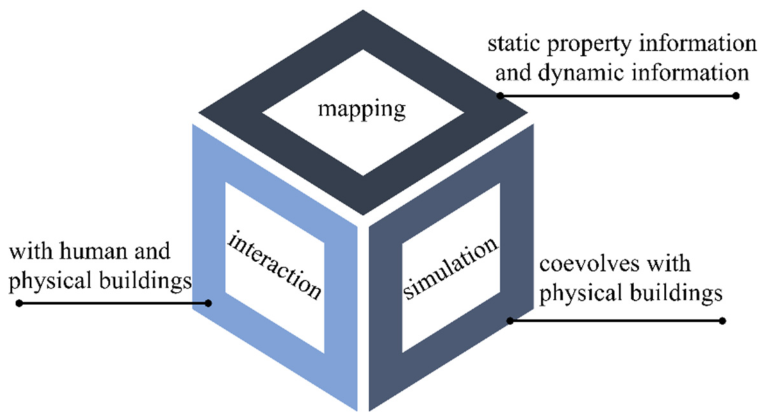


Figure 1. Essential features of BDTs.

As shown in Figure 1, the foundation model for BDTs includes both passive mapping of information and active interaction and can also reflect a high-fidelity portrayal of the physical laws; thus, a triple architecture of the foundation model, which can be depicted as the following expression, is proposed:

$$\mathrm{BDT} = (\mathrm{SIM}, \mathrm{PM}, \mathrm{IM}) \tag{1}$$

where SIM refers to the static information model, PM is the physical model and IM stands for the interaction model.

3.1.1. Static Information Model

The static information model maps the constituent elements and static properties of buildings to the virtual world, which reflects the static attributes of building elements, such as shape, material, size and location. The dimension and granularity of this information are diverse, and the static properties are convenient to be presented visually; hence, the key to constructing a static information model is the organization and visualization of the information. To clarify this, we represent the static information model as Equation (2).

$$\mathrm{SIM} = (\mathrm{BDTRecorder}, \mathrm{BDTVisualizer}) \tag{2}$$

where BDTRecorder is mainly employed to realize the storage, organization and management of model data, and BDTVisualizer is a declaration of the model rendering engine with responsibility for the visualization of the 3D model.

The preceding shows that the static information model is highly consistent with BIM, which is widely applied in the building sector [38]. As a novel model paradigm, it is acceptable to take BIM as the realization of a static information model to fully utilize the information integration capability and mature 3D visualization technology of BIM. For buildings that have established BIM models, BIM implementation as a static information model also reduces costs and improves development efficiency.

3.1.2. Physical Model

A physical model is a digital description of the rules governing the building behavior involved in a variety of physical processes. They may be divided into time-varying characteristics and functional characteristics. The former describes fatigue, aging and depletion throughout the building operation process, and the latter describes dynamic responses to outside stimuli [39]. The core of the physical model is to maintain an accurate simulation of the physical mechanism over time; hence, the concept of "evolution" is employed to demonstrate the coexistence of a BDT and the corresponding physical building in this paper, and the physical model is expressed as Equation (3).

$$\mathrm{PM} = (\mathrm{BDTCondition}, \mathrm{BDTSimulator}, \mathrm{BDTEvolver}) \tag{3}$$

where BDTCondition represents the specific conditions under which a physical process occurs, since even the same physical process differs in the physical properties it exhibits under different conditions. BDTSimulator represents a simulation of physical mechanisms and works in conjunction with BDTEvolver to achieve a synchronized reflection of the physical properties. Under the specific conditions represented by the BDTCondition, BDTEvolver defines when and in what way the model encapsulated by the BDTSimulator is updated, thus, enabling the BDTs to evolve with the physical buildings.

Accurate fitting of the physical mechanism of buildings is the key to constructing a physical model. Some physical processes that involve few physical variables and straightforward relationships can be described by mathematical models. However, for some complicated physical processes, mathematical models must be established under a simplified premise or idealized conditions. This approach reduces the difficulty of modeling but weakens the realism of the model.

The "white-box model" that relies solely on mathematical models cannot satisfy the simulation of complex physical processes, and a data-driven "black-box model" can be built with the assistance of neural networks, artificial intelligence, big data computing and other intelligent approaches to further enhance the accuracy of physical process fitting. The "black-box model" pays less attention to the strict mathematical relationships between the variables in the model and concentrates on the calculation outcomes rather than the interpretability of the model [40].

This data-driven model construction method facilitates online updates of the model. By comparing the predicted output with the actual output, new data can be used to iteratively fit the model when the error exceeds the threshold. The parameters of neural networks or intelligent algorithms of the model are corrected so that accuracy is continuously optimized. Internal changes to the building are transmitted through these new data, enabling intelligent self-evolution of the BDT.

3.1.3. Interaction Model

The interaction model establishes bi-directional information flows between the physical building and the BDT, which provide feedback to the physical world instead of mere simulation. The running data that flow from the physical world into the cyber world enrich the dynamics of BDTs and provide data input for the intelligent computation of BDTs. Computational modules that encapsulate the control strategy utilize these data to solve for the optimal control solutions. When the physical building fails, control modules assist operation and maintenance staff in troubleshooting and preparing appropriate solutions by linking multidimensional data. Computational advantages reduce the time to diagnose faults.

For intelligent decisions generated by the interaction model, it is necessary to further specify the execution components of the control instructions and encapsulate the instructions into a form of data identifiable to the executing component. Therefore, a set of virtual actuators corresponding to the physical actuators is also required in the interaction model to serve as an interface for feedback from the BDT to the physical building.

To present the interaction behaviors of the BDT and the physical building, a dynamic visualization module should also be integrated into the interaction model. This enhances the dynamic visualization capability of BDTs and supplements the 3D images of static information.

Hence, the interaction model of BDTs is expressed as Equation (4).

$$\mathrm{IM} = (\mathrm{BDTSensor}, \mathrm{BDTController}, \mathrm{BDTActuator}, \mathrm{BDTAnimation}) \tag{4}$$

where BDTSensor binds to the physical sensor and serves as the entrance of real-time data for BDT. The BDTController is the control center, which integrates dynamic and static data with intelligent algorithms to optimize decision making, fault diagnosis and situation prediction. The BDTActuator encapsulates control instructions into specific formats which can be recognized by the physical actuator. BDTs implement feedback to the physical world. BDTAnimation dynamically displays the operating status of the BDT to achieve consistent presentation of the BDT and the physical building in motion status. Figure 2 shows the collaboration between the four components and the control closed loop formed with the physical building.

The solid line in Figure 2 indicates that the element at the tail of the arrow is attached to the element at the head, and the dotted line means that the element at the tail of the arrow transfers data to the element at the head.

The interaction model guarantees the dynamic synchronization of BDTs with physical buildings. With the assistance of computing advantages, physical buildings run in a smarter, more energy-efficient way.

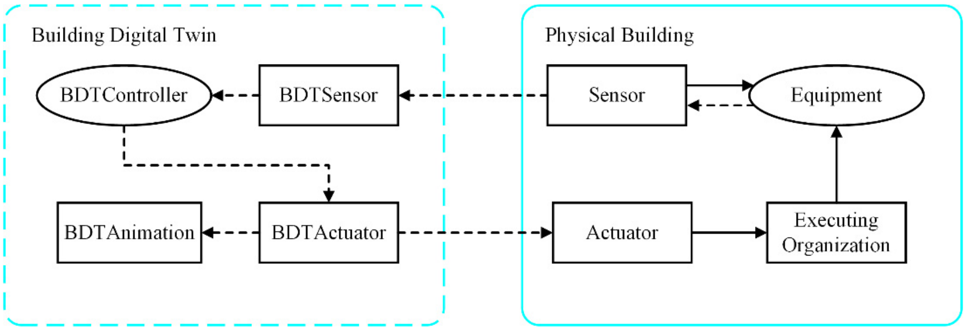


Figure 2. The process of interaction between the BDT and the physical building.

3.2. Coupling Relationships between Submodels

The static information model, physical model and interaction model are the basic components of BDTs which represent different characteristics: the static information model is a replica of the existing information of the physical building; the establishment of the physical model ensures the internal consistency between the BDT and the physical building; and the interaction model establishes an effective interaction mechanism between the cyber and physical world. The coupling relationships between the three are illustrated in Figure 3.

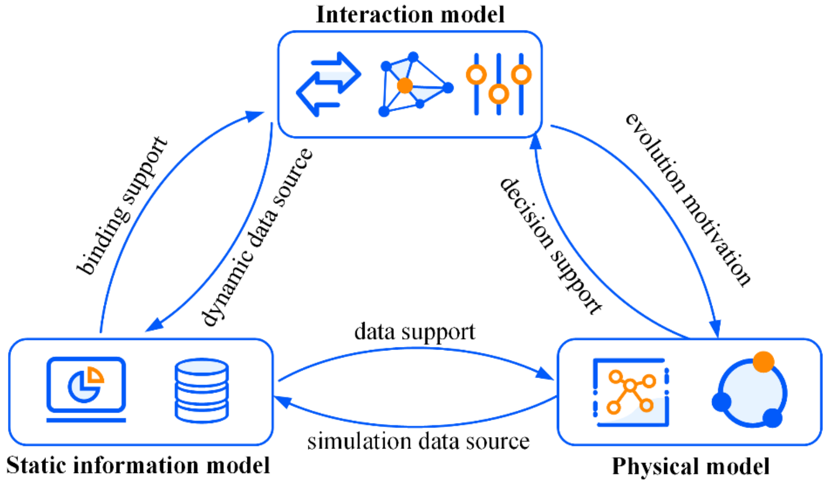


Figure 3. Coupling relationships between submodels.

3.2.1. Static Information Model and Physical Model

As a warehouse for static information, a static information model not only records physical building design information and historical operating data but also provides physical process models and simulation data records for the physical model. Through the accumulation and analysis of these data, the source of errors in modeling can be found, and the effect of twinning can be evaluated. As the warehouse of the physical model, the static information model can also compare the calculation results of the same process to select the best model and realize the online switching of models to maintain the optimal reflection of the physical building.

3.2.2. Static Information Model and Interaction Model

The static information model stores a large number of static attributes of physical buildings, including identifier information. By providing this information to the interaction model, the interaction model automatically binds to the corresponding physical sensor and physical actuator through the binding mechanism. Both the transmission of real-time status data from physical to cyber and control instructions from cyber to physical are realized.

There are two types of data in the interaction model that are particularly valuable to the self-evolution and function performance of BDTs. One is the real-time status information of the physical buildings obtained by the interaction model. The snapshot at the previous moment continuously becomes static historical data, and the accumulation of a large amount of historical data is an important basis for predicting the future state of physical buildings because these data reflect the actual characteristics of physical buildings under specific conditions.

The other type is the control instruction information of the interaction model. The actual effects of these instructions (i.e., status information of physical buildings) are reported by closed-loop feedback control. The combination of control instructions and effects become an effective reference for formulating future control strategies.

These two types of data become static information that cannot be changed at the next moment and are stored in a static information model to enrich the lifecycle information of BDTs.

3.2.3. Interaction Model and Physical Model

The real-time state of physical buildings perceived by the interaction model provides a reference for the accuracy of the physical model. When the comparison error between the simulation output of the physical mechanism process and the actual value exceeds the acceptable range, it works as a trigger signal to drive the physical model to improve accuracy. This process runs through the lifecycle of a BDT so that the reflection of the physical mechanism is always in a dynamic approach.

For some large and complex equipment systems, each operation must be carefully demonstrated to ensure that the possibility of misoperation or unsatisfactory control effects is minimized to reduce the interference with the normal building environment and costs. A physical model is an accurate reflection of the physical mechanism of a building. Experiments based on physical models are not limited by time and space and have a high degree of credibility and do not need to interrupt the operating state of the existing building equipment. Therefore, control operation and failure prediction are necessary to rely on a physical model for preverification and reasoning.

3.3. Metamodel of Building Digital Twins

The foundation model provides the conceptual framework for BDTs, and there are various ways to describe this framework. However, diverse description approaches are not conducive to the sharing and exchange of BDT information, and the explanation of the same BDT can be ambiguous due to the differences in representation. Continued development of the foundation model is limited.

A unified, standardized approach to modeling BDTs is necessary. Standard modeling elements and interpretation maintain the consistent meaning of the model under various scenarios so that the certainty of the model information is ensured. In addition, as an intermediate state of BDTs, the platform-independent model keeps information from distorting when driving the implementation of different forms of BDT [41].

In the foundation model, each submodel is represented as consisting of a specific set of elements, and the combination of these elements forms an abstraction of the BDTs. It is reasonable to adopt them as modeling elements for the BDT model. In the field of software engineering, models are described using metamodels [42], which define the constituent elements of models and the relationships between the elements. In this paper, the metamodel of BDTs, defined based on the foundation model, is shown in Figure 4.

The metamodel of the BDTs is divided into three layers. The meta-term layer provides the basic corpus for BDT modeling. The concept of BDTEntity is defined in the meta-term layer as an aggregate with the same properties and behaviors. Meanwhile, BDTParameter and BDTBehavior are defined to represent the properties and acts of BDTEntity. The relationships between BDTEntity are expressed by BDTPipe.

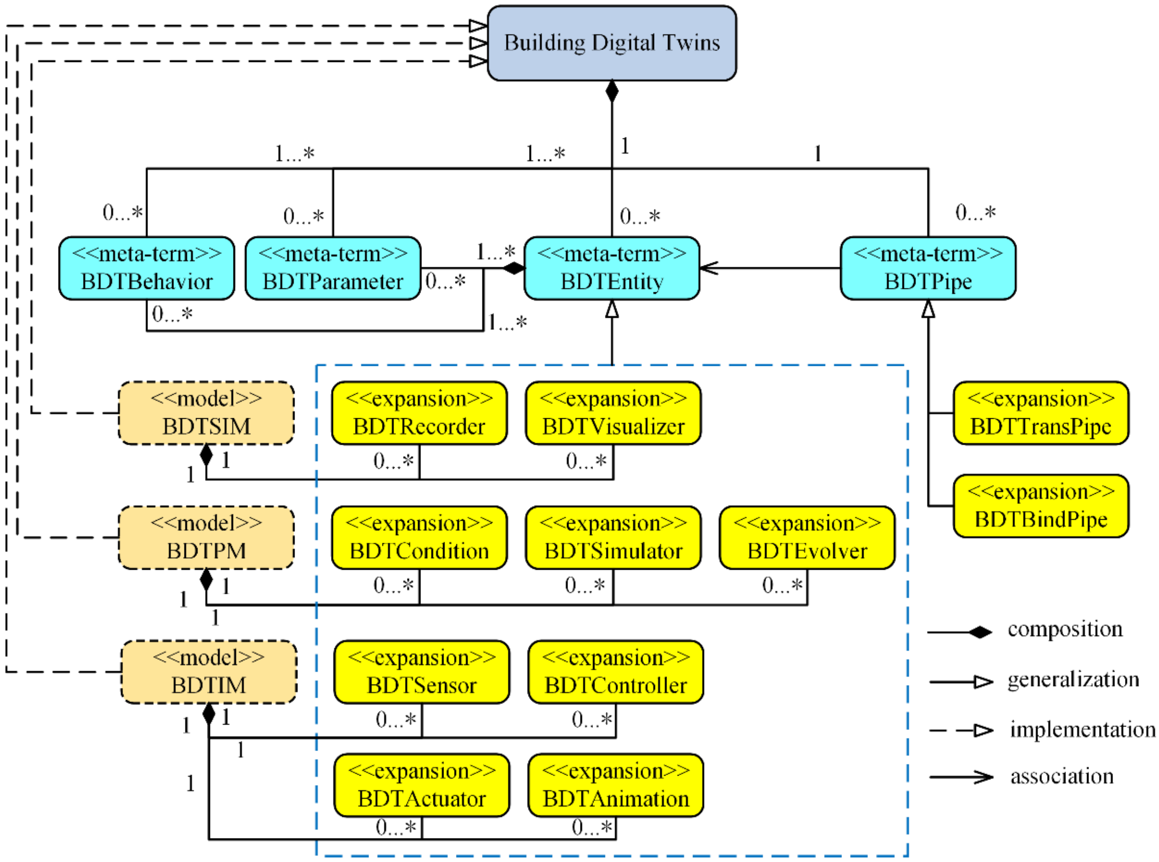


Figure 4. Metamodel of building digital twins.

The model layer defines the implementation forms of BDTs based on the triad structure of the foundation model. The model layer includes BDTSIM (representing the static information model), BDTPM (representing the physical model) and BDTIM (representing the interaction model) as the concrete implementations of BDTs.

To fit the actual modeling requirements, modeling elements need to be more domain specific, and features of BDTs are encapsulated in modeling elements in advance to improve modeling efficiency and specify the information expression of the model. The specification of BDTEntity and BDTPipe constitutes the expansion layer. Section 3.1 defines the constituent elements for the foundation model that are appropriate as expansions of BDTEntity, assigning characteristics of information mapping, physical mechanism simulation and dynamic interaction to BDT models. In addition, to distinguish the relationships between BDTEntity, BDTPipe is extended as BDTTransPipe and BDTBindPipe, which denote the data transfer relationship and logical composition relationship, respectively.

To further illustrate that the BDTEntity in the metamodel can effectively achieve realistic reflection and feedback control to the physical world through collaboration and complete BDT self-evolution, a sequence diagram constructed with unified modeling language (UML) is used for illustration, as shown in Figure 5.

Two types of entity, PhysicalSensor and PhysicalActuator, are added to the time sequence diagram, representing the source of building information in the physical world and the BDT feedback control object. PhysicalSensor transmits the real state of the building to BDTSensor, and BDTSensor synchronizes the real-time information to BDTController in the interaction model, BDTRecorder in the static information model and BDTCondition in the physical model. The BDTController judges whether the building is operating in a good zone or whether there is fault potential. BDTAnimation is notified to execute the rendering of the digital model. BDTConditon selects the appropriate physical mechanism for BDTSimulator to load based on the context. BDTEvolver evaluates whether the running

BDTSimulator meets the application requirements. When the error limits are exceeded, BDTEvolver performs an update process to keep BDTSimulator synchronized with the physical building in real time.

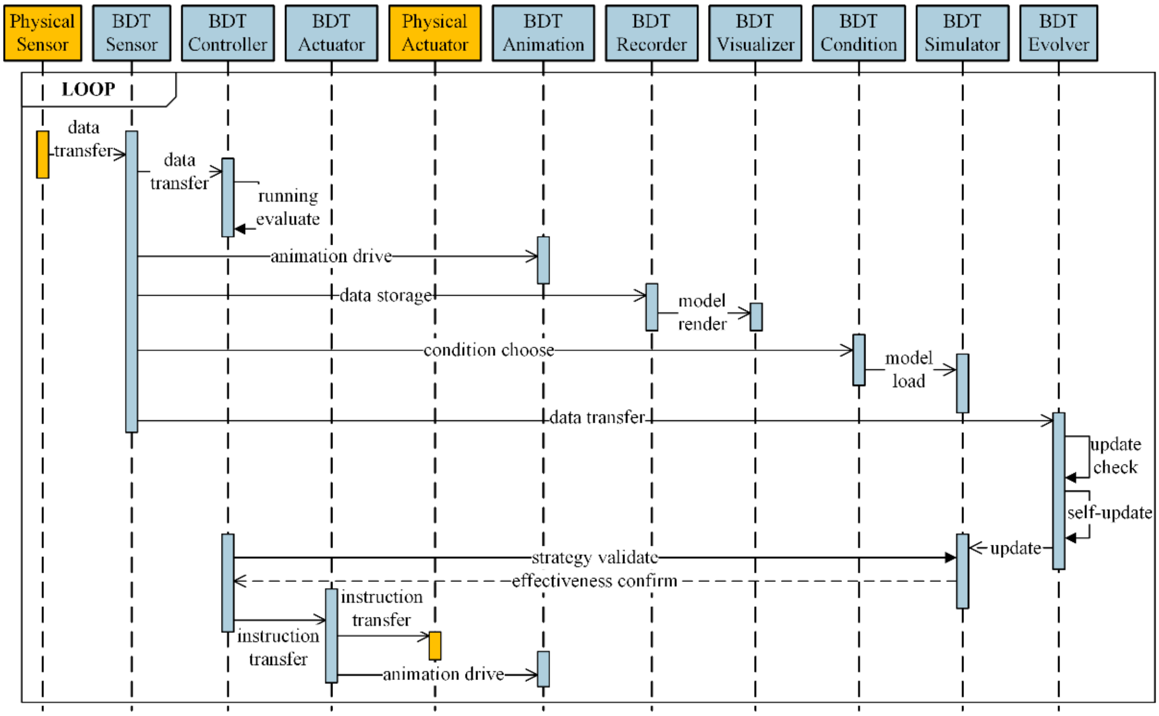


Figure 5. Sequence diagram of BDTEntity in the metamodel.

Intelligent calculations of the BDTController on operation optimization and fault diagnosis need to be validated by BDTSimulator, which satisfies the same physical mechanism as the physical building, to complete validation without interrupting the normal operation of the physical building. Finally, the validated control policies are encapsulated as hardware instructions by the BDTActuator and passed to the corresponding physical actuator, where the actions involved are rendered by BDTAnimation.

The above cyclic process occurs in the lifecycle of the physical building, forming a closed loop of control with dynamics, intelligence and interaction.

3.4. Formal Modeling and Verification

Sections 3.1 and 3.2 elaborate the composition of the BDT model and the internal coupling relationships. As to whether the foundation model can achieve the expected functions, i.e., whether a BDT constructed based on the foundation model can form a complete two-way data transmission system with the physical building, and give full play to the computational capability to provide intelligent assistance such as operation optimization, fault detection and state prediction for real operation and maintenance activities, the foundation model for BDTs still needs to be rigorously verified.

Formal verification is a rigorous proof method based on mathematical knowledge for testing the correctness of the properties of a system. The use of formal verification supports the understanding and analysis of the system and the detection of errors, such as inconsistencies, ambiguities and incompleteness, in it as much as possible.

A DT, which accompanies the lifecycle of the corresponding physical entity and responds in real time to contextual changes, is a time-constrained system. As a formal modeling and verification theory, time automata support the intuitive portrayal of time-dependent behavior in systems [43]. In our research, the formal verification of the foundation model for BDTs is completed using the model checking tool UPPAAL (version 4.1.26), which was jointly developed by Uppsala University in Sweden and Aalborg University in

Denmark. UPPAAL has a user-friendly visual interface and excellent computational performance [44] and has been successfully applied to embedded systems [45], IoT systems [46] and cyber–physical systems [47] for real-time analysis and verification.

3.4.1. Time Automata Modeling

Time automata modeling is divided into two steps: first, we establish the time automata of each entity in the foundation model; then, we define the global channels of message and flags to make time automata interact, thus, obtaining a time automata network. Based on the sequence diagram of Figure 5, eleven time automata are established, as shown in Figure 6.

As in the sequence diagram of Figure 5, the time automata network is also defined to execute cyclically, which is achieved by defining a sense state for PhysicalSensor every five units of time. Via the M_RealState message channel, the building real-time operation data trigger BDTSensor into the send2PM and send2IM states. BDTCondition and BDTController receive running data through the M_RealState2PM and M_ RealState2IM message channels, respectively. At the same time, the F_WriteData flag is set to 1, which triggers BDTRecorder to enter the state of write. In this state, the running data are saved and then the F_WriteData flag is set to 0 to wait for the next write.

BDTCondition chooses to load the appropriate physical model. After loading the physical model, the BDTSimulator notifies BDTEvolver via message channel M_Check to validate the current model, thus, deciding whether to keep the current model or to update it.

BDTController determines whether to enter the state of optimization or prediction. Both states need a physical model to verify the calculation result, so it obtains the assistance of BDTSimulator through message channel M_Verify. BDTSimulator uses message channel M_ Confirm to return the confirmation to BDTController. This process is repeated until the control target is satisfied. To simplify the description of the process, this process is executed once by default. Control instructions are sent to BDTActuator for execution through the message channel M_Instruction.

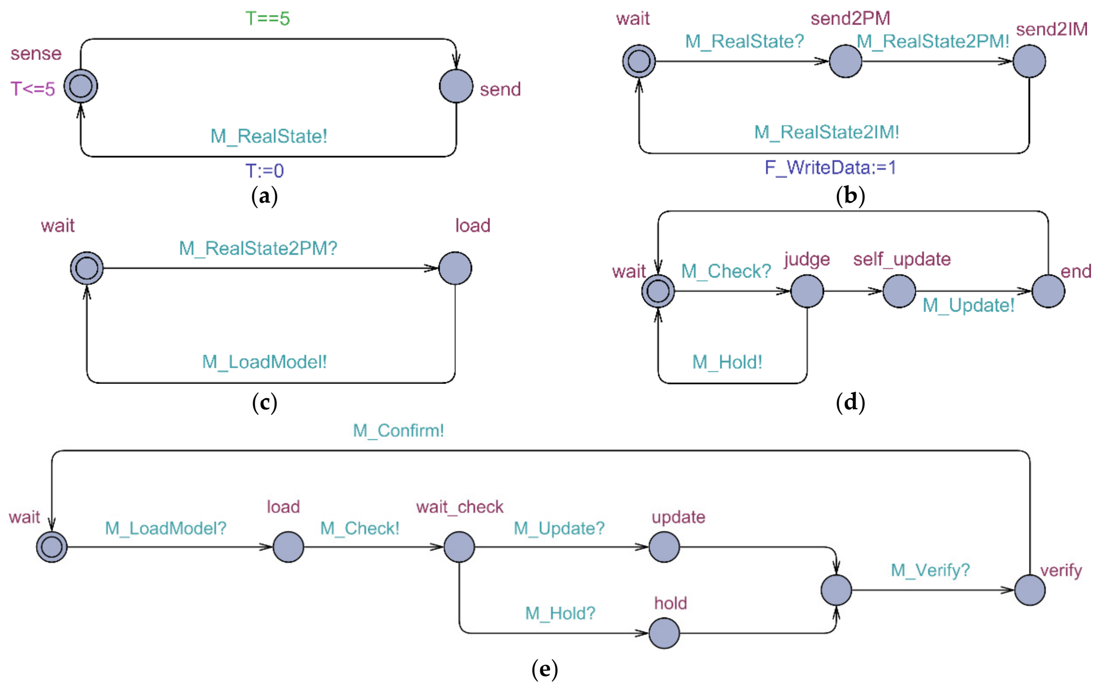


Figure 6. *Cont.*

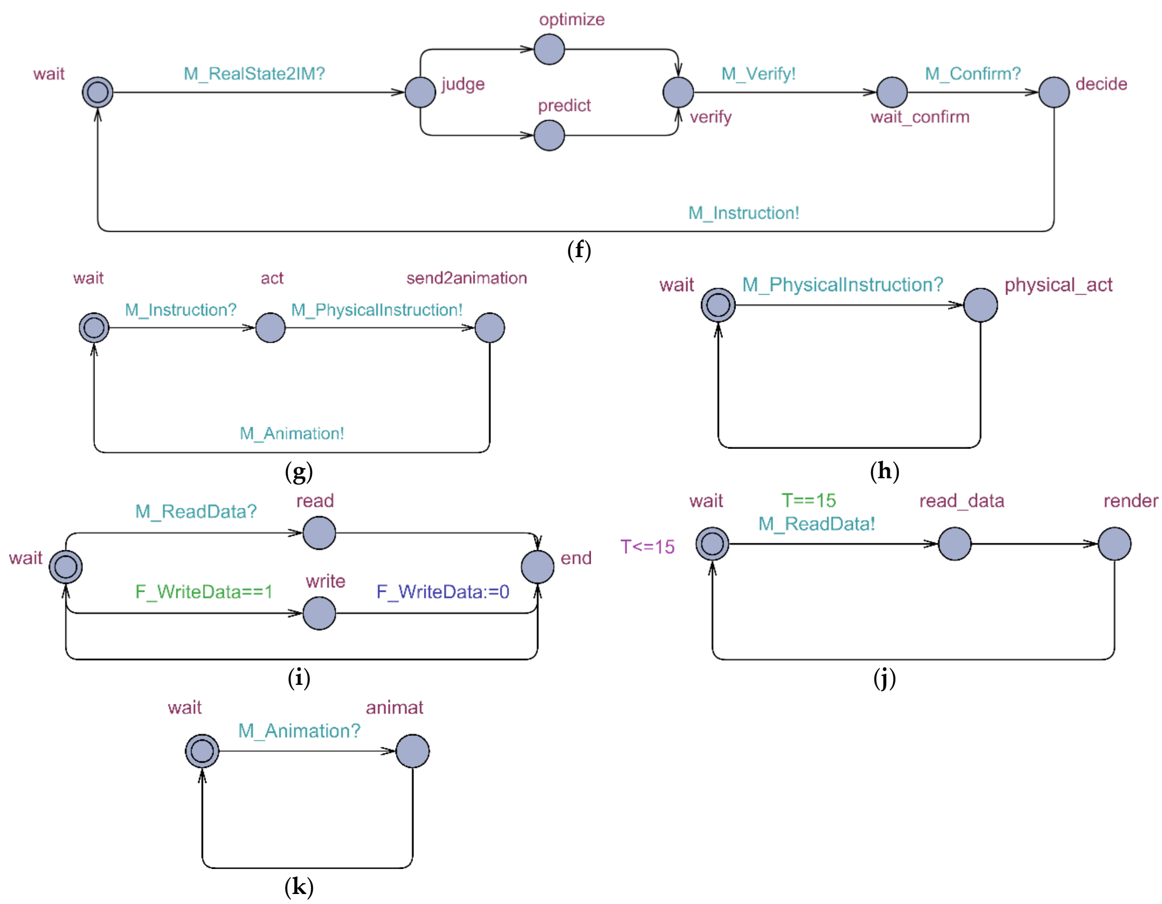


Figure 6. Time automata network of entities in the BDT foundation model. (**a**) is the time automata of PhysicalSensor; (**b**) is the time automata of BDTSensor; (**c**) is the time automata of BDTCondition; (**d**) is the time automata of BDTEvolver; (**e**) is the time automata of BDTSimulator; (**f**) is the time automata of BDTController; (**g**) is the time automata of BDTActuator; (**h**) is the time automata of PhysicalActuator; (**i**) is the time automata of BDTRecorder; (**j**) is the time automata of BDTVisualizer; (**k**) is the time automata of BDTAnimation.

BDTActuator enters the state of act after receiving control instructions, and PhysicalActuator, bound to it, enters the state of physical_act through the message channel M_PhysicalInstruction. BDTAnimation enters the state of animat through the message channel M_ Animation, presenting the real action changes brought by the control process.

It should be noted that, due to the relative stability of static information, the 3D model in the static information model is less frequently updated after the initial rendering, but, to illustrate this action, it is assumed in BDTVisualizer that the model information is read and rendered every fifteen units of time via the message channel M_ReadData. A demonstration of this time automata network execution process provided by UPPAAL is shown in Figure 7.

By comparing the sequence diagram in Figure 5 and the state switching process of the time automata in UPPAAL, it is demonstrated that the established time automata network provides an accurate description of both the static structure and dynamic activities of the foundation model for BDTs. The formal verification based on this time automata network is credible.

3.4.2. Verification Process and Analysis of Results

The established time automata network describes the foundation model for BDTs in a formal way, which, in turn, allows a rigorous formal verification of the model to demonstrate the deadlock-free accessibility and consistency of the foundation model.

UPPAAL supports the use of computational tree logic (CTL) to write formulas to load into the model verification engine to verify these properties. The CTL formulation rules are as follows [48], and the CTL statements used to verify the properties of the foundation model are shown in Table 1.

- A[ ]p: for all routes, p always holds;
- A< >p: for all routes, p finally holds;
- E[ ]p: there exists a route, p always holds;
- E< >p: there exists a route, p finally holds;
- p imply q: if p holds, q must eventually hold.

Table 1. CTL statements and meanings for verifying the properties of the foundation model.

Properties	CTL Statement	Meaning
Deadlock free	A[ ]not deadlock	There is no deadlock inside the model.
Accessibility	E< >Process_Controller.decide	There exists a route, Process_Controller finally in the state of decide.
	E< >Process_P_Actuator.physical_act	There exists a route, Process_P_Actuator finally in the state of physical_act.
	E< >Process_Simulator.verify	There exists a route, Process_Simulator finally in the state of verification.
	E< >Process_Evolver.self_update	There exists a route, Process_Evolver finally in the state of self_update.
	E< >Process_Visualizer.render	There exists a route, Process_Visualizer finally in the state of render.
Consistency	E[ ](Process_Simulator.update imply Process_Evolver.self_update)	There exists a route. When Process_Simulator is in the state of update, Process_Evolver must satisfy the state of self_update.
	E[ ](Process_Controller.wait_confirm imply Process_Simulator.verify)	There exists a route. When Process_Controller is in the wait_confirm state, Process_Simulator must satisfy the state of verify.
	E[ ](Process_Actuator.act imply Process_Controller.decide)	There exists a route. When Process_Actuator is in the state of act, Process_Controller must satisfy the state of decide.
	E[ ](Process_P_Actuator.physical_act imply Process_Actuator.act)	There exists a route. When Process_P_Actuator is in the state of physical_act, Process_Actuator must satisfy the state of act.

Loading the CTL formulas into the UPPAAL validation engine obtains the result shown in Figure 8.

The formal validation results of UPPAAL show that the entities in the foundation model do not deadlock during the designed activities and that the model always satisfies the requirements to achieve the visual presentation of the static information model, self-evolution of the physical model, intelligent decision making of the interaction model and, finally, feedback to the physical world, i.e., the physical_act state of the time automata of PhysicalActuator. The restricted relationships between the entities are also verified, where the decision making of the interaction is always based on the physical model to provide validation support and ensure that the decision is credible before feeding back to the physical world; the actuator in the interaction model can smoothly execute the controller's instructions, and the actuator in the physical world can always complete the adjustment based on the corresponding BDTActuator. Building a BDT system based on the proposed foundation model can effectively complete a realistic mapping of the physical world, thus, realizing a two-way data exchange with the physical building, optimizing the physical building state and assisting operation and maintenance decisions.

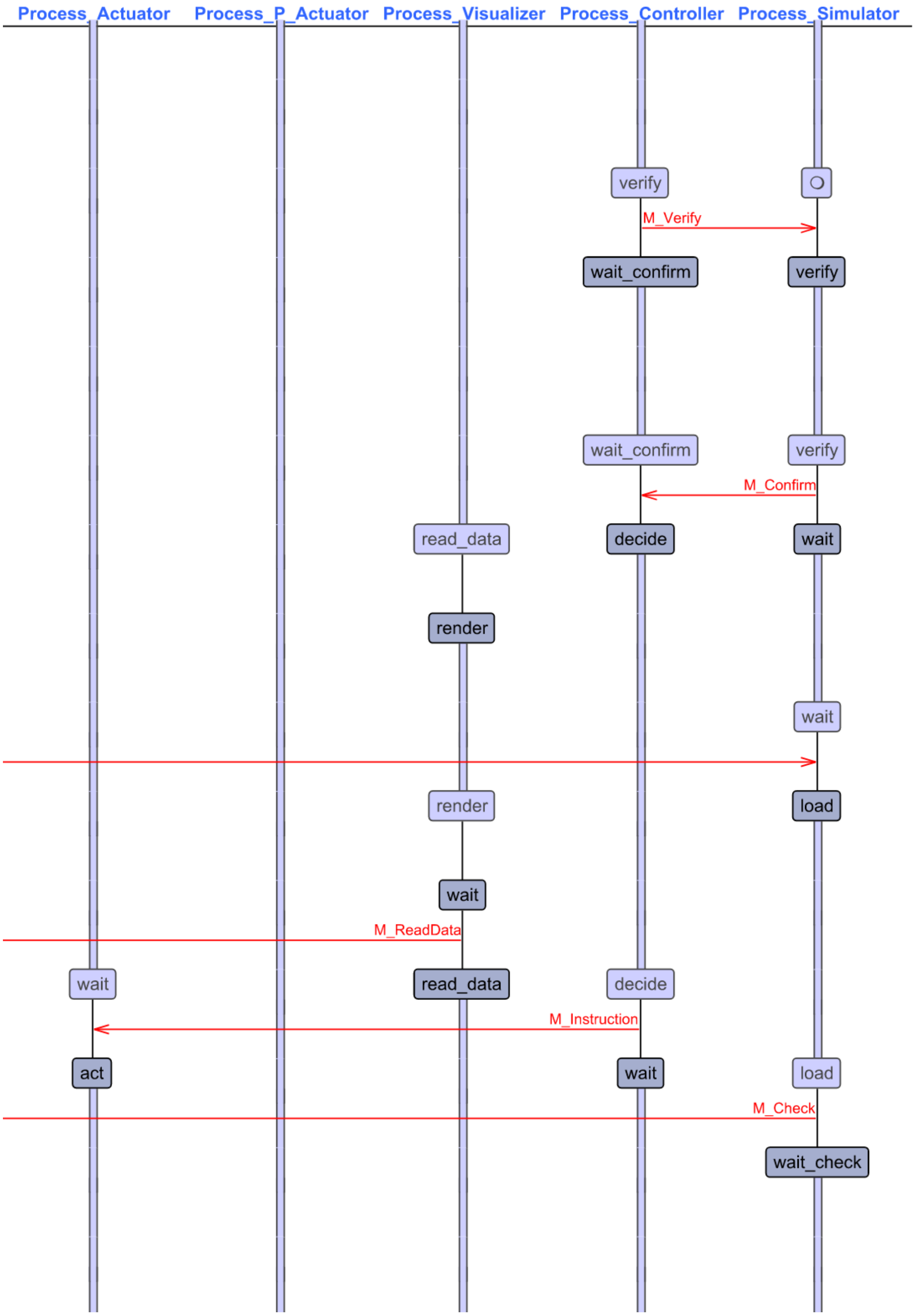


Figure 7. Execution process of the time automata network (part).

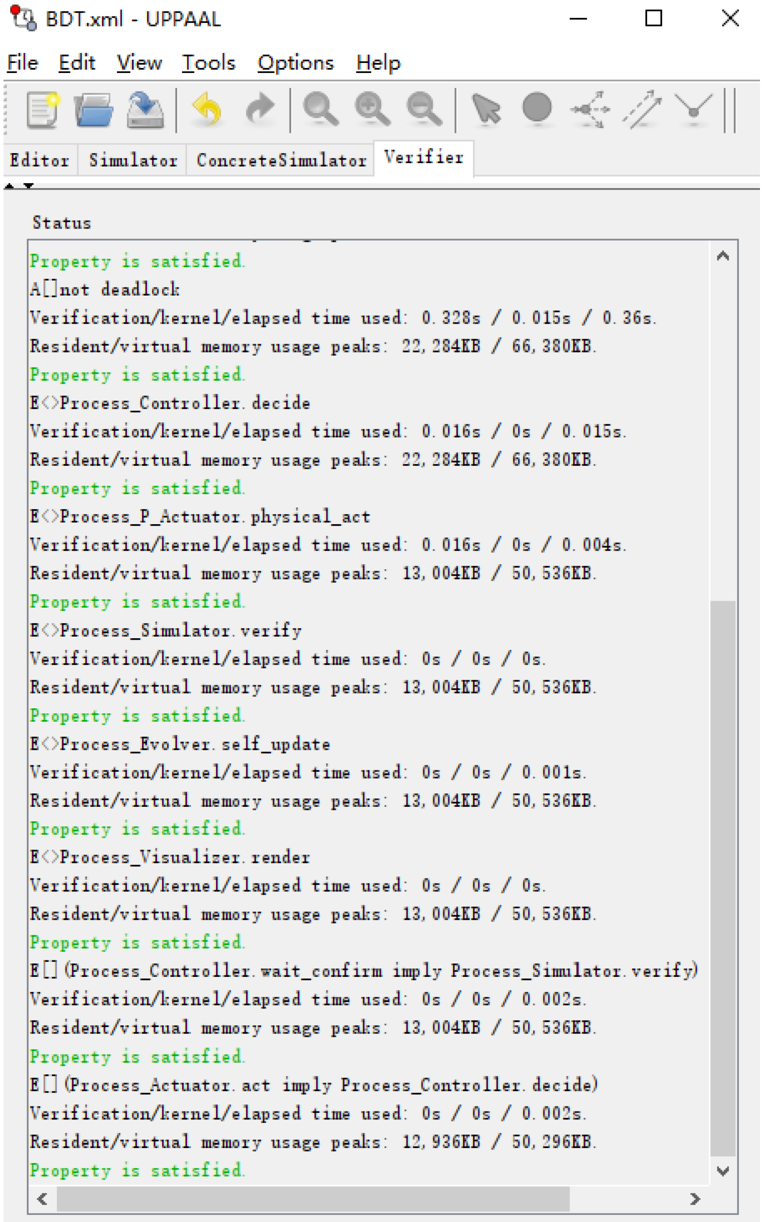


Figure 8. Results of property verification of the time automata network.

4. Case Study

To verify the effectiveness of the proposed foundation model, we chose the chiller, which is one of the key facilities in intelligent buildings, to build the BDT application. The chiller DT aims to realize the monitoring of the real-time working status of the chiller and to optimize the set values of the chiller and improve energy efficiency based on the physical model of the chiller cooling mechanism and intelligent algorithms.

In general, the energy conversion efficiency of a chiller is measured by the coefficient of performance (COP). A larger COP means more energy savings under the same conditions. The physical interpretation of the COP is as follows [49]:

$$\mathrm{COP} = Q_e / P_{chiller} \tag{5}$$

Here, Q_e is the cooling capacity of the chiller, and $P_{chiller}$ is the power of the chiller; that is, the COP represents the proportion of the electrical energy consumed by the chiller that is used for refrigeration. Adjusting various parameters of the chiller to meet the requirements of a production or living environment while pursuing the maximization of the COP has important practical significance.

4.1. Modeling the Chiller DT with the Metamodel

Utilizing the metamodel of BDTs, an instance model that assists software engineers in developing concrete software was built. The physical process of "outlet temperature regulation of the chilled water" is illustrated in Figure 9 as an example.

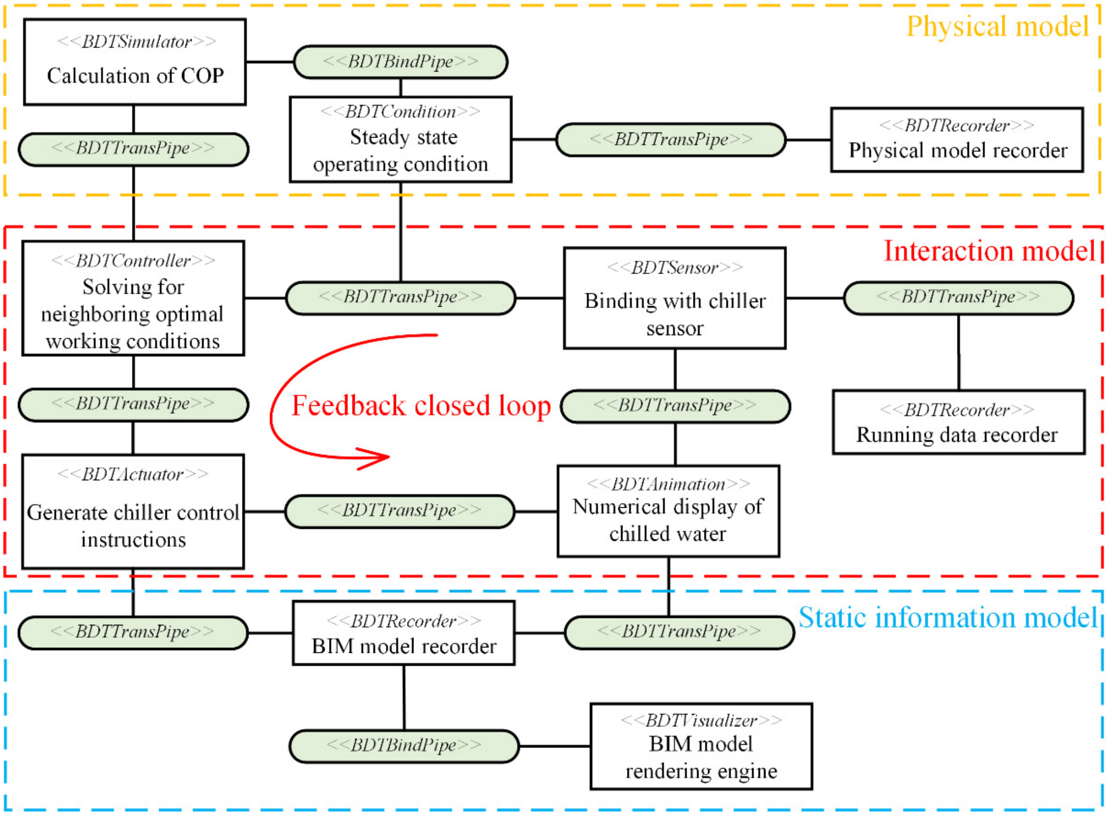


Figure 9. Process model for optimizing chiller COP.

According to the composition of the foundation model, the model of chiller DT can also be viewed as a coupling of the static information model, the physical model and the interaction model. When building a DT of a chiller, the BIM model file of the chiller is obtained from the BDTRecorder in the static information model, and the graphical model is processed by the BIM model rendering engine to complete the visual presentation.

After the BDTSensor bound to the physical sensor obtains the running data of the chiller, they are sent to the BDTController responsible for the control of the chilled water outlet temperature and the BDTCondition that judges the running context of the physical chiller. The BDTController determines whether the physical chiller setting parameters need to be adjusted. The BDTCondition loads the physical model that best matches the current physical characteristics of the chiller. If an adjustment is needed, BDTController uses the BDTSimulator in the physical model to verify whether the control instructions can achieve effects. After the control instruction is generated and verified, it is passed to the BDTActuator bound to the physical actuator to drive the physical chiller to adjust the running parameters.

In this dynamic process, BDTAnimation is responsible for displaying the running data obtained by BDTSensor and the control data of BDTActuator, and BDTRecorder, which stores the historical data of chiller running, also records the data passed in by BDTSensor.

4.2. Software Implementation of the Chiller DT

The chiller DT was developed in the Browser/Server architecture, and users obtain the services through the browser, thus, reducing the complexity of software installation. The implementation framework of the software is shown in Figure 10.

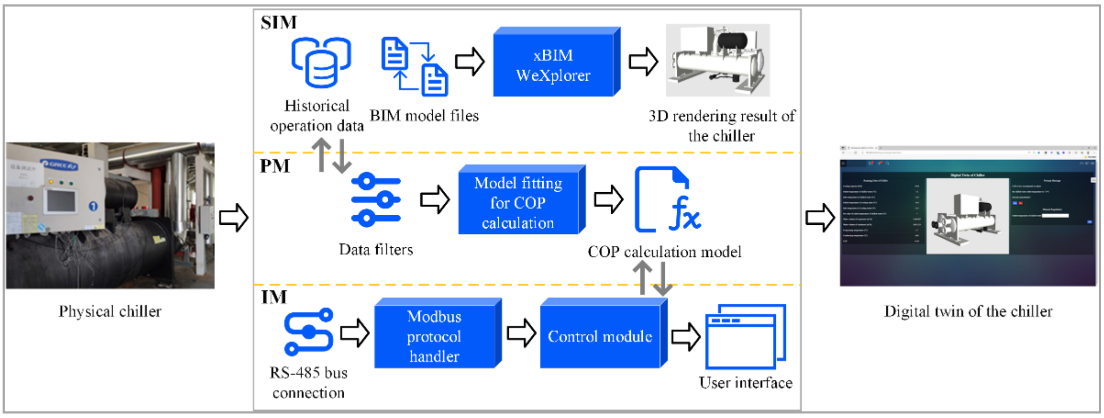


Figure 10. Implementation framework of the chiller DT.

The foundation model was proposed to drive the development of the BDTs. Therefore, the various elements in Figure 10 are the technical implementations of the components in the foundation model. In the static information model, BDTRecorder consists of a database for storing the chiller BIM model and historical operation data. BDTVisualizer is implemented by an engine capable of rendering the BIM models. In the physical model, data filters are applied to BDTCondition to limit the scenarios for which the model fitted by BDTSimulator is applicable. BDTSimulator obtained the COP prediction model by fitting the chiller historical running data. The prediction of COP under different operating conditions is used to provide validation support for the decision making of the interaction model. In the interaction model, BDTSensor and BDTActuator are integrated together to complete the parsing and encapsulation of data when the physical chiller and the chiller DT establish a connection. BDTController is composed of control algorithms programmed into the software. Based on the real-time data of the physical chiller perceived by BDT-Sensor, the operation settings for higher COP are calculated. BDTAnimation describes the dynamic behaviors of the chiller DT and is consequently presented visually as part of the user interface.

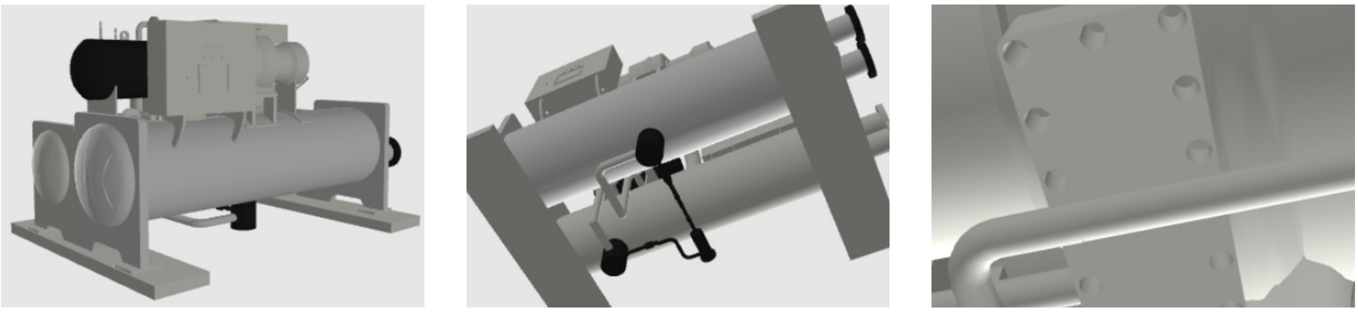
4.2.1. Visualization of the Static Information Model

BIM produces a standardized, machine-readable information model [3] and is considered a promising way to transfer appropriate data between different software entities and stakeholders. Therefore, in this paper, a BIM model of the chiller was imported; this approach not only allowed existing models to be reused but also simplified the development process while maintaining consistency of the underlying technologies to facilitate expansion to more complex BDT systems in the future.

During the development of BIM, to effectively integrate and store model data, the International Alliance for Interoperability (IAI) proposed the Industry Foundation Class (IFC) data standard in 1997, which enables collaborative exchange of all BM information based on an open standard and procedure. After six major revisions, IFC2 × 3 is now the version supported by most BIM software.

Since the BIM model of the chiller is loaded on web page, a Web Graphics Library (WebGL) plugin with IFC data parsing capability should be selected as the rendering engine. The existing open-source projects for parsing IFC mainly include: BiMserver, BIM surfer, IfcOpenShell, IfcPlusPlus, FreeCAD, xBIM Toolkit, etc. Among them, BIM Surfer and xBIM Toolkit are the ones supporting web parsing. In the interest of balancing development cost and complexity of operation, xBIM WeXplorer (version 1.0.4) in xBIM Toolkit was chosen as the rendering engine.

xBIM WeXplorer supports parsing files with the suffix ".wexBIM", which are converted from files with the suffix ".ifc". Compared with the latter, wexBIM files enhance the web rendering capability and improve the rendering speed. Figure 11 shows the effects of the rendering.



(**a**) (**b**) (**c**)

Figure 11. Implementation effect of the static information model. (**a**) is the overview of the model; (**b**) is the rotation and translation of the model; (**c**) is a segment detail of the model.

The implementation of the static information model provides practitioners with a way to view the static characteristics of the chiller. The 3D model shows the chiller construction more clearly. Information such as size and material of specific components is also presented after selection.

4.2.2. Model Fitting of the Physical Model

Since the aim of this section was to effectively implement the BDTs based on the proposed foundation model, an in-depth discussion of the COP prediction method is not presented in the implementation of the physical model of the chiller DT. Referring to the research efforts of Zhao et al. [50], their method of constructing a multi-input-single-output regression model was applied to the implementation of the physical model. The COP prediction model was constructed with the chilled water temperature difference (ΔT_{chw}) and the cooling water temperature difference (ΔT_{cw}). Running data over two weeks were used for model fitting. The COP model was developed in Origin (version 2018), and the output of the nonlinear surface fitting is shown in Figure 12.

It is obvious that most of the points are near the surface, and the correlation coefficient (R^2) is 0.99785, which also indicates that the predicted COP values fitted well with the actual value. According to the estimate of coefficients, the COP prediction model can be expressed as Equation (6).

$$\mathrm{COP} = 13.29597 - 2.60483 * \Delta T_{chw} - 0.90367 * \Delta T_{cw} \tag{6}$$

As an implementation of the physical model, the COP prediction model was coded in the software project through Python to provide validation support for the interaction model of the chiller DT.

4.2.3. Construction of Bi-Directional Data Flow for the Interaction Model

The key to the implementation of the interaction model is to establish a closed loop of feedback control. The physical chiller in this study was pre-assembled with sensors. The running data of the chiller were gathered in the attached control box, as shown in Figure 13a. Two RS-485 bus lines originating from the control box were connected to a serial converter, as shown in Figure 13b, and, after conversion to a USB interface, a further connection to the chiller DT was established, as shown in Figure 13c. As the physical chiller followed the Modbus protocol, with the help of a library named "modbus_tk" in

Python, the real-time operation data of the physical chiller and the instructions from the chiller DT were conveniently parsed and packaged. The physical connection and the logical connection between the physical chiller and the chiller DT were established.

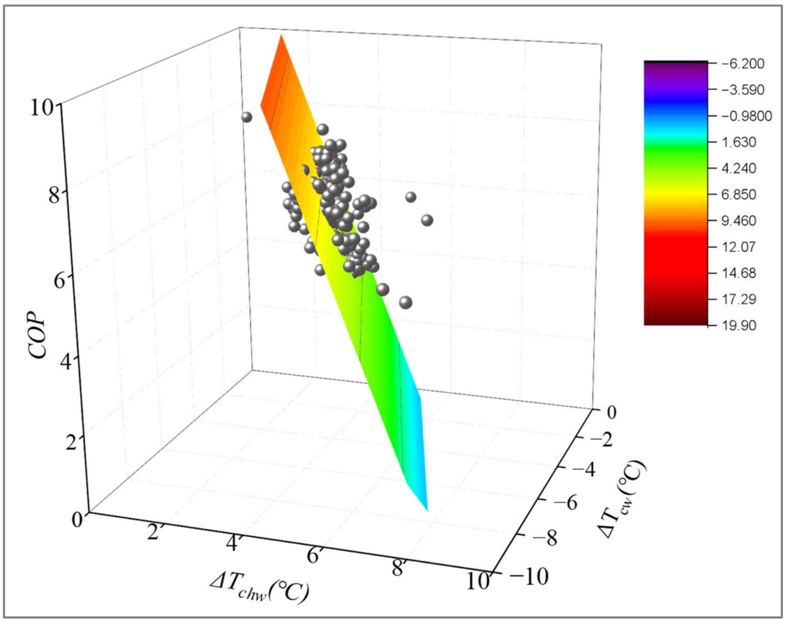


Figure 12. Output of COP prediction model fitting.

(**a**) (**b**) (**c**)

Figure 13. Connection between the physical chiller and the chiller DT. (**a**) is the physical chiller of the experiment; (**b**) is the connection of the chiller control box and the computer; (**c**) is the interface converter between the physical chiller and the computer.

Virtual–real connection provides a bi-directional transmission channel for information sensing and control feedback. In general, the working environment of a chiller does not suffer sudden and drastic changes, and sharp adjustments of the operating parameters triggers the self-protection mechanism of the chiller. Therefore, the optimization strategies for the chiller were gradual and moderate. The pseudo-code of the control strategies is shown in Algorithm 1.

The above algorithm takes the current operating conditions of the chiller as a starting point and searches for better operating conditions from the neighbor that can improve the COP [50]. This allows each adjustment of the chiller to be made in smaller magnitudes, reducing the disruption to the workspace cooled by the chiller. The physical model supports the prediction of COP values. Without disturbing the normal operation of the physical chiller, a judgment can be made on whether the COP can be improved by the neighboring working conditions. The optimal neighboring conditions are encapsulated as control commands and sent to the physical chiller through the connection established between the chiller DT and the physical chiller, realizing a closed loop of "sense–control–execute".

Besides the judgments and decisions made by the DT of chiller with automation, the operation interface is also designed to allow users to realize a human-in-the-loop control process.

Algorithm 1: Optimal Control Algorithm of the Interaction Model.

Input: actual COP, ΔT_{chw}, ΔT_{cw}, σ_{chw}, σ_{cw} (Step length of single adjustment of ΔT_{chw} and ΔT_{cw}.)
Output: Optimal ΔT_{chw}, ΔT_{cw} for adjustment
Content:

1. **while** countdown = 0 **do**
2. **check** adjustment status
3. **if** status = locked **do**
4. **exit** adjustment
5. **else**
6. **set** status = locked
7. set array = $\{ \{\Delta T_{chw} + \sigma_{chw}, \Delta T_{cw}\}, \{\Delta T_{chw} - \sigma_{chw}, \Delta T_{cw}\}, \{\Delta T_{chw} + \sigma_{chw}, \Delta T_{cw} + \sigma_{cw}\}, \{\Delta T_{chw} - \sigma_{chw}, \Delta T_{cw} + \sigma_{cw}\}, \{\Delta T_{chw}, \Delta T_{cw} + \sigma_{cw}\}, \{\Delta T_{chw}, \Delta T_{cw} - \sigma_{cw}\}, \{\Delta T_{chw} + \sigma_{chw}, \Delta T_{cw} - \sigma_{cw}\}, \{\Delta T_{chw} - \sigma_{chw}, \Delta T_{cw} + \sigma_{cw}\} \}$
8. **set** COP prediction array = {}
9. **for** array **do**
10. **set** temp = calculate COP prediction value
11. **add** temp into COP prediction array
12. **end for**
13. **set** max = select the maximum value in the COP prediction array
14. **if** max > actual COP **do**
15. encapsulate control instructions to adjust the physical chiller
16. **else**
17. **exit** adjustment
18. **set** status = unlocked
19. **set** countdown = 20 min
20. **end while**

4.3. Performance of the Chiller DT

After gradually completing the implementation of the static information model, the physical model and the interaction model, the chiller DT was obtained, as shown in Figure 14.

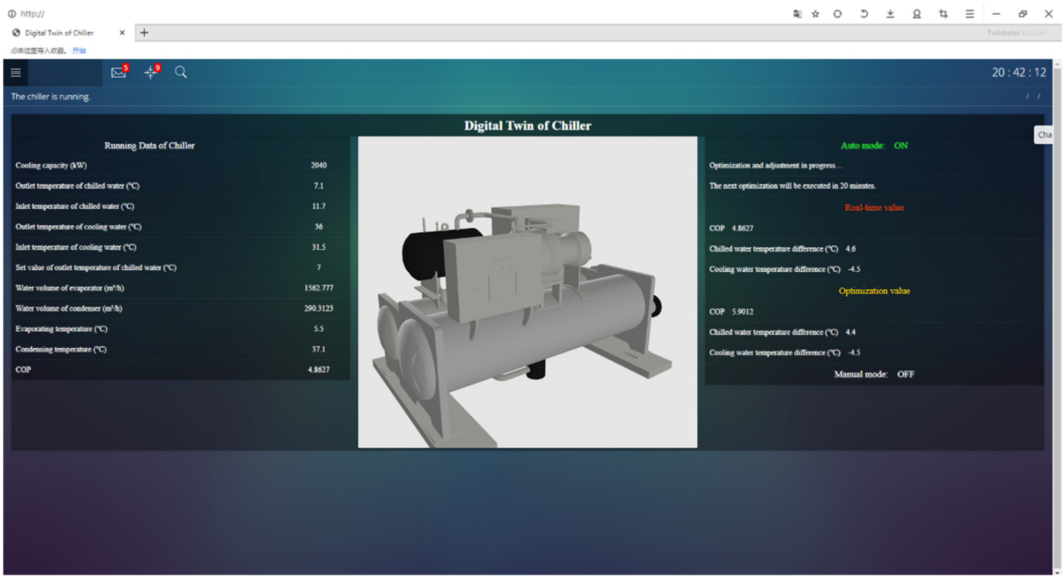


Figure 14. Running page of the chiller DT.

It is divided into three main functional areas. The left shows the real-time operation data of the physical chiller, and the middle is the 3D rendering of the physical chiller. On the right is the control area, when the chiller DT judges the neighboring working conditions more energy efficient, a prompt message appears in the control area to provide the user with an operation that can be carried out. A manual control area is also available to allow the user to control the chiller when necessary.

Four days of running data were used to analyze the control effectiveness of the chiller DT. The physical chiller was disconnected from the chiller DT on the first two days, and they interacted with each other on the next two days. The actual values of COP are shown in Figure 15.

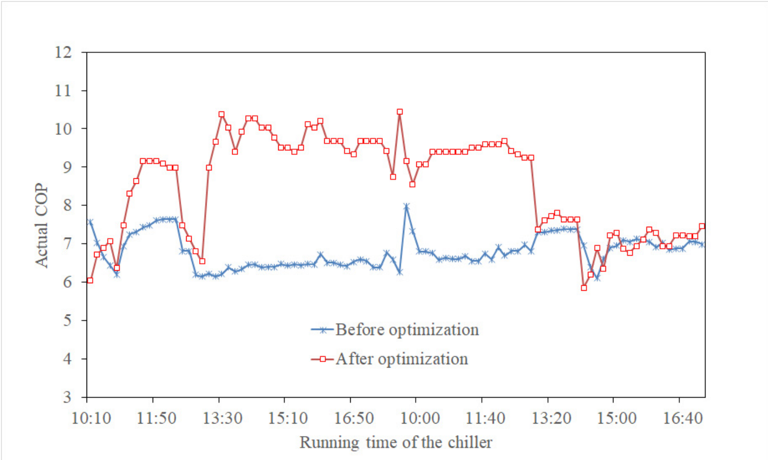


Figure 15. Optimal control effect of the chiller DT on the physical chiller.

The climatic conditions were similar over the four days, so the trend of the COP is similar in both data sets. However, based on the optimal control from the chiller DT, the values of the COP were higher on the last two days than on the first two days most of the time. The improvement of COP achieved by the chiller DT is more clearly illustrated in Figure 16.

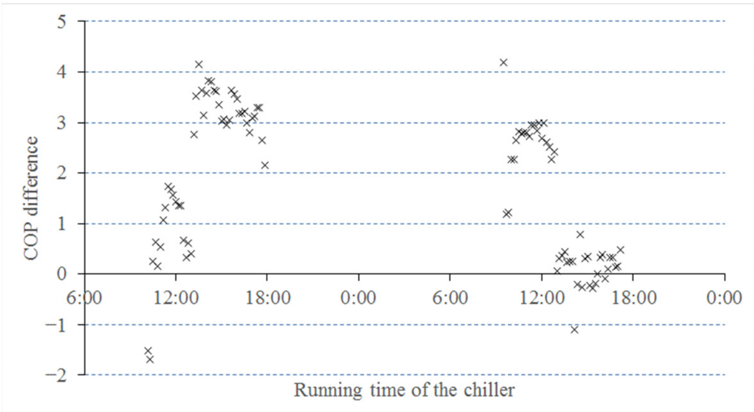


Figure 16. COP difference from optimal control of the physical chiller.

Among the 94 time points of data, COP was improved in 85 collection points, accounting for 90.43%. In particular, there were 48 data points with COP differences greater than 2, accounting for 51.06%. The chiller DT shows great potential for improving the efficiency of physical chiller cooling and saving power. Moreover, based on the chiller DT, workers are able to obtain real-time operating data in a remote and timely manner, allowing them to monitor the status of the chiller while avoiding potential harm to their health caused by the noisy and high-temperature environment of the physical chiller.

5. Discussion

With the ongoing development of information technology, DTs are becoming a key technology for realizing intelligent activities in the building lifecycle. Although there is a lack of a uniform definition, it is significant to discuss the fundamental characteristics of BDTs. For researchers and practitioners, this helps to provide a perspective for understanding BDTs and a reference framework for the implementation of BDTs.

The proposed foundation model is an innovative new generation of building information model that fully considers the requirements of information mapping, physical simulation and dynamic interaction. The foundation model is not only a digital version of the physical building, thus, reaping the benefits of digitization in storage, presentation and information transfer. The added value is that the coupling of submodels in the foundation model realizes intelligent feedback from the BDTs to the corresponding physical buildings. This is one of the major reasons why DT technologies is valued and widely used in the manufacturing and aerospace industries. In the submodels, the static information model and the physical model assure that the BDTs are consistent with the physical buildings, and the interaction model guarantees that this consistency is not weakened over time. Such reliable consistency makes automated or manual decisions made based on the BDTs credible and helpful. As more comprehensive insights into the BDTs, the triad structure composed of the static information model, the physical model and the interaction model may also be extended. We will continue to improve the foundation model proposed in this paper and add more detailed attributes to the components of the model.

Models can help researchers to block out irrelevant information and concentrate on the essential characteristics, so we picked the foundation model as the starting point for the investigation of the BDTs. In terms of the development of BDT technologies, the proposed BDT theories should contribute to the implementation of specific BDT applications. As shown in previous research works, most BDT applications are implemented in the form of software systems. In the field of software engineering, the model-driven approach to software development has been found to reduce duplication of effort and improve efficiency when developing complex systems. Our research was initiated from the foundation model for BDTs, and gradually specified this model to derive more concrete and domain-specific model components so as to obtain a unified and scalable model description method. The metamodel of BDTs is another contribution of this paper. A unified model expression allows the model to support deconstruction and reuse and improve the efficiency of model construction. It also eliminates the information confusion of the same model in different application scenarios and maintains the accuracy of the information transferred by the model. Further, model files with a standardized format provide the basis for automatic computer parsing.

Future work includes expanding the metamodel into a syntactically accurate and semantically complete domain modeling language, investigating the automatic transformation from BDT models to programming languages and developing supplementary modeling tools to support the use of the modeling language. This will accelerate the development of BDT software and promote the deployment of DT technologies in the building sector.

6. Conclusions

To conclude, this paper proposed a new approach to integrating DTs with intelligent buildings by developing a foundation model for BDTs which includes a static information model, a physical model and an interaction model for mapping a physical building into the virtual world. In addition, the characteristics of the submodels and the collaboration between them were fully discussed, and the dynamic process of real-time sensing of the physical building operation status and feedback control by BDTs was clearly described. Based on the metamodel of BDTs, not only does this provide a platform-independent and unified modeling method for the description of the BDT model, but it also uses a formal approach to rigorously verify that the foundation model is deadlock free and that the individual states designed are reachable and consistent with each other. Finally, a case study of a chiller DT was presented to demonstrate an application for real-time operation and maintenance, and an analysis of the modeling concept and method provided further validation of the effectiveness of the foundation model. Moreover, this case study can also serve as a reference for stakeholders who are interested in improving the intelligence of building operations and maintenance.

In summary, this paper was limited to explaining the constituents and properties of the foundation model, with less discussion of the implementation technologies for each part. This means the paper remained on a theoretical level. The DT was concomitantly symbiotic with the corresponding physical entities in the lifecycle, while dynamic evolution of the foundation model was not sufficiently discussed. Finally, the twin object used in the case study was relatively simple. For future work, we will try to establish a more complex system to validate the foundation model and modeling methods for BDTs.

Author Contributions: Conceptualization, Q.Y. and S.L.; methodology, Q.Y.; software, S.L.; validation, J.X. and R.Z.; data curation, W.C.; writing—original draft preparation, S.L.; writing—review and editing, Q.Y. and W.C.; visualization, S.L. and R.Z.; supervision, J.X. and Q.Y.; funding acquisition, Q.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 52178307, and the Natural Science Foundation of Jiangsu Province, grant number BK20201335.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

References

1. Ding, L.Y.; Zhou, Y.; Akinci, B. Building Information Modeling (BIM) application framework: The process of expanding from 3D to computable nD. *Autom. Constr.* **2014**, *46*, 82–93. [CrossRef]
2. Charef, R.; Alaka, H.; Emmitt, S. Beyond the third dimension of BIM: A systematic review of literature and assessment of professional views. *J. Build. Eng.* **2018**, *19*, 242–257. [CrossRef]
3. Gao, X.H.; Pishdad-Bozorgi, P. BIM-enabled facilities operation and maintenance: A review. *Adv. Eng. Inform.* **2019**, *39*, 227–247. [CrossRef]
4. Quinn, C.; Shabestari, A.Z.; Misic, T.; Gilani, S.; Litoiu, M.; McArthur, J.J. Building automation system-BIM integration using a linked data structure. *Autom. Constr.* **2020**, *118*, 16. [CrossRef]
5. Mishra, S.; Glaws, A.; Cutler, D.; Frank, S.; Azam, M.; Mohammadi, F.; Venne, J.S. Unified architecture for data-driven metadata tagging of building automation systems. *Autom. Constr.* **2020**, *120*, 14. [CrossRef]
6. Terreno, S.; Asadi, S.; Anumba, C. An Exploration of Synergies between Lean Concepts and BIM in FM: A Review and Directions for Future Research. *Buildings* **2019**, *9*, 147. [CrossRef]
7. Sacks, R.; Brilakis, I.; Pikas, E.; Xie, H.S.; Girolami, M. Construction with digital twin information systems. *Data-Cent. Eng.* **2020**, *1*, e14. [CrossRef]
8. Opoku, D.J.; Perera, S.; Osei-Kyei, R.; Rashidi, M. Digital twin application in the construction industry: A literature review. *J. Build. Eng.* **2021**, *40*, 15. [CrossRef]
9. Ali, K.N.; Alhajlah, H.H.; Kassem, M.A. Collaboration and Risk in Building Information Modelling (BIM): A Systematic Literature Review. *Buildings* **2022**, *12*, 571. [CrossRef]
10. Al Dakheel, J.; Del Pero, C.; Aste, N.; Leonforte, F. Smart buildings features and key performance indicators: A review. *Sustain. Cities Soc.* **2020**, *61*, 19. [CrossRef]
11. Vieira, R.; Carreira, P.; Domingues, P.; Costa, A.A. Supporting building automation systems in BIM/IFC: Reviewing the existing information gap. *Eng. Constr. Archit. Manag.* **2020**, *27*, 1357–1375. [CrossRef]
12. O'Grady, T.; Chong, H.Y.; Morrison, G.M. A systematic review and meta-analysis of building automation systems. *Build. Environ.* **2021**, *195*, 13. [CrossRef]
13. Tang, S.; Shelden, D.R.; Eastman, C.M.; Pishdad-Bozorgi, P.; Gao, X.H. BIM assisted Building Automation System information exchange using BACnet and IFC. *Autom. Constr.* **2020**, *110*, 14. [CrossRef]
14. Barricelli, B.R.; Casiraghi, E.; Fogli, D. A Survey on Digital Twin: Definitions, Characteristics, Applications, and Design Implications. *IEEE Access* **2019**, *7*, 167653–167671. [CrossRef]
15. Grieves, M.; Vickers, J. Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. In *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*, 1st ed.; Kahlen, F.J., Flumerfelt, S., Alves, A., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 85–113.

16. Laaki, H.; Miche, Y.; Tammi, K. Prototyping a Digital Twin for Real Time Remote Control over Mobile Networks: Application of Remote Surgery. *IEEE Access* **2019**, *7*, 20325–20336. [CrossRef]
17. Zheng, Y.; Yang, S.; Cheng, H.C. An application framework of digital twin and its case study. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 1141–1153. [CrossRef]
18. Qiao, Q.Z.; Wang, J.J.; Ye, L.K.; Gao, R.X. Digital Twin for Machining Tool Condition Prediction. In Proceedings of the 52nd CIRP Conference on Manufacturing Systems (CMS), Ljubljana, Slovenia, 12–14 June 2019; pp. 1388–1393.
19. Lu, Q.; Parlikad, A.K.; Woodall, P.; Ranasinghe, G.D.; Xie, X.; Liang, Z.; Konstantinou, E.; Heaton, J.; Schooling, J. Developing a Digital Twin at Building and City Levels: Case Study of West Cambridge Campus. *J. Manag. Eng.* **2020**, *36*, 05020004. [CrossRef]
20. Wei, Y.; Lei, Z.; Altaf, S. An Off-Site Construction Digital Twin Assessment Framework Using Wood Panelized Construction as a Case Study. *Buildings* **2022**, *12*, 566. [CrossRef]
21. Mannino, A.; Moretti, N.; Dejaco, M.; Luciano, B.; Re Cecconi, F. Office building occupancy monitoring through image recognition sensors. *Int. J. Saf. Secur. Eng.* **2019**, *9*, 371–380. [CrossRef]
22. Lu, Q.C.; Xie, X.; Parlikad, A.K.; Schooling, J.M. Digital twin-enabled anomaly detection for built asset monitoring in operation and maintenance. *Autom. Constr.* **2020**, *118*, 16. [CrossRef]
23. Zhao, J.F.; Feng, H.B.; Chen, Q.; de Soto, B.G. Developing a conceptual framework for the application of digital twin technologies to revamp building operation and maintenance processes. *J. Build. Eng.* **2022**, *49*, 12. [CrossRef]
24. Tao, F.; Zhang, M. Digital Twin Shop-Floor: A New Shop-Floor Paradigm towards Smart Manufacturing. *IEEE Access* **2017**, *5*, 20418–20427. [CrossRef]
25. Lu, Y.Q.; Liu, C.; Wang, K.I.K.; Huang, H.Y.; Xu, X. Digital Twin-driven smart manufacturing: Connotation, reference model, applications and research issues. *Robot. Comput. Integr. Manuf.* **2020**, *61*, 14. [CrossRef]
26. Tuegel, E.J.; Ingraffea, A.R.; Eason, T.G.; Spottswood, S.M. Reengineering Aircraft Structural Life Prediction Using a Digital Twin. *Int. J. Aerosp. Eng.* **2011**, *2011*, 154798. [CrossRef]
27. Pan, Y.; Zhang, L.M. A BIM-data mining integrated digital twin framework for advanced project management. *Autom. Constr.* **2021**, *124*, 15. [CrossRef]
28. Boje, C.; Guerriero, A.; Kubicki, S.; Rezgui, Y. Towards a semantic Construction Digital Twin: Directions for future research. *Autom. Constr.* **2020**, *114*, 16. [CrossRef]
29. Lu, Q.C.; Chen, L.; Li, S.; Pitt, M. Semi-automatic geometric digital twinning for existing buildings based on images and CAD drawings. *Autom. Constr.* **2020**, *115*, 19. [CrossRef]
30. Khajavi, S.H.; Motlagh, N.H.; Jaribion, A.; Werner, L.C.; Holmstrom, J. Digital Twin: Vision, Benefits, Boundaries, and Creation for Buildings. *IEEE Access* **2019**, *7*, 147406–147419. [CrossRef]
31. Angjeliu, G.; Coronelli, D.; Cardani, G. Development of the simulation model for Digital Twin applications in historical masonry buildings: The integration between numerical and experimental reality. *Comput. Struct.* **2020**, *238*, 20. [CrossRef]
32. Kaewunruen, S.; Lian, Q. Digital twin aided sustainability-based lifecycle management for railway turnout systems. *J. Clean. Prod.* **2019**, *228*, 1537–1551. [CrossRef]
33. Pan, Y.D.; Braun, A.; Brilakis, I. Enriching geometric digital twins of buildings with small objects by fusing laser scanning and AI-based image recognition. *Autom. Constr.* **2022**, *140*, 19. [CrossRef]
34. Zhou, Y.; Wei, X.; Peng, Y. The Modelling of Digital Twins Technology in the Construction Process of Prefabricated Buildings. *Adv. Civ. Eng.* **2021**, *2021*, 11. [CrossRef]
35. Zhao, Y.H.; Wang, N.Q.; Liu, Z.S.; Mu, E.Y. Construction Theory for a Building Intelligent Operation and Maintenance System Based on Digital Twins and Machine Learning. *Buildings* **2022**, *12*, 17. [CrossRef]
36. Peng, Y.; Zhang, M.; Yu, F.Q.; Xu, J.L.; Gao, S. Digital Twin Hospital Buildings: An Exemplary Case Study through Continuous Lifecycle Integration. *Adv. Civ. Eng.* **2020**, *2020*, 13. [CrossRef]
37. Liu, Z.S.; Shi, G.L.; Meng, X.L.; Sun, Z. Intelligent Control of Building Operation and Maintenance Processes Based on Global Navigation Satellite System and Digital Twins. *Remote Sens.* **2022**, *14*, 20. [CrossRef]
38. Begić, H.; Galić, M. A Systematic Review of Construction 4.0 in the Context of the BIM 4.0 Premise. *Buildings* **2021**, *11*, 337. [CrossRef]
39. Caixeta, M.; Fabricio, M.M. Physical-digital model for co-design in healthcare buildings. *J. Build. Eng.* **2021**, *34*, 13. [CrossRef]
40. Saad, A.E.H.; Dong, Z.; Karimi, M. A Comparative Study on Recently-Introduced Nature-Based Global Optimization Methods in Complex Mechanical System Design. *Algorithms* **2017**, *10*, 120. [CrossRef]
41. David, I.; Aslam, K.; Faridmoayer, S.; Malavolta, I.; Syriani, E.; Lago, P.; Soc, I.C. Collaborative Model-Driven Software Engineering: A Systematic Update. In Proceedings of the 24th International ACM/IEEE Conference on Model-Driven Engineering Languages and Systems (MODELS), Fukuoka, Japan, 10–15 October 2021; pp. 273–284.
42. do Amaral, J.V.S.; Montevechi, J.A.B.; Miranda, R.D.; de Sousa, W.T. Metamodel-based simulation optimization: A systematic literature review. *Simul. Model. Pract. Theory* **2022**, *114*, 21. [CrossRef]
43. Deng, X.; Zhou, B.; Sun, X.; Yang, H.; Chen, L. A method for reliability detection of automated guided vehicle based on timed automata. *Syst. Sci. Control. Eng.* **2021**, *9*, 570–579. [CrossRef]
44. Meng, Z.; Tang, T.; Wei, G.; Yuan, L. Analysis of ATO System Operation Scenarios Based on UPPAAL and the Operational Design Domain. *Electronics* **2021**, *10*, 503. [CrossRef]

45. Gong, X.L.; Ma, J.; Li, Q.C.; Zhang, J. Automatic Model Building and Verification of Embedded Software with UPPAAL. In Proceedings of the 10th IEEE Int Conf on Trust, Security and Privacy in Computing and Communications (TrustCom), Changsha, China, 16–18 November 2011; pp. 1118–1124.
46. Siavashi, F.; Iqbal, J.; Truscan, D.; Vain, J. Testing Web Services with Model-Based Mutation. In Proceedings of the 11th International Joint Conference on Software Technologies (ICSOFT-EA), Lisbon, Portugal, 24–26 July 2016; pp. 45–67.
47. Huang, L.; Liang, T.; Kang, E.Y. Tool-Supported Analysis of Dynamic and Stochastic Behaviors in Cyber-Physical Systems. In Proceedings of the 19th IEEE International Conference on Software Quality, Reliability and Security (QRS), Sofia, Bulgaria, 22–26 July 2019; pp. 228–239.
48. Kochaleema, K.H.; Santhoshkumar, G. Methodology for Integrating Computational Tree Logic Model Checking in Unified Modelling Language Artefacts: A Case Study of an Embedded Controller. *Def. Sci. J.* **2019**, *69*, 58–64. [CrossRef]
49. Alves, O.; Monteiro, E.; Brito, P.; Romano, P. Measurement and classification of energy efficiency in HVAC systems. *Energy Build.* **2016**, *130*, 408–419. [CrossRef]
50. Zhao, T.Y.; Wang, J.M.; Xu, M.; Li, K.S. An online predictive control method with the temperature based multivariable linear regression model for a typical chiller plant system. *Build. Simul.* **2020**, *13*, 335–348. [CrossRef]