machines

MDPI

Article

Information Reuse Methods for Multi-Dimensional Models in Discrete Workshops

Ruiping Luo [1,2,3] and Jiaxing Zhu [4,*]

[1] School of Mechanical and Electrical Engineering, Hubei Polytechnic University, Huangshi 435003, China; 201115@hbpu.edu
[2] Huangshi Donper Compressor Co., Ltd., Huangshi 435111, China
[3] Hubei Key Laboratory of Intelligent Convey Technology and Device, Hubei Polytechnic University, 16 Guilin Rd.(N), Huangshi 435003, China
[4] School of Mechanical and Electrical Engineering, Wuhan University of Technology, Wuhan 430070, China
* Correspondence: jiaxing_zhu@whut.edu.cn

Abstract

With the gradual development of digital twin technology from theory to practice, the importance of the efficient reuse of existing digital twin models has become increasingly prominent in order to reduce the waste of resources and additional costs caused by repeated modeling. To address the difficulty of reusing multi-dimensional model information (*MMI*) in existing digital twin models during the conversion process from geometric models to digital twin models, this paper proposes a method for reusing *MMI* in discrete workshops. First, *MMI* and its representations are defined and constructed. Subsequently, a model-matching approach is introduced to identify appropriate *MMI*s for geometric models. Following this, a reuse strategy for workshop *MMI*s is thoroughly explained. Finally, the effectiveness of the proposed method is validated through case studies in the arc-welding workshop. The accuracy of single-model matching remains consistently at 1 across all model tests, and the proposed method reduces the total number of operations by 126 (94.7%) compared to existing methods in multi-device model construction. The results show that this method can effectively organize the workshop digital twin model, compensate for the shortage of digital twin model reuse, and help engineers reuse the existing *MMI* to build a digital twin model.

Keywords: digital twin model; multi-dimensional models; reuse; matching; discrete workshops

check for updates

Academic Editor: Panagiotis Kyratsis

Received: 28 May 2025
Revised: 9 July 2025
Accepted: 16 July 2025
Published: 17 July 2025

Citation: Luo, R.; Zhu, J. Information Reuse Methods for Multi-Dimensional Models in Discrete Workshops. *Machines* **2025**, *13*, 614. https://doi.org/10.3390/machines13070614

Copyright: © 2025 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 deep integration of new-generation information technologies such as artificial intelligence, cloud computing, and the Internet of Things with advanced manufacturing technologies, a gradual realization of real-time interaction and seamless integration between the physical workshop space and virtual environment is emerging, offering robust support for the efficient optimization of manufacturing processes [1]. In this context, the utilization of digital twin technology as an effective means to facilitate such interaction has gained increasing significance in factory applications [2,3], leading to the emergence of the concept of digital twin factories [4].

The conceptual model of a digital twin workshop contains physical entities, a digital twin model, data, and services [5]. The integration of these four elements enables digital twin workshops to exhibit significant advantages in optimizing energy consumption [6],

intelligent material tracking [7], and dynamic shop scheduling [8]. However, the prerequisite for the practical application of digital twin workshops is to build a digital twin model [9]. The digital twin model encompasses geometric, physical, behavioral, and rule models, and its construction necessitates multidisciplinary knowledge, an understanding of the application scenario, and expert support [10]. With the promotion of digital twin technology in workshop scheduling [11], production line optimization [12], and production scheduling [13,14], the number of digital twin models of discrete-workshop elements and the required information exhibit a rapid growth trend. This poses challenges for companies regarding personnel training, model management, and reuse [9]. Therefore, it is imperative to optimize the model reuse process by implementing a well-designed mechanism that enhances modeling efficiency and promotes higher rates of model reuse.

According to statistics, 40% of designs can be obtained through direct reuse, while another 40% can be achieved by modifying existing designs [15]. Therefore, the effective reuse of multi-dimensional model information (*MMI*) in the existing digital twin models is of great significance for significantly improving the construction efficiency of digital twin models in discrete workshops.

Current research rarely addresses the reuse of *MMI* in discrete workshops for constructing digital twin models [16]. To address the difficulty of reusing *MMI* in existing digital twin models during the conversion process from geometric models to digital twin models, this paper proposes a method for reusing *MMI* in discrete workshops. In this study, geometric models primarily represent the shape of physical entities using three-dimensional mesh models in formats such as STL, OBJ, and OFF [17]. These geometric models, to be built into digital twins, are collectively referred to as imported geometric models. The main contributions of this study are as follows:

(1) An *MMI* and its representation are proposed to facilitate the retrieval of single and multiple devices, thereby enabling effective transfer to imported geometric models.

(2) A model-matching method is proposed for the efficient retrieval of devices and multi-device *MMI*s for importing geometric models.

(3) A workshop *MMI* reuse strategy is proposed to establish the mapping relationship between the imported geometric model and the retrieved *MMI*s, thus initially constructing the digital twin model.

The remaining sections of this paper are organized as follows: Section 2 provides an overview of relevant studies. Section 3 introduces the method employed in this study. Sections 4–6 describe *MMI* and its representation, model-matching methods, and reuse strategies in detail. Section 7 validates the effectiveness of the proposed method through a case study. Section 8 concludes this study and provides an outlook on future research.

2. Literature Review

2.1. Manufacturing System Simulation Software for Digital Twins

Given market demand and competitive pressure, efficient modeling has become the focus of current research. Production system simulation software is often used in the construction of discrete-workshop digital twin models to support subsequent simulation optimization activities. However, such simulation software is not a professional geometric modeling software and typically exhibits limited modeling capabilities. Therefore, digital twin models are often built by combining existing geometric models, software-built scripting languages, and functional components [18]. Common simulation software includes Flexsim [19], DELMIA [20], Factory Simulation [21], Witeness [22], and Visual Components [23]. These applications typically package basic function computation functions into structured functional components, such as forward kinematics, volume sensors, and robot controllers [24–26]. The above software relies heavily on manual selection and model

name-based retrieval when reusing existing geometric models, functional components, and digital twin models [23]. This method is both time-consuming and often yields inaccurate search results, making it difficult to effectively reuse existing models.

2.2. Model Reuse

Model reuse has emerged as a pivotal strategy within digital twin technology, particularly in the context of discrete manufacturing workshops. The efficient utilization of existing modeling resources can substantially reduce development time and associated costs [16]. Model reuse involves leveraging pre-existing geometric, physical, behavioral, and rule-based models to develop new digital twin models, thereby minimizing redundant modeling efforts. With the advancement of digital twin technology, the integration of multidisciplinary data, such as geometry, kinematics, and behavioral patterns, has become essential for creating comprehensive virtual representations of physical systems [4]. This section reviews existing research on model reuse, with particular emphasis on both one-dimensional and multi-dimensional modeling approaches. It also identifies current limitations that underscore the need for systematic model reuse strategies applicable to multi-dimensional models across single and interconnected devices.

Many scholars have conducted modeling research on various dimensional models based on the existing libraries. In the one-dimensional model, Xiao et al. [27] proposed a workshop geometric modeling method based on model reuse, which includes semantic representation, semantic matching, and adaptive modification steps. However, this approach primarily focuses on the reuse of geometric models for single devices. Aiming to reconstruct features of geometric models, Kim et al. [28] proposed a stepwise volume decomposition method based on a feature template library to identify features in geometric models; nevertheless, this method is constrained to the reuse of local features. Wang et al. [29] explored four feature-parameter extraction methods for reconstructing the solid model of a three-dimensional mesh model, namely stretching, rotating, scanning, and lofting, yet only four conventional feature parameters can be reused. Hu et al. [30] proposed a method for point-cloud CAD model recognition and fitting based on deep learning, which can recognize and fit point-cloud models and then construct CAD models. However, this method emphasizes the reuse of local geometric features and does not support the reuse of multi-dimensional information across single or multiple devices. In the multi-dimensional modeling aspect, Liu et al. [31] proposed a method for building digital mockup motion models based on simulation components, enabling the rapid development of motion simulation models for mechanisms. However, this approach lacks a mechanism for establishing associations between simulation components and corresponding geometric models. Qin et al. [32] proposed a CAD/CAE integration modeling method based on feature domain and structural components, which can improve the design efficiency and modeling accuracy of multi-domain simulation models. Zhang et al. [33] proposed a fast construction method for device-level digital twin models in discrete workshops, which constructs a digital twin model of a device by providing basic components; however, it still requires significant manual operations. Song et al. [34] proposed transfer methods for geometric, behavioral, and algorithmic dimensions to achieve model reuse.

Despite the achievements in model reuse, most studies have primarily focused on the reuse of geometric models. The few approaches that involve the reuse of multi-dimensional models also rely heavily on the direct reuse of componentization. Therefore, in response to the emerging demand for multi-dimensional model construction across single and multiple devices, it is imperative to develop a reuse-based approach that enables the efficient utilization of model resources and achieves effective multi-dimensional modeling.

3. Method Overview

The method of this study is shown in Figure 1. The process begins when the designer builds a digital twin model based on the imported geometric model. First, the information imported into the geometric model is extracted, and its similarity to the Digital Twin Model Representation (*DTMR*) is calculated to retrieve the reusable *MMI*. Subsequently, the *MMI* retrieved is mapped to the imported geometric model using the reuse strategies proposed in this study, thereby facilitating the initial construction of the new digital twin model for *MMI* reuse. The method mainly consists of three parts:

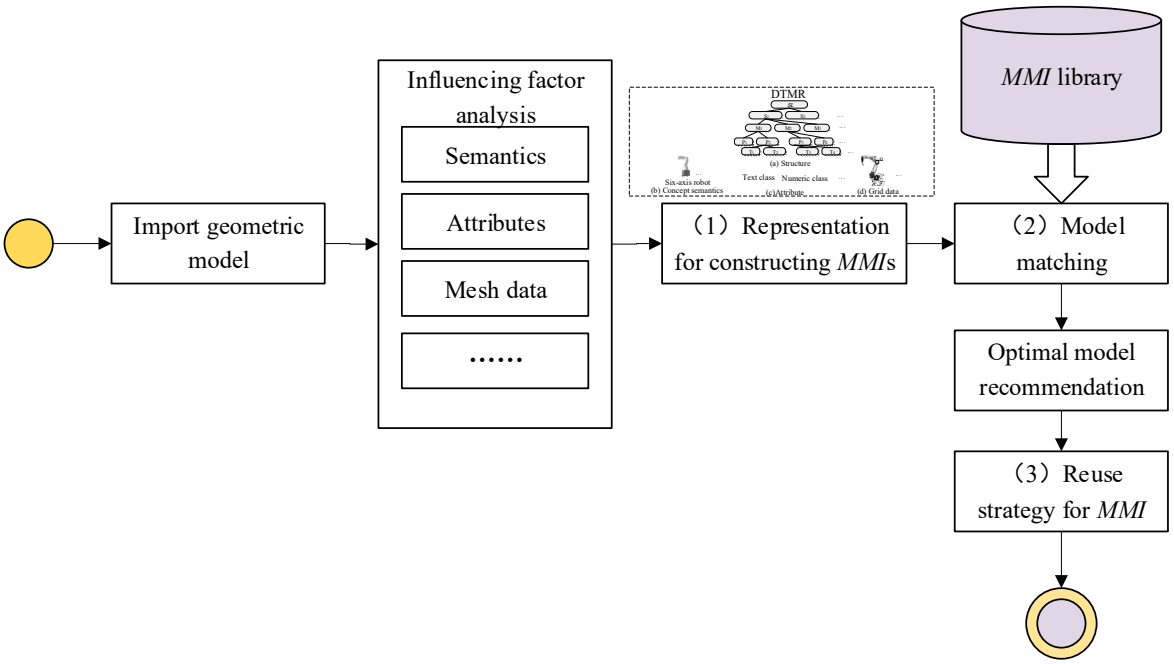


Figure 1. Digital twin model reuse process in a discrete workshop.

(1) *MMI* and its representation. The digital twin model is organized at both the single-device level and the multi-device level, and the *MMI* is characterized by aspects such as assembly hierarchy, semantic description, attributes, and grid data. Furthermore, the *DTMR* is constructed to support the retrieval of *MMI*.

(2) Model matching. After constructing a comprehensive *DTMR*, the similarity of the imported geometric model to the *DTMR* is calculated to retrieve the *MMI* associated with the DTMR. Similarity is calculated at both single and multi-device levels. Search results are sorted in descending order of similarity for subsequent reuse.

(3) Reuse strategy for workshop *MMI*. The retrieval-based *MMI* method aims to offer a systematic framework for the reuse of *MMI*. The retrieved *MMI*s can be effectively mapped onto the imported geometric model, thereby facilitating the initial construction of a novel digital twin model.

4. MMI and Its Characterization

The digital twin model contains reusable information on multiple aspects, such as physical properties, behavioral patterns, and rules. The information is based on geometric models [35,36]. The effective discovery and utilization of this information necessitates a systematic organization and precise expression when transferring it into the imported geometric model, as well as during the preliminary construction of a new digital twin model. Therefore, based on the structure and data of the discrete workshop digital twin model, an *MMI* is constructed, and a *DTMR* is established to support the effective retrieval

and transfer of the *MMI*. The *MMI* and *DTMR* are expressed by Equations (1) and (2), respectively (Figure 2):

$$MMI = \{Geo, PyD, BehD, RuleD\} \quad (1)$$

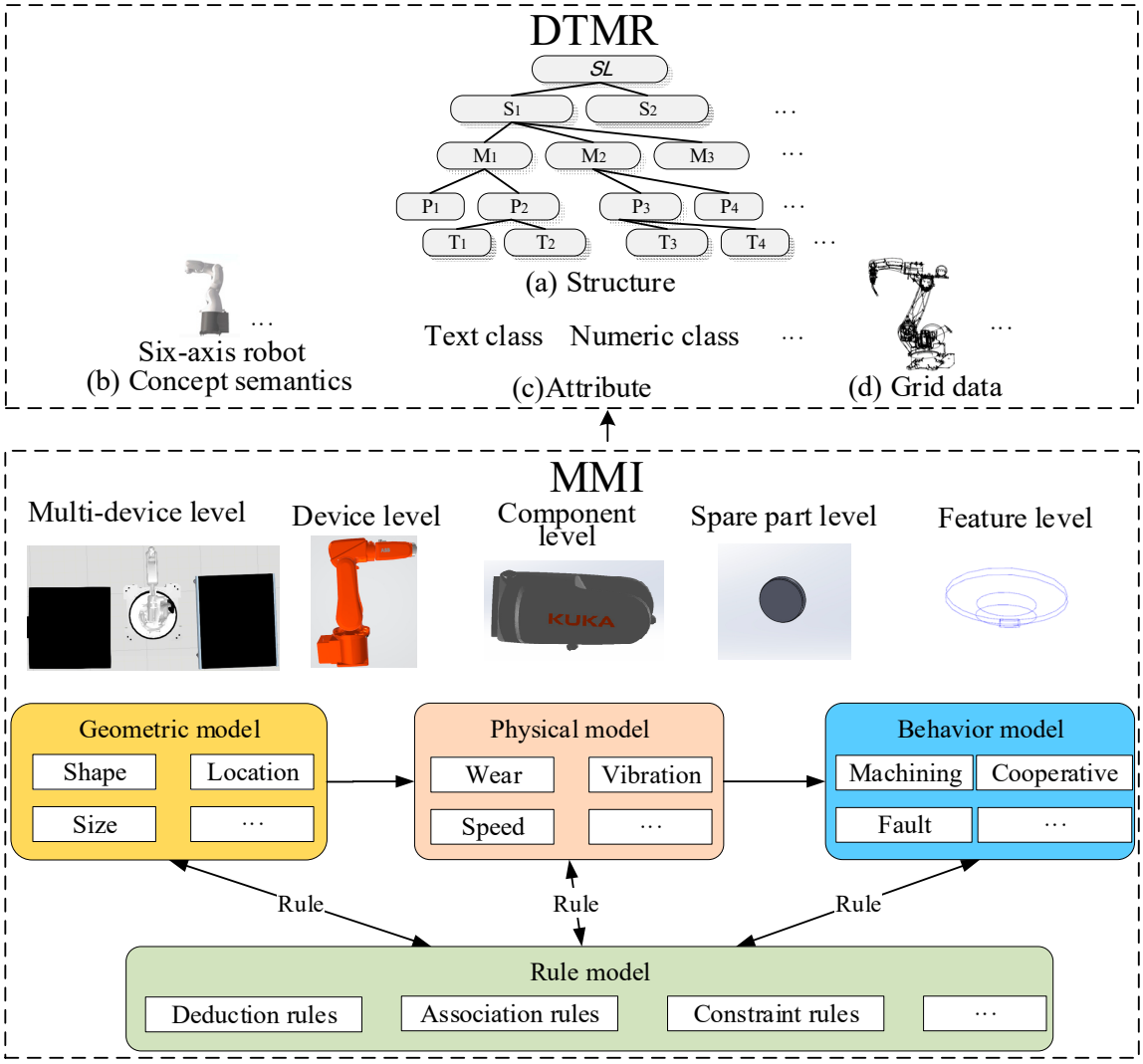


Figure 2. *MMI* and its *DTMR*.

GeoD contains geometric models from the multi-device level or the device level and below. *PhyD*, *BehD*, and *RuleD* represent the behavioral, physical, and rule model information embedded in *GeoD* at different scales of *GeoD* geometry, respectively. Based on *GeoD* at different scales, multi-dimensional models such as behavioral, physical, and rule models of *MMI* can be mapped into the imported geometric model, facilitating the initial construction of a digital twin model.

$$DTMR = \{SL, DL, LP, WG\} \quad (2)$$

$$SL = \{S_1, S_2, \ldots, S_n\} \quad (3)$$

$$S = \begin{cases} M = \{M_1, M_2, \ldots, M_m\} \\ P = \{P_1, P_2, \ldots, P_p\} \\ T = \{T_1, T_2, \ldots, T_t\} \end{cases} \quad (4)$$

$$DL = \{D_1, D_2, \ldots, D_n\} \quad (5)$$

$$WG = \{W_1, W_2, \ldots, W_n\} \quad (6)$$

$$LP = \{P_1, P_2, \ldots, P_p\} \tag{7}$$

When the geometric model in the *MMI* contains multi-device-level geometric models, SL contains the assembly hierarchy between and below these devices. *DL*, *LP*, and *WG* contain the conceptual semantics, attributes, and mesh data of the geometric model for multiple devices, respectively. When the geometric information of the *MMI* contains device-level geometric information, *SL* contains the assembly hierarchy within the device, and *DL*, *LP*, and *WG* contain the conceptual semantics, attributes, and mesh data of the device, respectively.

5. Model Matching

To improve the efficiency of model reuse, the first step is to search for *MMI*. Based on the imported geometric model, a similarity analysis is executed for single-device MMI and multi-device MMI retrieval, as shown in Figure 3.

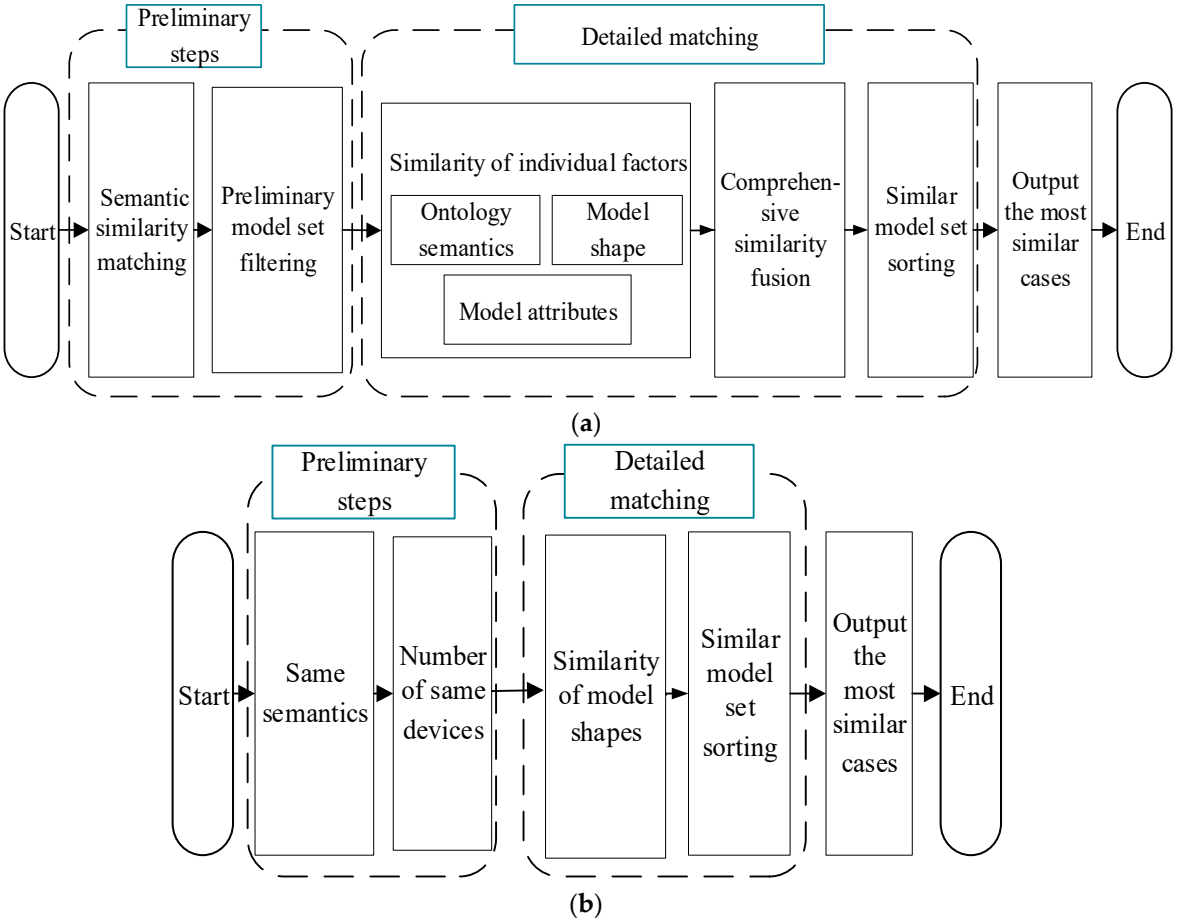


Figure 3. *MMI* matching: (**a**) single-device *MMI* matching; (**b**) multi-device *MMI* matching.

In single-device *MMI* retrieval, the initial screening of a suitable model set is based on calculating semantic similarity within the ontology. The subsequent steps involve extracting ontology semantics, model shape features, and model attributes for similarity calculation. Subsequently, a composite similarity calculation is performed, as expressed by Equation (8). Finally, the retrieved *MMI* is sorted in descending order based on the comprehensive similarity, thus facilitating the manual selection of the most appropriate *MMI*.

$$\delta = \sum_{g \in P_e} \omega_g \times S_g \quad , \sum_{g \in P_e} \omega_g = 1 \tag{8}$$

where g denotes the ontology semantics, model shape, or model attribute factor; ω_g is the weight occupied by element g; S_g represents the similarity of element g. The weights of the components are set based on the heuristic experience of experts.

For the retrieval of multi-device *MMI*, the set of eligible multi-device *MMI* is preliminarily filtered based on specific conditions to ensure that the number of device models in the filtered set matches the semantics, as expressed by Equation (11). Then, based on the model shape feature similarity Sim_{geo}, the similarity values corresponding to each device in the two sets are calculated, and these values are summed and averaged. Finally, the retrieved *MMI* is sorted in descending order based on the calculated mean value to select the most appropriate *MMI*.

$$K = \{K_1, \ldots, K_m\} \tag{9}$$

$$E = \{E_1, \ldots, E_n\} \tag{10}$$

$$AD = \begin{cases} Sim_{Concept}(K_i, E_j) = 1 \\ m = n \end{cases} \tag{11}$$

where K and E are two matched sets of multi-device models. In the sets K and E, Ki and Ej are the two models that have the highest similarity scores in the matching process.

5.1. Ontology Semantic Similarity Analysis

With the continuous development of semantic web services, ontology semantic similarity calculation has been widely employed in fields such as text classification, information retrieval, and data mining [37]. Theoretically, only descriptions with the same or similar semantics have a certain reuse value. The semantic structure tree of discrete shop elements is shown in Figure 4.

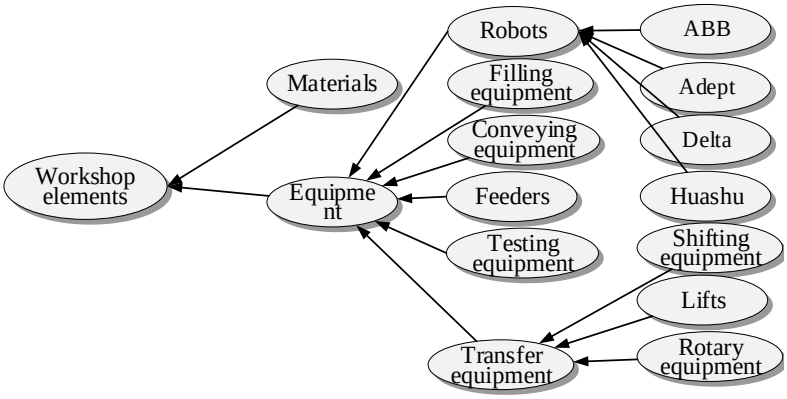


Figure 4. The ontological concepts and relationships of workshop elements.

Semantic distance refers to the length of the shortest path connecting two concepts in the ontology hierarchy tree, as expressed by Equation (12).

$$Dist(M_1, M_2) = \sum_{i=1}^{n} edge_i \tag{12}$$

where $edge_i$ denotes the ith edge on the shortest path connecting concepts M_1 and M_2.

A smaller semantic distance between two concepts indicates a greater similarity. Considering the influence of ontology tree depth, the similarity of the two concepts M_1 and M_2 in terms of semantic distance is shown in Equation (13):

$$Sim_{dist}(M_1, M_2) = \frac{2(Dep(Tree) - 1) - Dist(M_1, M_2)}{2(Dep(Tree) - 1)} \tag{13}$$

where *Dep* (*Tree*) represents the depth of the ontology tree; *Dist* (M_1, M_2) denotes the semantic distance between two concepts.

Semantic overlap reflects the degree of semantic similarity between two concepts on their common ancestor nodes. The semantic overlap between two concepts M_1 and M_2 is denoted as $Sim_{dop}(M_1,M_2)$, as expressed by Equation (14):

$$Sim_{dop}(M_1, M_2) = \sqrt{\frac{2Dep(Lca) - 1}{Dep(M_1) + Dep(M_2) + 1}} \tag{14}$$

where *Lca* denotes the nearest common ancestor node of M_1 and M_2 in the ontology tree; *Dep*(*M*) represents the depth of node *M* in the tree.

Node depth indicates the level of hierarchy within the concept in the ontology tree. As the depth of the nodes increases, the content represented by the concept becomes more specific, and the similarity between concepts increases. The similarity between concepts M_1 and M_2 in terms of node depth is denoted as $Sim_{dep}(M_1,M_2)$, as expressed by Equation (15):

$$Sim_{dep}(M_1, M_2) = \frac{1}{2}\left(\frac{Dep(M_1) + Dep(M_2)}{|Dep(M_1) - Dep(M_2)| + 2Dep(Tree)} + \frac{Dep(Lca)}{Dep(Tree)}\right) \tag{15}$$

where *Dep*(*M*) represents the depth of node *M*; *Dep*(*Tree*) denotes the depth of the ontology tree.

Based on the above factors, the semantic synthetic similarity calculation is proposed, as expressed by Equation (16):

$$Sim_{Conpect}(M_1, M_2) = \begin{cases} pSim_{dist}(M_1, M_2) + qSim_{dop}(M_1, M_2) + wSim_{dep}(M_1, M_2) \\ p + q + w = 1 \\ p \geq q \geq w \geq 0 \end{cases} \tag{16}$$

where p, q, and w are weight adjustment factors. The weights can be assigned by analyzing the degree to which each influencing factor affects the similarity, with p, q, and w taking the values of 0.5, 0.3, and 0.2, respectively.

5.2. Similarity Analysis of Model Attributes

The feature parameter description of the device includes both the feature name and its corresponding value. When two concepts contain more of the same data attributes and the values of the same attributes are closer, the two concepts tend to be more similar. Considering the effect of characteristics with the same or different names on similarity, it is necessary to compare the differences in the values of characteristics with the same names to accurately reflect the similarity between concepts. The steps for calculating the overall similarity of model attributes are as follows:

Step 1: Input the characterization of models M_1 and M_2: $M_{1Attr} = \{x_i =< N_i - V_i >, 1 \leq i \leq m\}$, $M_{2Attr} = \{y_j =< N_j - V_j >, 1 \leq j \leq n\}$. N and V represent the characteristic name and the characteristic value, respectively; m and n are the number of characteristics in M_1 and M_2, respectively.

Step 2: Calculate the similarity of characteristic names, as expressed by Equation (17):

$$Sim_{Type}(M_1, M_2) = \frac{f(M_1 \cap M_2)}{f(M_1 \cap M_2) + \theta f(M_1 - M_2) + (1 - \theta) f(M_2 - M_1)} \tag{17}$$

where $f(M_1 \cap M_2)$ is the number of characteristics with the same name between concepts M_1 and M_2. $f(M1 - M2)$ and $f(M_2 - M_1)$ are the number of characteristics that are present in $M_1(M_2)$ but not in $M_2(M_1)$; θ is the symmetry moderator, and θ = 0.5, i.e., the difference in the degree of similarity induced by the symmetry, is not taken into account.

Step 3: Extract the common characteristic elements of M_1 and M_2. The number of elements with identical characteristic names is k, and their one-to-one corresponding characteristic values are denoted as $Value(M_1) = \{V_{1,M_1}, \ldots, V_{t,M_1}, \ldots, V_{k,M_1}\}$, $Value(M_2) = \{V_{1,M_2}, \ldots, V_{t,M_2}, \ldots, V_{k,M_2}\}$, V_1~V_t for numerical characteristic values and V_{t+1}~V_k for textual characteristic values.

Step 4: Calculate the characteristic value of the numeric class $Sim_{Number}(M_1, M_2)$ and text class $Sim_{String}(M_1, M_2)$; see Equations (18) and (21). Different similarity calculation methods are used depending on the type to which the characteristic value belongs. The similarity of numerical class characteristic values is calculated using the principle of relative error, and the similarity of textual characteristic values is calculated using the Jaro–Winkler algorithm [38], as expressed by Equation (20):

$$Sim_{Number}(M_1, M_2) = \varepsilon / (\varepsilon + \sqrt{\sum_{i=1}^{k} (\frac{A_{i,M_1} - A_{i,M_2}}{A_{i,M_1}})^2}) \tag{18}$$

$$Jaro(S, T) = \frac{1}{3} \cdot (\frac{m}{|S|} + \frac{m}{|T|} + \frac{m - t}{m}) \tag{19}$$

$$\mathrm{Sim}_{character} = Jaro - Winkler(S, T) = Jaro(S, T) + lp(1 - Jaro(S, T)) \tag{20}$$

$$Sim_{String}(M_1, M_2) = \sum_{i=t+1}^{k} Sim_{character}(V_{i,M_1}, V_{i,M_2}) / (k - t) \tag{21}$$

where $|S|$ and $|T|$ represent the lengths of strings S and T, respectively; m is the number of character matches between the two strings; t denotes the number of character permutations between S and T; l is the prefix-matching length between S and T; p is the scaling factor constant, which is the contribution of the common prefix to the similarity, with the default value of 0.1; ε is the relative error adjusting factor, which is taken as ε = 2.0.

Step 5: Calculate the composite similarity $Sim_{Attr}(M_1, M_2)$, as expressed by Equation (23):

$$Sim_{Value}(M_1, M_2) = \frac{t}{k} \times Sim_{Number}(M_1, M_2) + (\frac{k - t}{k}) \times Sim_{String}(M_1, M_2) \tag{22}$$

$$Sim_{Attr}(M_1, M_2) = \lambda Sim_{Type}(M_1, M_2) + (1 - \lambda) Sim_{Value}(M_1, M_2) \tag{23}$$

where λ is the moderating factor, which is the weight assigned based on the similarity of characteristic names, with λ = 0.6.

5.3. Similarity Analysis of Model Shapes

To obtain the difference in shape information of geometric models, the shape distribution method proposed by Osada et al. [39] was used to evaluate the similarity between universal three-dimensional models represented by triangular meshes. Let the shape distribution curves of two compared geometric shapes be $q(x)$ and $p(x)$. Then, the similarity of the shape is computed using an effective similarity evaluation method for discrete or continuous probability distributions, the Bhattacharyya distance [40], as expressed by Equation (24):

$$Sim_{geo} = 1 + \ln(\int \sqrt{p(x)q(x)}dx) \tag{24}$$

6. Workshop *MMI* Reuse Strategy

Based on the *MMI* retrieved in Section 5, further effective transfer of the multi-dimensional model in the *MMI* to the imported geometric model is necessary for the initial construction of the digital twin model. Therefore, a reuse strategy for *MMI* is proposed, as shown in Figure 5:

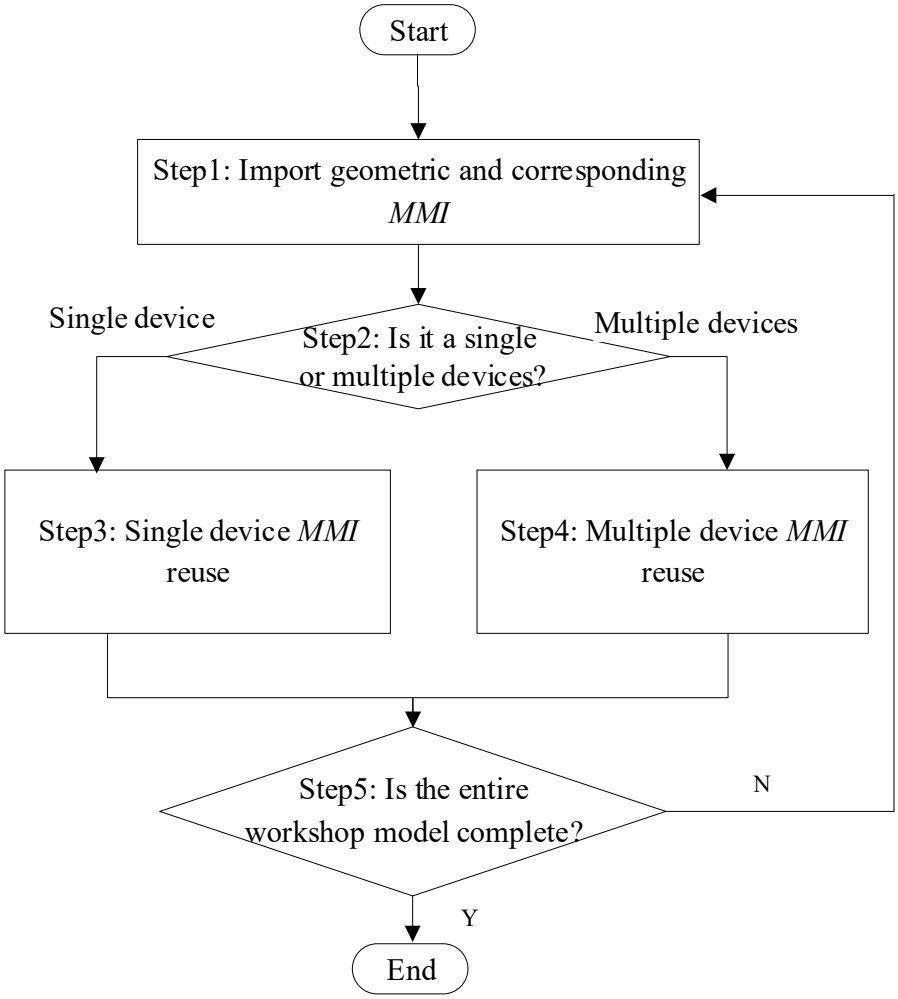


Figure 5. Model reuse process.

Step 1: Determine the imported geometric model and its corresponding *MMI*.

Step 2: Determine whether the currently selected imported geometric model is a single-device or multi-device model. If it is a single device, Step 3 is executed; otherwise, Step 4 is executed.

Step 3: Reuse *MMI* for single devices. The correspondence between the *MMI* and the geometric information of the imported geometric model is identified at the device level, component level, part level, and local feature level. Then, the physical, behavioral, and rule-related information associated with the geometric in the *MMI* is mapped to the imported geometric model layer by layer. In order to better assist local feature capture, the grid model can be processed by geometric feature partitioning [39] (a technique specifically designed for grid models) to obtain geometric information.

At the device level, physical characteristics (e.g., device performance, kinematics) and behaviors (e.g., performance degradation, response mechanisms) relevant to the entire device are mapped to the imported geometric model.

At the component level, the *MMI* is further mapped to the individual components of the device (e.g., the functional description of the component, material type) based on the corresponding components of the imported geometric model.

At the local feature level, the *MMI* of the local features (e.g., geometric information, mechanical properties) is mapped to the corresponding locations in the imported geometric model.

Step 4: Reuse *MMI* for multiple devices. The corresponding relationship between the geometric models of the devices is established, and then the relevant information in *MMI* is mapped to the imported geometric model. This includes mapping logical relationships between devices and reusing global behavior to ensure that multiple devices work together in a coordinated manner.

Step 5: Determine if the entire workshop model has been fully executed. If yes, terminate the process; otherwise, return to Step 1.

7. Case Analysis

7.1. Case Implementation

To verify the effectiveness of the proposed method, a specific arc-welding workshop is selected as a case study, and the implementation results are analyzed. The geometric model of the arc-welding workshop is selected as the model to be imported. The single-device and multi-device geometric models, comprising seven groups, are shown in Figure 6 (Visual Components 4.7). Furthermore, the case implementation process is shown in the following steps in Figure 7: (a) The determination of the imported geometric model; (b) model matching; (c) optimal model recommendation; and (d) model reuse. In (a), the geometric model is first imported. In (b), the imported geometric models are matched in the model library, and the reusable *MMI* is filtered in (c) based on the similarity value of the matching results. Finally, in (d), *MMI* reuse is completed by transferring the selected *MMI* into the imported geometric model.

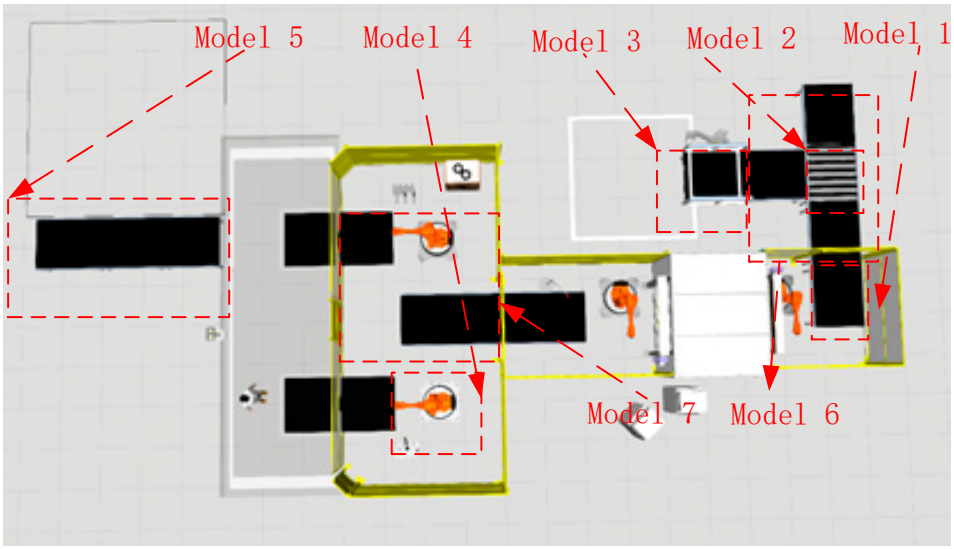


Figure 6. Device geometric model and multi-device geometric model.

7.2. Analysis of Results

7.2.1. Analysis of Matching Results

(1) Evaluation indicators

To verify the effectiveness of the proposed model-matching method, model-matching tests and analyses are conducted using the device-level geometric models shown in Figure 6. The precision and recall ratios [41] are adopted as evaluation indicators, which can be expressed as follows:

$$P_N = \frac{M_S}{M_R} \tag{25}$$

$$P_C = \frac{M_S}{M_A} \tag{26}$$

where P_N represents the recall ratio; M_S represents the number of relevant models matched; M_R represents the number of all relevant models; P_C represents the precision ratio; and M_A represents the total number of matched models.

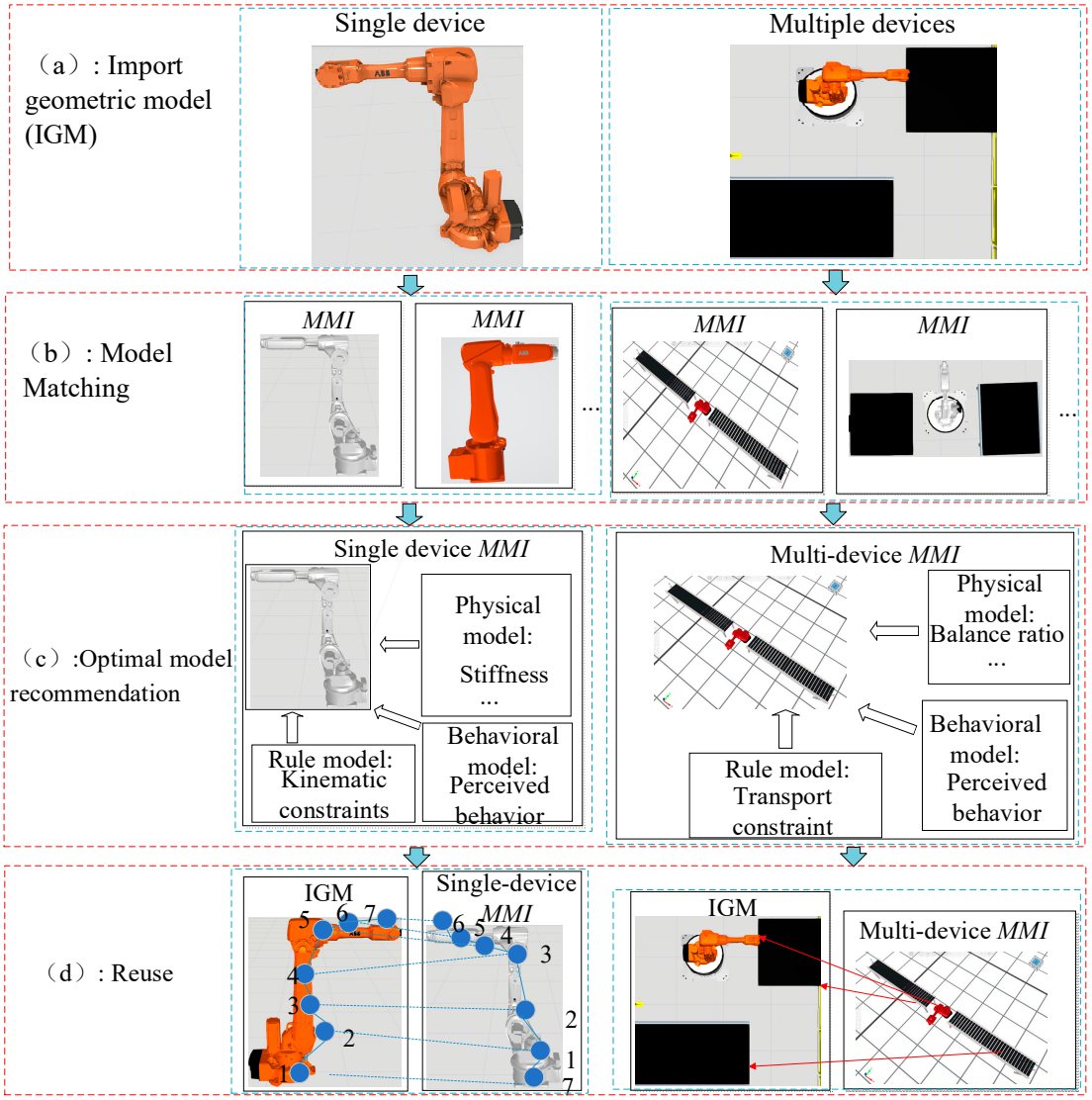


Figure 7. Case implementation process.

(2) Analysis of results

The matching results are analyzed from a device-level model. In the experimental setup for device-level model retrieval, the TOP-N values are set to 1, 5, and 10. The proposed method shows excellent performance in terms of recall and precision (see Figures 8 and 9). A single geometric model shape-matching method leads to multiple class models in the matching results, indicating that this method primarily emphasizes the similarity of shape features among geometric models. However, it poses challenges in distinguishing between models within the same class. In contrast, the method proposed in this study integrates ontology semantics, model attributes, and model shape feature information to enable targeted retrieval. Compared to data information on only one feature, the proposed method can match the model more accurately.

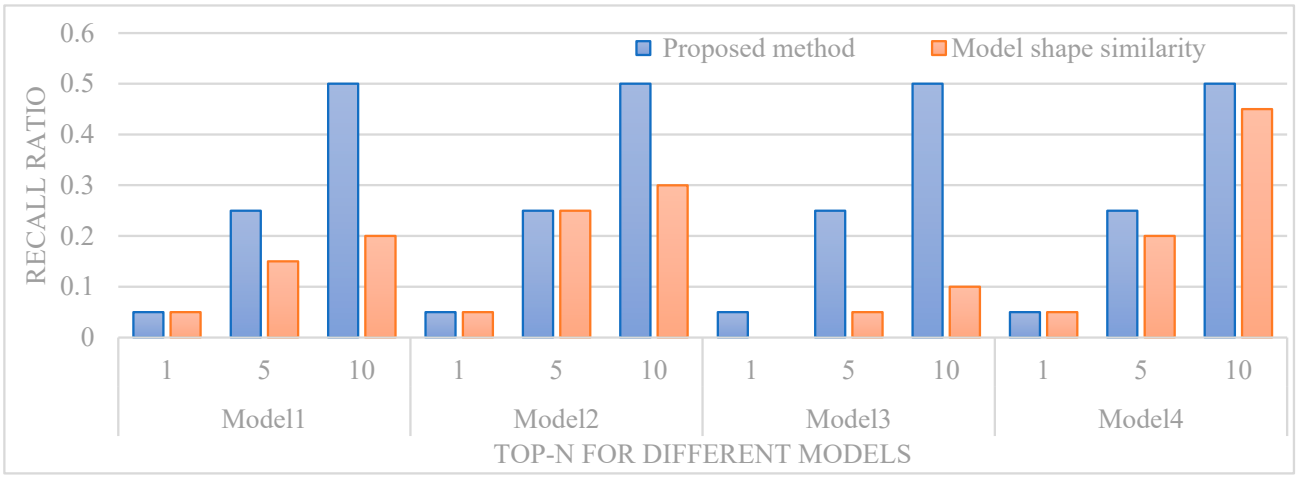


Figure 8. Recall.

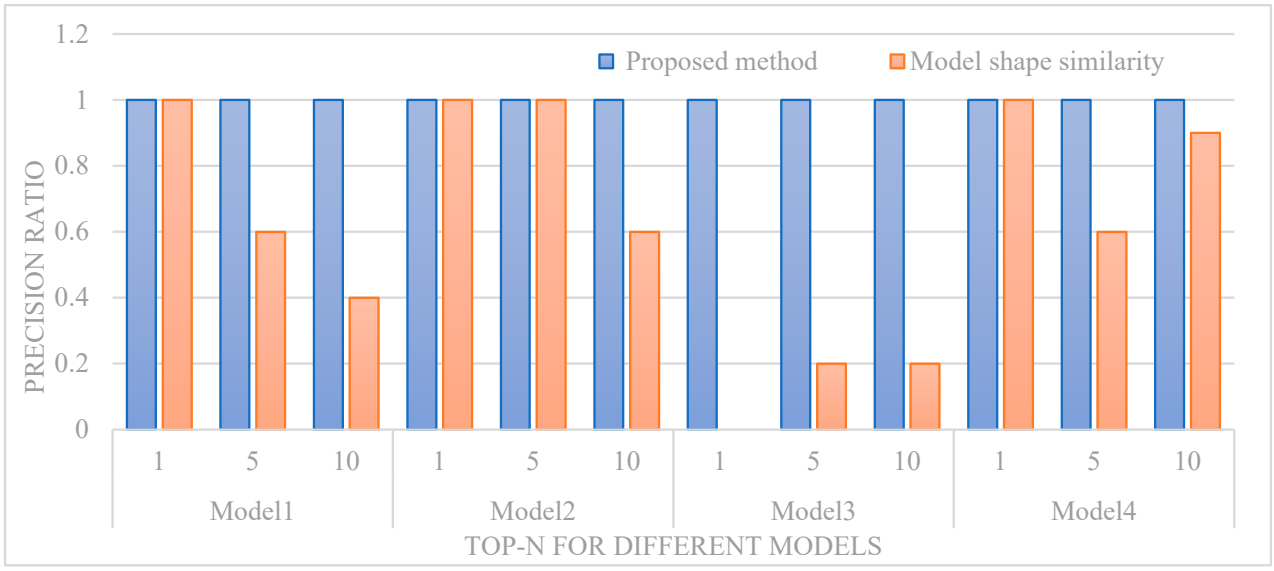


Figure 9. Precision.

Based on the analysis results, the proposed method can better capture the differences between the local details of the model and match the appropriate *MMI* for the imported geometric model.

7.2.2. Modeling Efficiency Analysis

In order to verify the advantages of the proposed method in the construction efficiency of the workshop digital twin model, a comparative experiment is designed between the traditional construction process (Process A) based on functional components and the optimization process (Process B) applied in the proposed method, as shown in Figure 10. Figure 11 presents the comparison results of the number of operations required for constructing each model. The geometric model in Figure 6 is selected as the test object and evaluated based on the minimum number of operations required to construct the model.

In Process A, engineers must manually add attributes, functional components, and associations, which requires knowledge across multiple domains. In B processes, manual operations can be significantly reduced by reusing existing *MMI*s. When calculating the number of operations, the addition of a single functional component (a), a single digital twin model (b), a single behavioral state (c), and a single behavioral event (d) are counted

as 1 operation, while the rest of the additions of association relations, constraint rules, and behavioral actions are not counted.

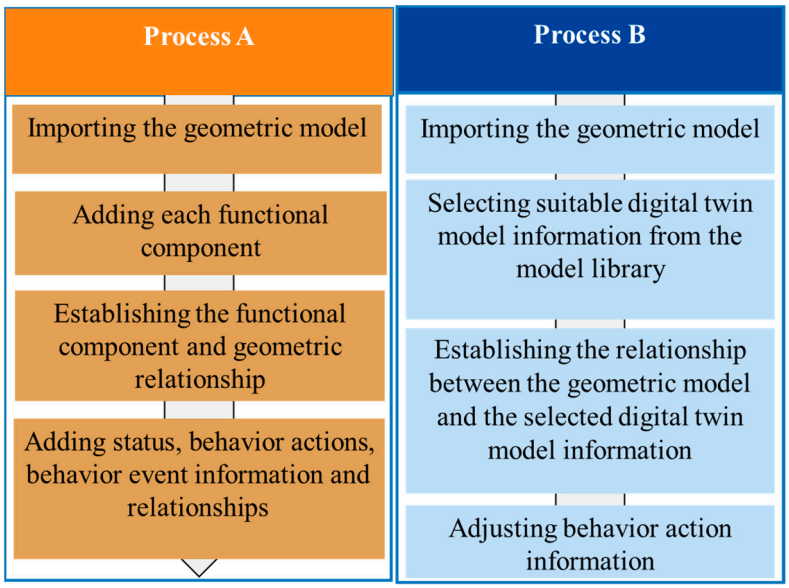


Figure 10. Model construction process.

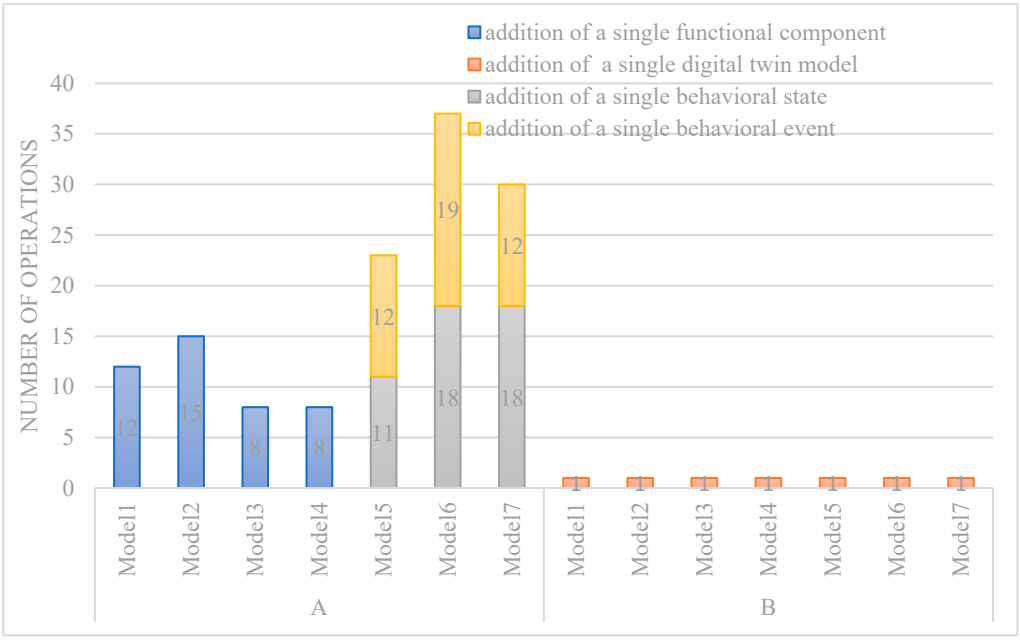


Figure 11. A comparison of the number of operations for each model construction under different model construction processes.

As shown in Table 1 and Figure 11, Process B results in a significant reduction in the number of operations (a total of 126) compared to Process A, demonstrating the effectiveness of the proposed method in enhancing the construction efficiency of the workshop digital twin model. Process A requires engineers to have a high level of professionalism and has a cumbersome and complex build process. Process B significantly reduces the frequency of human–computer interaction through a reuse mechanism, thus improving the modeling efficiency.

Table 1. Comparison of total data.

Process	Process A	Process B
Number of operations	133	7

8. Conclusions

This paper discusses the construction process of the workshop digital twin model and proposes the *MMI* reuse method. This method describes device and multi-device *MMI*s and retrieves the corresponding *MMI*s efficiently. Subsequently, the construction of the digital twin model is initiated by mapping the selected *MMIs* to the imported geometric model using the proposed model reuse strategy. The effectiveness of the proposed method is validated through case studies in the arc-welding workshop. The accuracy of single-model matching remains consistently at 100% across all model tests. Furthermore, the proposed method reduces the number of operations by 126 compared to existing approaches in multi-device model construction. Despite the initial progress achieved in workshops regarding the reuse of digital twin models, there exists scope for further enhancement, thereby providing valuable directions for subsequent research: (1) The current model reuse strategy still requires a lot of manual operations, and future research will be devoted to the automatic association of geometric models with *MMI*; (2) further research will be conducted on matching algorithms to achieve more efficient retrieval. (3) Additional evaluation of the method's effectiveness, utility, and limitations should be conducted through surveys involving practitioners across various discrete manufacturing environments.

Author Contributions: All authors contributed to the study of this paper. R.L.: Conceptualization, methodology, software, writing—original draft; J.Z.: Supervision, writing—review, resources, investigation, formal analysis. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Research Project of Hubei Provincial Department of Education (nos. Q20234504), the Open Fund Project of Hubei Key Laboratory of Intelligent Convey Technology and Device, Hubei Polytechnic University, China (nos. 2024XM102), and the Talent Introduction Fund Project of Hubei Polytechnic University (nos. 24xjz04R).

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: Ruiping Luo is engaged in postdoctoral research at Huangshi Donper Compressor Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Cimino, C.; Negri, E.; Fumagalli, L. Review of digital twin applications in manufacturing. *Comput. Ind.* **2019**, *113*, 103130. [CrossRef]
2. Liu, J.; Liu, J.; Zhuang, C.; Liu, Z.; Miao, T. Construction method of shop-floor digital twin based on MBSE. *J. Manuf. Syst.* **2021**, *60*, 93–118. [CrossRef]
3. Boyes, H.; Watson, T. Digital twins: An analysis framework and open issues. *Comput. Ind.* **2022**, *143*, 103763. [CrossRef]
4. Luo, R.; Sheng, B.; Song, K.; Jian, Y.; Fu, G.; Zhao, F. Assembly feature construction method of equipment mesh model for digital twin workshops. *Proc. Inst. Mech. Eng. B J. Eng. Manuf.* **2024**, *238*, 1940–1951. [CrossRef]
5. Zhang, M.; Tao, F.; Nee, A.Y.C. Digital twin enhanced dynamic job-shop scheduling. *J. Manuf. Syst.* **2021**, *58*, 146–156. [CrossRef]
6. Zhang, L.; Zhuang, C.; Tian, Y.; Yao, M. Construction and application of energy footprint model for digital twin workshop oriented to low-carbon operation. *Sensors* **2024**, *24*, 3670. [CrossRef] [PubMed]
7. Wang, Y.; Wang, Y.; Ren, W.; Jiang, Z. Knowledge driven multiview bill of material reconfiguration for complex products in the digital twin workshop. *Int. J. Adv. Manuf. Technol.* **2024**, *130*, 3469–3480. [CrossRef]
8. Ouahabi, N.; Chebak, A.; Kamach, O.; Laayati, O.; Zegrari, M. Leveraging digital twin into dynamic production scheduling: A review. *Robot. Comput. Integr. Manuf.* **2024**, *89*, 102778. [CrossRef]

9. Jia, W.; Wang, W.; Zhang, Z. From simple digital twin to complex digital twin Part I: A novel modeling method for multi-scale and multi-scenario digital twin. *Adv. Eng. Inform.* **2022**, *53*, 101706. [CrossRef]
10. Luo, R.; Sheng, B.; Lu, Y.; Huang, Y.; Fu, G.; Yin, X. Digital twin model quality optimization and control methods based on workflow management. *Appl. Sci.* **2023**, *13*, 2884. [CrossRef]
11. Tliba, K.; Diallo, T.M.; Penas, O.; Ben Khalifa, R.; Ben Yahia, N.; Choley, J.Y. Digital twin-driven dynamic scheduling of a hybrid flow shop. *J. Intell. Manuf.* **2023**, *34*, 2281–2306. [CrossRef]
12. Guo, H.; Chen, M.; Mohamed, K.; Qu, T.; Wang, S.; Li, J. A digital twin-based flexible cellular manufacturing for optimization of air conditioner line. *J. Manuf. Syst.* **2021**, *58*, 65–78. [CrossRef]
13. Agostino, Í.R.S.; Broda, E.; Frazzon, E.M.; Freitag, M. Using a digital twin for production planning and control in Industry 4.0. In *Scheduling in Industry 4.0 and Cloud Manufacturing*; Springer: Cham, Switzerland, 2020; pp. 39–60.
14. Negri, E.; Pandhare, V.; Cattaneo, L.; Singh, J.; Macchi, M.; Lee, J. Field-synchronized digital twin framework for production scheduling with uncertainty. *J. Intell. Manuf.* **2021**, *32*, 1207–1228. [CrossRef]
15. Chen, J.; Jia, X. An approach for assembly process case discovery using multimedia information source. *Comput. Ind.* **2020**, *115*, 103176. [CrossRef]
16. Tao, F.; Xiao, B.; Qi, Q.; Cheng, J.; Ji, P. Digital twin modeling. *J. Manuf. Syst.* **2022**, *64*, 372–389. [CrossRef]
17. Grodotzki, J.; Müller, B.; Tekkaya, A.E. Introducing a general-purpose augmented reality platform for the use in engineering education. *Adv. Ind. Manuf. Eng.* **2023**, *6*, 100116. [CrossRef]
18. Luo, R.; Sheng, B.; Fu, G.; Huang, Y.; Jian, Y.; Chen, G. A virtual model knowledge updating method driven by data. *Int. J. Adv. Manuf. Technol.* **2024**, *131*, 5403–5418. [CrossRef]
19. Tokgöz, E. Industrial engineering and simulation experience using Flexsim software. *Comput. Educ. J.* **2017**, *8*, 86–95.
20. Bzymek, Z.M.; Nunez, M.; Li, M.; Powers, S. Simulation of a machining sequence using DELMIA/QUEST software. *Comput. Aided Des. Appl.* **2008**, *5*, 401–411. [CrossRef]
21. Zhu, H. *Modeling and Simulation of Production Systems*, 1st ed.; Tsinghua University Press: Beijing, China, 2022; pp. 1–200.
22. Yasir, A.S.H.M.; Mohamed, N.M.Z.N. Assembly line efficiency improvement by using WITNESS simulation software. In Proceedings of the International Conference on Materials Science and Engineering, Penang, Malaysia, 6–7 March 2018.
23. Luo, R.; Sheng, B.; Huang, Y.; Jian, Y.; Song, K.; Lu, Y.; Jiang, F. Key technologies and development trends of digital twin-based production system simulation software. *Comput. Integr. Manuf. Syst.* **2023**, *29*, 1965–1978.
24. Beyer, D. CCVisu: Automatic visual software decomposition. In Proceedings of the 30th International Conference on Software Engineering, Leipzig, Germany, 10–18 May 2008; ACM: New York, NY, USA, 2008; pp. 1–10.
25. Jamwal, P.K.; Xie, S.Q.; Tsoi, Y.H.; Aw, K.C. Forward kinematics modelling of a parallel ankle rehabilitation robot using modified fuzzy inference. *Mech. Mach. Theory* **2010**, *45*, 1537–1554. [CrossRef]
26. Wang, H.; Xiang, Z.; Giorgia, P.; Mu, X.; Yang, Y.; Wang, Z.L.; Lee, C. Triboelectric liquid volume sensor for self-powered lab-on-chip applications. *Nano Energy* **2016**, *23*, 80–88. [CrossRef]
27. Xiao, B.; Qi, Q.; Tao, F. Multi-dimensional modeling and abnormality handling of digital twin shop floor. *J. Ind. Inf. Integr.* **2023**, *35*, 100492. [CrossRef]
28. Kim, B.C.; Mun, D. Stepwise volume decomposition for the modification of B-rep models. *Int. J. Adv. Manuf. Technol.* **2014**, *75*, 1393–1403. [CrossRef]
29. Wang, J.; Gu, D.; Gao, Z.; Yu, Z.; Tan, C.; Zhou, L. Feature-based solid model reconstruction. *J. Comput. Inf. Sci. Eng.* **2013**, *13*, 011004. [CrossRef]
30. Hu, S.; Polette, A.; Pernot, J.P. SMA-Net: Deep learning-based identification and fitting of CAD models from point clouds. *Eng. Comput.* **2022**, *38*, 5467–5488. [CrossRef]
31. Liu, Z.; Fu, Y.; Tan, J. Motion model construction and reuse of digital prototype based on simulation components. *J. Mech. Eng.* **2009**, *45*, 118–124. [CrossRef]
32. Qin, B.; Yan, C.; Liu, F. An integrated CAD/CAE modeling approach based on feature domains and structural components. *Comput. Integr. Manuf. Syst.* **2011**, *17*, 1397–1403.
33. Zhang, Y.; Zhang, C.; Yan, J.; Yang, C.; Liu, Z. Rapid construction method of equipment model for discrete manufacturing digital twin workshop system. *Robot. Comput. Integr. Manuf.* **2022**, *75*, 102309. [CrossRef]
34. Song, J.; Liu, S.; Ma, T.; Sun, Y.; Tao, F.; Bao, J. Resilient digital twin modeling: A transferable approach. *Adv. Eng. Inform.* **2023**, *58*, 102148. [CrossRef]
35. Zhang, D.; Liu, Z.; Li, F.; Zhao, Y.; Zhang, C.; Li, X.; Zhang, Y. The rapid construction method of the digital twin polymorphic model for discrete manufacturing workshop. *Robot. Comput. Integr. Manuf.* **2023**, *84*, 102600. [CrossRef]
36. Liu, J.; Ji, Q.; Zhou, H.; Du, C.; Liu, X.; Li, M. A multi-dimensional evolution modeling method for digital twin process model. *Robot. Comput. Integr. Manuf.* **2024**, *86*, 102667. [CrossRef]
37. Karabulut, E.; Pileggi, S.F.; Groth, P.; Degeler, V. Ontologies in digital twins: A systematic literature review. *Future Gener. Comput. Syst.* **2023**, *149*, 101–117. [CrossRef]

38. Doan, A.; Madhavan, J.; Dhamankar, R.; Domingos, P.; Halevy, A. Learning to match ontologies on the semantic web. *VLDB J.* **2003**, *12*, 303–319. [CrossRef]
39. Osada, R.; Funkhouser, T.; Chazelle, B.; Dobkin, D. Shape distributions. *ACM Trans. Graph.* **2002**, *21*, 807–832. [CrossRef]
40. Kashyap, R. The perfect marriage and much more: Combining dimension reduction, distance measures and covariance. *Physica A* **2019**, *536*, 120938. [CrossRef]
41. Li, Y.; Fan, B.; Zhang, W.; Jiang, Z. TireNet: A high recall rate method for practical application of tire defect type classification. *Future Gener. Comput. Syst.* **2021**, *125*, 1–9. [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.