

Article

A Hierarchical Integrated Modeling Method for the Digital Twin of Mechanical Products

Menglei Zheng  and Ling Tian *

Department of Mechanical Engineering, Tsinghua University, Beijing 100084, China; zml16@mails.tsinghua.edu.cn

* Correspondence: tianling@tsinghua.edu.cn

Abstract: With the development of information and communication technology, massive amounts of data are generated during the entire lifecycle of mechanical products. However, their isolated and fragmented state hinders further empowerment of smart manufacturing. Digital twins have attracted considerable attention as they enable a user to rebuild all elements of a physical entity in a virtual space, targeted at the effective fusion of data from multiple sources with different formats, while its modeling method still needs further research. In this context, we propose a native, full-element digital twin modeling method for mechanical products. This ontology-based method establishes a unified and computer-understandable model framework for mechanical products by abstracting the essential content and relationships of data and by storing them in a graph database efficiently. The developed model could serve as a data center for the entire lifecycle of the product or could be combined with existing data management systems, integrating the previously isolated, fragmented, and scattered data on various platforms. In addition, the model utilizes the structural characteristics of mechanical products and is developed as a hierarchical digital mapping to better meet the application requirements. Finally, a case study of a helicopter digital twin is presented to verify the proposed method.



Citation: Zheng, M.; Tian, L. A Hierarchical Integrated Modeling Method for the Digital Twin of Mechanical Products. *Machines* **2022**, *10*, 2. <https://doi.org/10.3390/machines10010002>

Academic Editors: Jiafu Wan, Yuqian Lu, Sang Do Noh and Dan Zhang

Received: 21 October 2021

Accepted: 16 December 2021

Published: 21 December 2021

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



Copyright: © 2021 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/>).

Keywords: digital twin; ontology model; hierarchical model; graph database; mechanical product

1. Introduction

The new generation of information and communication technology has produced profound changes to the manufacturing industry. Industry 4.0 has become a major factor in the future competitiveness of enterprises, and the use of sensors, data collection and analysis, information technology infrastructure, and mobile terminals are considered important basic technologies [1]. Digitization and networking make the acquisition, sharing, and use of data faster and cheaper, and massive amounts of data are generated in the entire lifecycle of mechanical products [2,3]. With high information density and utilization value, these data can be combined with emerging intelligent technologies such as finite element simulation, machine learning, and artificial intelligence. These data provide broad prospects for intelligent manufacturing, with considerable application potential in various product lifecycles, spanning design, manufacturing, operation, maintenance, etc. [4,5].

However, because of the lack of effective integration methods, these data are usually isolated or fragmented, hindering further empowerment of intelligent manufacturing [6]. This means that stakeholders such as designers, manufacturers, and maintainers still cannot obtain the data generated in other stages and then cannot conduct manual or intelligent analysis to improve the current link. Although digital threads provide data transmission, methods and technologies such as simulations can use the acquired data for analysis, the connection, that is, the integrated modeling of product data, is still missing. For example, Wuest et al. [7] indicated that machine learning, which can be used to process high-dimensional problems and data to discover implicit knowledge, etc., has application advantages in manufacturing, but the acquisition of relevant data is still a common challenge. The integration of product data and the fusion of virtual models

and physical entities are considered the foundations of product smart and sustainable manufacturing. In this context, digital twin is recognized as a key approach and has received substantial attention from scholars and the industry. In 2012, digital twin was included in the “Modeling, Simulation, Information Technology and Processing Roadmap” released by NASA [8]. For four consecutive years since 2016, Gartner [9], one of the world’s most authoritative information technology research and consulting companies, listed digital twins as one of the top ten strategic technology development tendencies. A comment by Tao et al. about digital twins published in *Nature* pointed out that the virtual model could boost smart manufacturing by simulating decisions and optimization, from design to operations [10].

However, modeling as a foundation technology for digital twin implementation still has many challenges to be solved. In the field of mechanical product digital twins, most of the models currently developed are partial mappings oriented toward specific purposes, or just real-time data and intelligent algorithm-added simulation models. Although they can meet the current needs, their application scope and scalability are limited. The existing PLM system has realized the integration of many product data and can serve as the foundation of functional and effective digital twins, but it still needs development in terms of cyber-physical integration and rapid iteration [11,12]. In addition, many characteristics of mechanical products, such as hierarchy, have not been fully utilized in existing research on digital twin modeling. Therefore, a model framework that has the ability to integrate all heterogeneous data from products and to iterate it quickly is needed. Although not all data may be used in a specific requirement, the model framework provides the possibilities for various applications.

In accordance with the above, this study provides the following contributions:

- A native full-element digital twin modeling method for mechanical products is proposed.
- The developed model could serve as a data center for the entire lifecycle of the product or could be combined with existing data management systems, integrating the previously isolated, fragmented, and scattered data on various platforms into a unified data architecture.
- The model utilizes the structural characteristics of mechanical products and is developed as a hierarchical digital mapping to better meet the application requirements.
- The model is easy to implement and can quickly iterate its structure to meet the response needs of cyber-physical systems flexibly, so the proposed method has the potential to be used by a series of emerging technologies, thereby enabling smart manufacturing.

The remainder of this paper is organized as follows. In Section 2, a review of product digital twins and related enabling technologies, including ontology and graph databases, is provided. The detailed proposed methodology is illustrated in Section 3. Based on the methodology proposed, a case for the development of a digital twin model about a helicopter and some of its transmission parts is presented in Section 4. Further discussions on the method and the case are provided in Section 5, and concluding remarks and an analysis of the limitations and future research directions are presented in Section 6.

2. Background and Related Work

Work on digital twins and its enabling technologies related to this study are presented in the following subsections. In the Section 2.1, the concept of digital twin and the state-of-the-art research on product digital twin modeling are introduced. In order to achieve universal data expression, the Section 2.2 introduces the ontology and its application in the digital model of mechanical products. To store the ontology model efficiently in the computer program, the graph database and its application in the ontology model are introduced in the Section 2.3.

2.1. Product Digital Twin

With the development of information and communication technology, the realization of the integration and management of multi-source heterogeneous dynamic data in the

product lifecycle has become an urgent problem. In this context, digital twin approaches have attracted the attention of scholars worldwide. The concept of a digital twin was originally proposed by Grieves and Githens [13,14]. It was originally defined as a digital copy that can abstractly express a physical device and be tested under real or simulated conditions. A digital twin aims to express relevant information about the device more clearly, thereby collating the data for higher-level analyses.

In recent years, the concept and characteristics of digital twins have been researched more in depth; in addition, their application scope has been broadened, including all types of mechanical products, and even larger entities or systems, such as workshops and factories, and processes, such as manufacturing and operation. Romero et al. [15] identified the differences and connections between the concepts of digital twins, digital shadows, and product avatars in detail and defined digital twins as a real-time virtual mapping of entities, processes, or systems in cyber-physical systems or cyber-physical production systems. Haag et al. [16] indicated that a digital twin is a comprehensive digital representation of a single product entity or a set of realistic models that can simulate the actual behavior of a physical entity. General Electric [17] defined digital twin as a software representation of a physical asset, system, or process designed to detect, prevent, predict, and optimize through real-time analytics to deliver business value. Tao et al. [18] developed a digital twin model from an initial three-dimensional (3D) structure to a five-dimensional structure. It includes physical entities, virtual models, service systems, twin data, and connections; the virtual model could be further decomposed into geometric, physical, behavior, and rule models, whereas twin data included the data from physical models, virtual models, service systems, domain knowledge, and fusion data. Thoben et al. [19] divided the product lifecycle into three stages—beginning of life, middle of life, and end of life—and studied the data and information among them. On this basis, Wellsandt et al. [20] indicated that the current MOL data for products that can be referred to mainly include product instance, stakeholders, and product environment information. West et al. [5,21] further highlighted that digital twins can enhance value co-creation for stakeholders by supporting data-based decision making.

Product digital twins are one of the main application directions of digital twin methods. Zhuang et al. [22] defined a product digital twin as a full-element reconstruction and digital mapping of the work progress and state of the product's physical entity in the virtual space, which is an integrated multi-physical, multi-scale, super-realistic, dynamic probability simulation model. As important existing system modeling and data integration methods, the characteristics of Systems Modeling Language (SysML), PLM, and digital twin are shown in Table 1.

According to the above comparison, SysML, as a graphical modeling language, aims to provide a standardized expression for complex systems. This is a different research direction than the product digital twin that need to realize the storage and management of specific data. Although both PLM and digital twin focus on the integration of product data, their characteristics are not the same: (a) As an enterprise-level business method, PLM's data is mainly used for internal and external collaboration, while the digital twin focuses on the product itself, and its data is mainly used to map physical entities; (b) in this context, PLM requires data consistency, traceability, and long-term archiving, while digital twins require more flexible data fusion methods that could support rapid response and iteration; (c) therefore, PLM is generally based on enterprise-level software solutions, and digital twins could be implemented through lighter modeling methods. In addition, the current research on product lifecycle data mainly focuses on physical entities rather than virtual models, and the convergence between product physical and virtual space is lacking [6]. Because of these differences in targets, requirements, and approaches, it would be complicated and inefficient to directly transform the PLM system to realize the product digital twin. In this context, PLM could be used as part of the data foundation of the digital twin. On this basis, how to implement a native modeling framework that could

integrate heterogeneous data, and is more lightweight, flexible, and fast iterable, remains to be further studied.

Some typical studies on product digital twin development are presented in Table 2.

Table 1. Comparison of product data integration and modeling methods.

	SysML [23]	PLM [24]	Product Digital Twin [10,22]
Concept	Standards from object management organizations to support the design, analysis, and verification of complex systems	A strategic business approach that integrates personnel, technical systems, and process management	A full-element reconstruction and digital mapping of the product's physical entity
Focus	Provide system engineers with a less software-oriented modeling language	Manage the entire product lifecycle to gain business efficiency and effectiveness	Simulate, monitor, diagnose, predict, and control the formation process, status, and behavior of products
Application	Complex system	Mainly include aviation, aerospace, automobile, and other equipment manufacturing fields [12]	Various products
Form	Modeling language	Enterprise-level solutions	Integrated digital model
Requirement	Model a wide variety of systems from different perspectives such as behavior, structure, or requirement	Information consistency, traceability, long-term archiving, and support internal and external collaborations	Multi-physics, multi-scale, hierarchical, integrated, and computable

Table 2. Typical product digital twin modeling work review.

Author	Method	Outcome
Schroeder et al. [25]	Automatic markup language	Model with a good cross-system interaction ability
Schluse et al. [26]	Integration of typical simulation algorithms	Experimental digital twin
Medini et al. [27]	Agent-based approach	Multi-agent system
Ullah et al. [28]	Semantic modeling	Computational virtual abstraction of complex manufacturing phenomena
Nafors et al. [29]	Virtual reality	Hybrid digital twin
Zheng et al. [30]	Tri-model-based approach	Generic cyber-physical system architecture
Bao et al. [31]	Product information unit	Heterogeneous data model for product development and manufacturing
Singh et al. [32]	Minimal data structure	Digital twin database management
Zheng et al. [33]	Multi-agent architecture	Manufacturing process digital twin
Li et al. [34]	Semantic modeling method	Integration between virtual entities
Fang et al. [35]	CAD model lightweight	Digital twin-oriented assembly model

These studies evaluated comprehensive digital expression methods for product physical entities from different perspectives. The modeling object of this study is a mechanical product, specifically a complex mechanical product composed of multiple parts, that is, a pure mechanical or electromechanical product with a multilevel assembly structure. For the digital twin modeling of these products, the methods in Table 1 still have some areas that can be further optimized: (a) most of the models developed by these methods are partial digital mappings oriented to their respective needs or to realize some of the functions for the digital twin rather than general full-element modeling. Therefore, when the application requirements change, the digital model can no longer support the new application; (b) these methods are oriented to general products, whereas mechanical products have

many characteristics in terms of structure and attributes. The use of these characteristics to optimize the digital twin modeling process and model quality requires further study.

2.2. *Ontology and Its Application in Mechanical Products*

Ontology is a general knowledge representation method that has been widely used by scholars to define conceptual models for diverse fields. In the 1990s, Gruber [36,37] introduced the concept of ontology into the informatics and artificial intelligence fields and defined it as a clear formal specification of a shared conceptual model. In essence, an ontology supports the management of field knowledge by describing the definitions, examples, and axioms of concepts in research fields. Compared with other knowledge representation methods such as frameworks and logic, ontology-based knowledge representation provides clear classification and hierarchical structure and has rich resource relationships, high computer understanding capabilities, and strong scalability [38,39].

As mechanical products have complex and coupled functions and structural relationships, applying ontology to product information modeling has become one of the research directions in digital design and manufacturing, such as engineering ontology for product design, product information model ontology, and product data semantic exchange ontology. Some scholars have attempted to use ontology for the management of multidisciplinary information. Eck et al. [40] proposed a semantic-based file storage system for the integration of mechanical, electrical, electronic, and software data in the electromechanical field. Based on causal graphs and Bayesian networks, Nagarajan et al. [41] proposed a metamodel that combines multiple domain sub-models to predict the impact of process parameters on product quality. Lu et al. [42] developed a product description framework based on semantic web technology and an ontology called ManuService. This approach addressed the conceptual description of products in a service-oriented business environment, promoting personalized manufacturing strategies in cloud manufacturing. The integration between digital twin and design ontology towards digital twin-driven product design has also received the attention of many scholars. Tao et al. [6] studied the detailed application method and framework of digital twin-driven product design, manufacturing, and service. Erkoyuncu et al. [43] developed an ontology-based digital twin design framework that can adapt to asset changes. Hu et al. [44] proposed a digital twin-driven improvement design method of drawing bench for brazing material; through the analysis based on the product ontology and simulation based on the digital twin model, a good improvement effect was achieved.

As information can be transformed into computer-understandable knowledge through ontology, it is also used for knowledge sharing and reuse of mechanical products. To address the problem of knowledge sharing and reuse in the process of complex product design, Luo et al. [45] proposed an ontology-based knowledge modeling method and ontology construction framework. Wu et al. [46] designed and developed an ontology-based knowledge representation model for complex products to organize knowledge resources and established a knowledge recommendation model based on vector space to promote the reuse of knowledge. Qin et al. [47] used hierarchical feature ontology and ontology mapping to improve the retrieval performance of heterogeneous product computer-aided design (CAD) models. Based on the enhancement of the main model, Yu et al. [48] extended the design process model, context model, product control structure model, and design principle model. They constructed an ontology to express the product and design information. Mandorli et al. [49] introduced an ontology analysis and feature classification based on the mechanical CAD system to represent the feature semantic levels and to enhance the cognitive understanding of digital models. Bao et al. [31] developed a product information unit model based on an ontology to express heterogeneous product data. Singh et al. [32] captured and modeled conceptual knowledge about the digital twin field based on an ontology model to map, query, and manage the database. Zheng et al. [33] defined the vocabulary and semantics related to the distributed agent based on the ontology and enabled the behavior model of the virtual entity to have logical capabilities such as

reasoning and autonomous decision-making, thereby supporting the digital twin modeling of the manufacturing process. Li et al. [34] proposed a semantic construction method based on a graph–object–attribute–point–role–relationship, and used OWL to describe information and realize the integration of heterogeneous virtual models. Yang et al. [50] further proposed a question-and-answer method for model-based systems engineering based on knowledge graph modeling.

In summary, ontology has advantages in heterogeneous information modeling, knowledge extraction, and sharing. It has been used by scholars to develop the required partial or cross-domain knowledge models and has achieved good application effects. However, few studies on the application of ontology in full-element expression and modeling of product digital twins are found in the literature, and the characteristics of complex mechanical products have not been fully utilized, which should be a focus of study.

2.3. Ontology Storage and Graph Database

Ontology storage patterns include memory-based storage patterns such as OWLim [51], text-based storage patterns such as Protégé [52], and database-based storage patterns. The first two patterns have limited scalability and are difficult to adapt to large amounts of data. Database-based storage patterns can be further divided into relational and non-relational databases. The first one is based on mature relational databases. However, in the context of large amounts of data, because the ontology model is essentially a graph model, its specific implementations have some problems, including the inherent flat relationship model of a two-dimensional table restricting the representation of complex semantics of the ontology, and excessive table size or complex maintenance [53].

The graph database originated from graph theory and is a non-relational database used to store entities and their relationships. As its organizational structure has a high degree of conformity with ontology, graph databases have attracted increasing attention in the field of ontology storage. In response to the challenges of supporting big data, Elbattah et al. [54] proposed a graph database-oriented storage and management method. They treated the ontology as a large graph to solve the limitations of the traditional relational model and verified the scalability and query efficiency of the method through a data subset. Pham et al. [55] expressed the constructed ontology model using a graph database as a storage mechanism for large-scale domain ontology, which was proven to be more efficient than the RDFS/owl method. Lampoltshammer et al. [56] proposed a graph database-based classification method for the ontology reasoning problem in remote sensing data classification, which was 80,000 times faster than the method based on Protégé. Aiming at the big data management problem in railway systems, Figueres-Esteban et al. [57] used graphic databases and ontologies to access and manage the data. Gong et al. [58] proposed a method based on graph database and domain ontology to solve the performance bottleneck in oilfield big data storage and search, which reduced the storage space by more than 13% and improved the retrieval efficiency by more than 30 times compared with relational databases.

In summary, the graph database has considerable potential in the expression and management of ontology models, owing to the advantages of its underlying structure. As an emerging technology, the graph database is also rarely seen in PLM-related research at present. In the context of digital twins, the development of a full-element ontology model based on graph database storage should be further studied.

3. Methodology

In this study, a hierarchical integrated modeling method for a digital twin is proposed in response to the need for full-element data fusion of mechanical products. The method provides a phased framework for digital twin modeling, and its goal is to build a comprehensive digital model that integrates various data in the entire lifecycle of mechanical products. Figure 1 shows the main process of the method, including object checking, structural modeling, feature modeling, data import, and data application. The method

starts from Phase 1; to meet the dynamic and scalable needs of digital twins, the modeling process in this method is not a one-time process but could continue to iteratively optimize the digital twin model as the entity evolves until the iteration process ends at Phase 5.

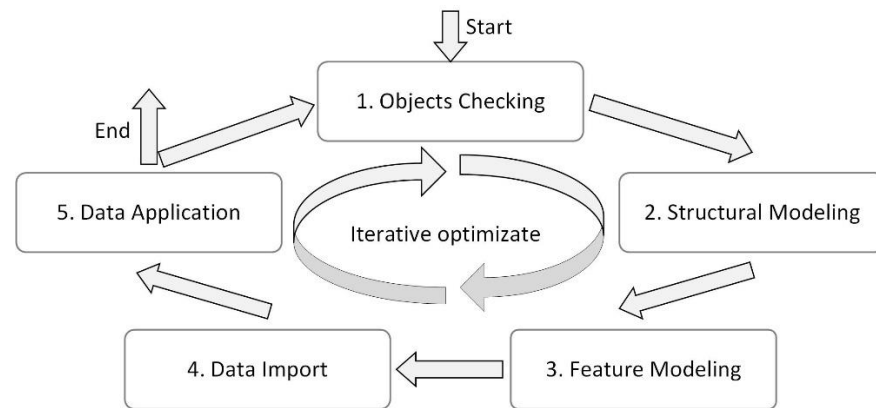


Figure 1. Framework of the proposed digital twin modeling method for complex mechanical products.

3.1. Objects Checking

In the first phase, we clarify the modeling scope of the digital twin according to the needs of mechanical products and collect the corresponding data. The digital twin model, as a digital mapping of the product entity in the entire lifecycle, may include various data generated in the design, manufacturing, operation, and maintenance stages, as shown in Figure 2.

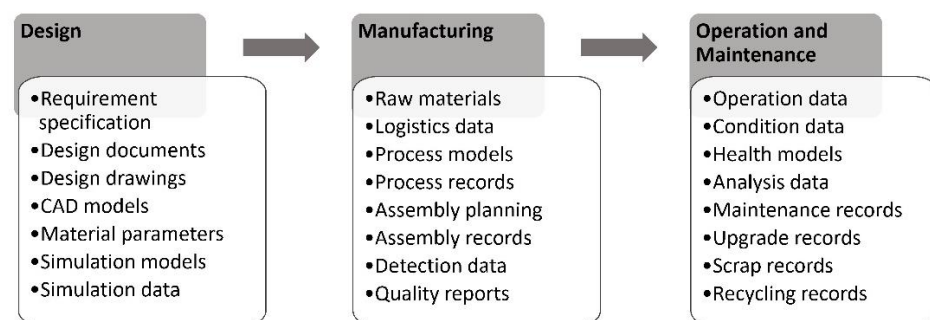


Figure 2. Schematic diagram of data generated by mechanical products at each stage of the lifecycle.

The goal of the digital twin approach is to develop a full-element map of the product, but this does not mean that all data must be included at the beginning. The development of a digital twin model is a dynamic and iterative process. The data that are temporarily not included in the current digital model can be improved in future optimizations. When this phase is finished, the modeling objects are checked, laying an important foundation for the subsequent steps.

3.2. Structural Modeling

The objective of the second phase is to use the mechanical product assembly structure to develop a hierarchical digital model as the fundamental structure of the entire digital twin model. First, we build a corresponding knowledge model based on the ontology method. The ontology model includes classes and instances. A class is a concept in a domain that represents a collection of instances of a certain type in the domain. An instance is a formal description of an individual in the field, including the name, class to which it belongs, and corresponding description. Owing to the need for design, manufacturing, and maintenance, mechanical products usually have a clear assembly structure, which is particularly evident in complex mechanical equipment. In view of this characteristic,

they are classified into two classes: parts and assemblies. A part refers to the smallest decomposition unit of a mechanical product structure in the digital twin, whereas an assembly is composed of multiple parts. The specific part or assembly instance in each mechanical product is defined as a corresponding instance.

Subsequently, orientation relationships are developed between the parts and assemblies. In an ontology model, the attribute relationship is the predicate connection of the class–class relationship or the instance–instance relationship, which functions as a logical rule definition. In generic mechanical products, the structural relationships mainly include composition, connection, and matching relationships. In view of these characteristics, the internal structural relationships of mechanical products are defined. The composition relationship from the assembly to the sub-assembly or part is defined as a “has assembly” or a “has parts” relationship, respectively. Connection and matching relationships include the “has connection” and “has match” relationships between assemblies and parts. A schematic of the hierarchical ontology model is shown in Figure 3.

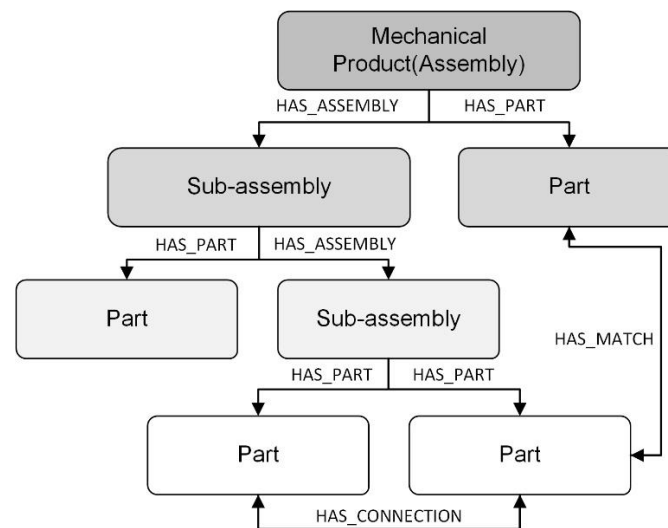


Figure 3. Diagram of the hierarchical structure ontology model of a mechanical product.

Furthermore, combined with the characteristics of the graph database using graph theory to store the relationship between instances, the developed ontology model is mapped into a graph database. First, the assembly and part are created as two types of nodes in the graph database. Then, the mechanical product and its sub-assemblies and parts are mapped to the nodes of the corresponding type. The relationships between the attributes in the ontology model are mapped to the directional relationships between nodes: “has assembly”, “has part”, “has connection”, and “has match” relationships are created between the developed assembly and part nodes. When the second phase is finished, all components of the mechanical product are mapped to the graph database hierarchically according to the assembly structure, and the corresponding structural relationships are defined using the edges between the nodes. Due to the objectivity of the mechanical product structure, the consistency of this part of the model can be guaranteed naturally. The developed structural model serves as the foundational architecture for the entire digital twin model.

3.3. Feature Modeling

The objective of the third phase is to model the feature data of the product on the basis of the developed structural model and to complete the construction of the digital twin model architecture. In this paper, feature refers to the product quality or characteristic to be modeled, parameter refers to specific data, and attribute is the term in ontology or the graph database.

In the first phase, the objects to be modeled are checked and the corresponding data are collected. According to the relationships between features and mechanical structure, these features can be divided into three categories: (1) features that belong to specific parts or assemblies and are directly related to the mechanical structure. For example, if a component is part of an assembly, then its corresponding two-dimensional or 3D drawing is also part of the drawing of the assembly; (2) features that independently belong to specific parts or assemblies are not related to the others, such as the name, number, or other basic data of a component; and (3) features that are public information, which are indirectly related to each part or assembly through transition attributes. For example, specific data such as the density, elastic modulus, or Poisson's ratio of a material are only indirectly related to corresponding parts or assemblies by the features of "has material."

Among the various features included in the digital twin of mechanical products, some can be clearly classified into the above three categories, such as basic information, 3D drawings, and material parameters. Others can be further decomposed into more sub-features. For example, the simulation model usually includes a 3D model, material parameters, and working condition parameters. These composited features can be decomposed and modeled in the above manner.

For the above three categories of features, different approaches are proposed for modeling, as shown in Figure 4. For the features of the first category, because of the structured logic inside the feature data, their attribute classes are defined in the ontology model. Relationships between instances in different levels of features are developed through attribute relationships, and then, they are similarly associated with corresponding assemblies or parts, as shown in Structured Feature A in Figure 4. Furthermore, they are mapped to the corresponding nodes and directional relationships between the nodes in the graph database. For the second category of features, as the feature data are only associated with a specific part or assembly, these features are added as attributes in the definition of the class of the corresponding part or assembly in the ontology model, such as the internal attribute Feature B of Assembly 1, as shown in Figure 4. This reduces the model complexity and modeling cost. Furthermore, they are mapped as attributes of the corresponding parts or assembly nodes in the graph database. The features of the third category are defined as independent attribute classes in the ontology model, and the data are stored in these public classes. This meets the need for public reuse of feature data and improves model consistency and storage efficiency. The parts or assemblies that use the feature data are associated with the public class through the "has...characteristic" attribute relationship, as Public Feature C is jointly owned by Assembly1, Part2, and Part3, as shown in Figure 4. Moreover, the public features are mapped to the corresponding feature nodes in the graph database, and orientation relationships are developed between the part or assembly nodes and the created feature nodes. The composite features are decomposed into the above three categories and then modeled. Composite Feature D in Figure 4 can be decomposed into two related Features E1 and E2, and then, an association with Assembly1 is developed, whereas Features E1 and E2 are also related to Part2 and Assembly1, respectively. Since a feature may have multiple expressions, in order to improve the consistency of the model, the terminology should be proofread and confirmed before adding operations to avoid adding the same feature repeatedly based on different descriptions. When the third phase is completed, the product digital twin model architecture is completed.

3.4. Data Import

The objective of the fourth phase is to import the corresponding data into the built product digital twin model architecture, that is, to store various feature data in the created nodes or attributes. Feature data can be divided into three categories according to the format. The first category of data can be stored directly as fields in the node attributes. For example, for material parameters including density and elastic modulus, relevant attribute fields can be directly designed in the material nodes and the corresponding data can be stored in them. For the second category of data, as their formats do not conform to the

field types in a graph database, they need to be structured in conjunction with external resources to be converted into field types that can be directly stored. For example, drawings of mechanical products can be stored in resources such as external servers or cloud disks. Then, the storage address is transferred to the corresponding field in the graph database as a uniform resource locator. The third category of data, such as design condition data or simulation results, can be stored in created nodes or attribute fields in a graph database. However, the data may form a mature management mode in external resources such as PLM systems, or their formats may be more suitable for other databases (e.g., relational databases). This category of data can also be processed for reference using the approach of the second category: first, it stores the data in suitable external resources, and then, it links the locator in a uniform manner. In addition, if the product feature data include unstructured data, it can be preprocessed first to improve the model quality of the digital twin. For example, transforming the obtained scans of drawings into original electronic digital ones through information recognition, converting image format documents into digital files, sorting or preprocessing various working condition data according to specific rules, and compressing files in complex formats.

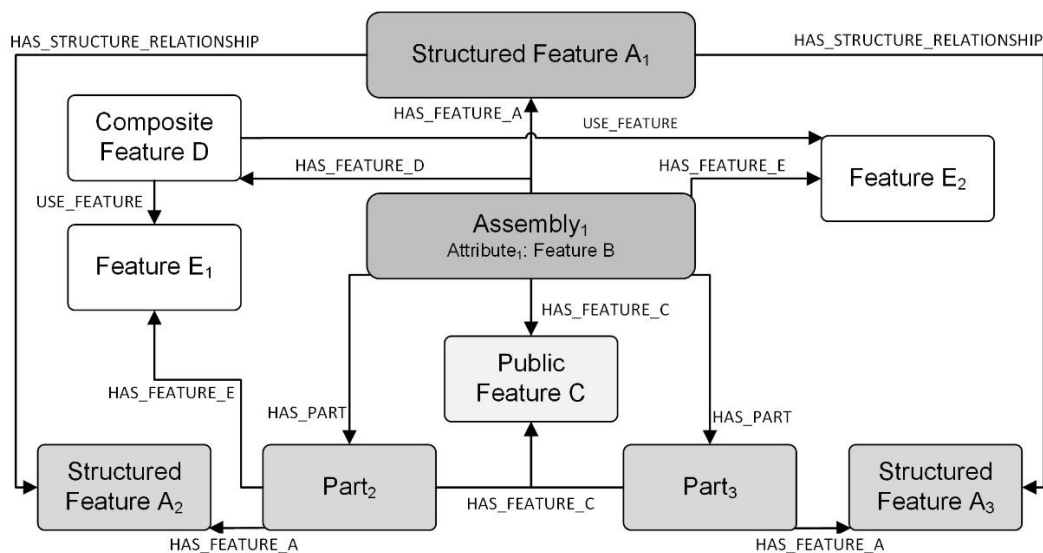


Figure 4. Diagram of the hierarchical feature ontology model of a mechanical product.

Through the discussed approaches, the data collected in the first phase are imported to complete the storage of the feature data in the nodes and attribute relationships. When the fourth phase is completed, the modeling of all of the mechanical product data is completed and the digital twin model of a mechanical product in a graph database is developed as expected.

3.5. Data Application

The development of a product digital twin model consists of effectively integrating and fusing the originally isolated and fragmented data in mechanical products, thereby empowering smart manufacturing. Therefore, the objective of this phase is to develop corresponding application tools according to the requirements and to combine them with various emerging intelligent technologies to fully utilize the value of product data. For example, we used a 3D model and real-time operating data in a digital twin to develop a visual interface to show the operating status of the product for real-time monitoring and scheduling by technicians. Alternatively, the simulation model and historical operating data in the digital twin are combined with artificial intelligence algorithms to perform health management and predictive maintenance of the product.

The development of a product digital twin model is a continuous iterative optimization process. If it is found that the current scope of data is not sufficiently large or that the

structure is unreasonable and further applications are restricted during usage, the modeling process cycle can be restarted from the first phase according to the new needs, such that the product digital twin model can be continuously improved.

4. Case Study

To verify the feasibility and effect of implementing the proposed method, the development process of a digital twin model prototype of an actual helicopter is presented in this section. The research background of this project is that aeronautical engineers already have plenty of data about the helicopter, including structural information, drawings, working condition data, simulation models, etc. However, these data are heterogeneous and scattered, so it is difficult to be directly used to map the physical entity. They also do not have much intention of deploying a PLM system from scratch because the organization already has an internal data archiving system, and now they just want to integrate the product data for mapping and analysis. Therefore, a lightweight and flexible modeling method is expected for the development of the helicopter digital twin model and the effective integration of heterogeneous data.

The goal of this case was to develop a hierarchical structure model that included the main systems of a helicopter and to develop a twin model containing feature data for some parts and assemblies to fully illustrate the details of the method. The developed digital twin model needs to integrate various heterogeneous data of the helicopter and its components to be directly used for further analysis, such as simulation. In this section, we conducted modeling strictly based on the steps of the proposed method and recorded the implementation process and the effects of each phase in detail.

4.1. Process and Results of Object Checking

In the first phase, the modeling scope of the product digital twin was clarified, and the relevant data were collected. Its implementation process corresponds to Phase 1 in the method. According to the goal of the case, the specific objects of the digital twin model of a helicopter developed in this section are listed in Table 3. In terms of assemblies and parts, 26 assemblies and two parts, including five main systems, were selected. In terms of feature data, seven types of features, including basic information, 3D drawings, finite element simulation models, simulation results, historical data, materials, and working conditions, and 10 specific pieces of data were selected.

Table 3. Checked modeling objects of the digital twin model in this case.

Category	Content
Structural data	Assembly (Helicopter, Drive system, Main reducer, Mid reducer, Tail reducer, Transmission shaft, Rotor System, Blade, Propeller hub, Sliding bearing, Landing gear system, Brake, Tail brace, Wheel, Shock absorber, Tail rotor system, Tail rotor propeller hub, Tail rotor blade, Fuselage, Mid fuselage, Cabin door, Stub wing, Workbench, Cockpit, Tail boom, and empennage) and Part (Inner wing and Outer wing)
Feature data	Basic information (3), 3D drawing (3), FE simulation model (1), Simulation results (1), Historical data (1), Material (2), Working condition (2)

4.2. Process and Results of Structural Modeling

The target of the second phase was to develop a digital hierarchical structure model of a helicopter based on its assembly structure. Its implementation process corresponds to Phase 2 in the method. In the process of ontology modeling, two classes of assembly and parts were defined. The 26 assemblies, including helicopters, transmission systems, and rotor systems, were created as instances of assemblies, and the inner and outer rings of the two parts were created as instances of parts. Then, according to the relationships between the assemblies and parts, the attribute relationships “has assembly,” and “has part,” were defined. Corresponding relationships were developed between the matching assembly

and the part instances. Subsequently, a structural knowledge model for the helicopter was obtained.

Based on this, the ontology model developed was mapped to a graph database. As the graph database uses graph theory to store the relationships between instances, its structure is very similar to that of the ontology model. First, the assembly and the part were created as two node types, and then, 26 assemblies and two parts were mapped to the corresponding types of nodes. Then, the attribute relationships in the ontology model were mapped to the orientation edges between the nodes, that is, the relationships “has assembly,” and “has part,” were created between the corresponding assembly and part nodes, which were abbreviated as “HAS_AS,” and “HAS_PA,” respectively. The implementation results of the second phase are shown in Figure 5, and the different levels of assembly are represented by colors from dark blue to light blue. At this time, a hierarchical structure model of a helicopter was created in a graph database. As shown in the figure, the assembly helicopter was in the first level; the transmission system, rotor system, and some other assemblies were in the second level; the main reducer, sliding bearing, and remainder assemblies were at the third level; and the inner ring and outer ring were at the fourth level.

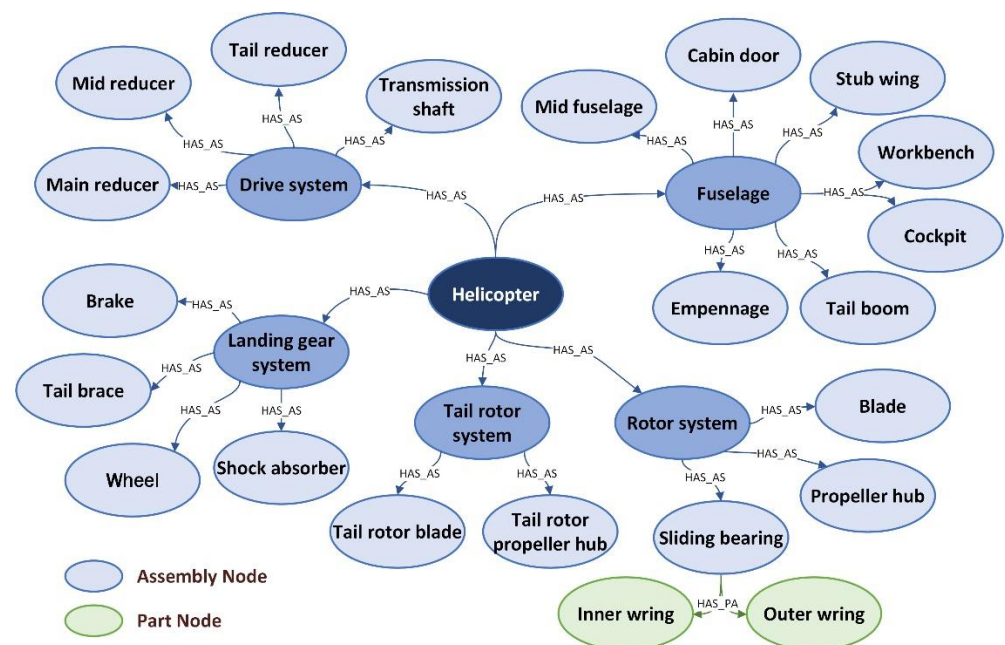


Figure 5. Schematic diagram of the hierarchical digital model about a helicopter in a graph database.

4.3. Process and Results of Feature Modeling

The objective of the third phase was to create the containers required for storage of the various feature data collected in the first phase and to complete the construction of the digital twin model architecture for a helicopter. The features were classified and processed according to the third phase of the proposed method, as shown in Table 4.

The first category contains the features associated with the mechanical structure, including the 3D drawings in this case. Three-dimensional drawing was defined as an attribute class, and sliding bearing 3D drawing, inner ring 3D drawing, and outer ring 3D drawing were created as the corresponding three instances. These three instances had the relationships “has 3D model” with the corresponding assembly and parts; in addition, according to the structural relationships between the sliding bearing, inner wring, and outer wring, corresponding subordination relationships were also provided between the three drawing instances.

The second category contained features that belonged to an assembly or part alone, including the basic information in this case. These data could be directly stored as an

attribute of the corresponding assembly or part, without the need to create additional instances or relationships.

Table 4. Product feature classification results.

Category	Class	Instance
The first category	3D drawing	Sliding bearing 3D drawing Inner wring 3D drawing Outer wring 3D drawing
The second category	Basic information	Sliding bearing info Inner wring info Outer wring info
The third category	Material	Steel1 Steel2
	Working condition	Hover Climb
The composite features	Historical data	Sliding bearing historical data
	FE simulation model Simulation results	Sliding bearing FE simulation model Sliding bearing simulation results

The third category contained common features that could be associated with multiple assemblies or parts, including the materials, working conditions, and historical data. In the ontology model, these three features were defined as attribute classes, and their specific content, including two materials, two working conditions, and one historical data, were created as instances. They were then associated with one or more corresponding assembly or part instances.

The composite features were decomposed and modeled, such as the finite element simulation model. The specific content of the finite element simulation model object was decomposed into 3D drawings, materials, working conditions, and simulation results. Among them, the classes and instances corresponding to the first three items were created in the preceding steps. As the simulation results might have multiple uses, the items were processed as a common feature to separately define a class and to create the corresponding instances. On this basis, the finite element simulation model was created as an attribute class, the simulation model of the sliding bearing was created as an instance, and the relationships between this instance and other decomposed instances were established.

The developed ontology model was mapped to a graph database, and the results are shown in Figure 6. After finishing this phase, the digital twin model architecture of the helicopter was completely developed.

4.4. Process and Results of Data Import

The objective of the fourth phase was to import data into the developed digital twin model architecture. Thus, various feature data were classified and processed.

The first category contained data that could be directly stored in the nodes, including the basic information and material parameters. Fields were added, and contents were stored in the assembly, part, and material nodes.

The second category contained data that could not be directly stored in the graph database because of their formats, including 3D drawings, finite element simulation models, and simulation results. These data were first stored in an external server or cloud disk, and then, the storage address was imported into the corresponding field of the nodes in the form of a uniform resource locator. For more complex application scenarios, dynamic interfaces with external resources can be designed. These data were integrated into the developed product digital twin model to meet the requirements of unified storage and recall.

The third category contained data that could be directly imported or processed with external resources, including historical data and working conditions. These data had a simple list format, and it was feasible to store them directly in the node. However, these data were stored in external resources such as PLM systems in advance, and relational

databases are more mature and efficient in managing these data. These data were stored on the server or cloud disk and could be called in the form of uniform resource locators, which conformed to the integrity of the digital twin model and improved the efficiency of data management.

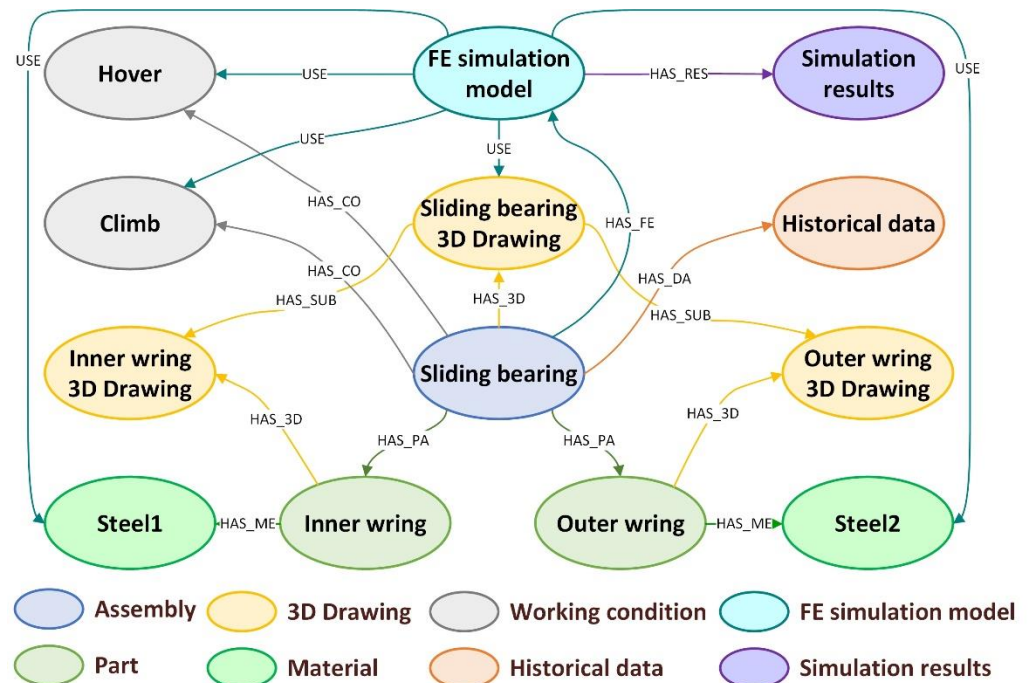


Figure 6. Schematic diagram of the digital twin model architecture about a helicopter in a graph database.

The above data were stored in the product digital twin model in a graph database. After finishing this phase, the development of a digital twin model for a helicopter was completed.

4.5. Process and Results of Data Application

The objective of the fifth phase was to develop application tools based on the digital twin model and to use the integrated data. Its implementation process corresponds to Phase 5 in the method. In this case, the need for engineers to quickly and conveniently perform finite element simulation on the sliding bearing according to the working condition data to obtain its force situations was urgent. The developed digital twin model integrates all of the data needed for simulation, such as product 3D drawings, material parameters, and working condition data. In this context, a simulation application tool with a graphical interface was further developed based on the digital twin model and was associated with the simulation software to perform calculations, as shown in Figure 7.

The simulation tool developed could automatically call the relevant data, enabling the original data with different formats and scattered storage to obtain a higher value after fusion. This avoids tedious work for engineers to manually collect data and develop simulation models, and lays the foundation for artificial intelligence, machine learning, and other algorithms to assist simulation optimization.

Thus, the objective of the case was achieved. A hierarchical digital model including the main systems of the helicopter containing characteristic data for some key components was developed. The model integrates various heterogeneous data of the helicopter and its components and can be directly used for further analysis, such as simulations. Compared with the existing PLM system, its characteristics include: (a) it could simulate and map the state of the product physical entity with input parameters; (b) focusing on the product

itself, it could quickly and lightly realize the integration and fusion of heterogeneous data, without the need for a customized software system; (c) it could easily respond and iterate. If the product structure or feature changes, engineers will track by adjusting the nodes or edges in the graph database, thereby recalling the required data.

Material Parameter				
Part/Param	Material Type	Density	Elastic Modulus	Poisson Ratio
Inner Wring	steel1	7.7E-9	210000	0.28
Coating	paintcoat	3.0E-9	250000	0.22
Outer Wring	steel2	7.8E-9	200000	0.29
Coating	paintcoat	2.0E-9	120000	0.15

Calculation Parameter								
Step	1 <input checked="" type="checkbox"/>	2 <input checked="" type="checkbox"/>	3 <input checked="" type="checkbox"/>	4 <input checked="" type="checkbox"/>	5 <input type="checkbox"/>	6 <input checked="" type="checkbox"/>	7 <input type="checkbox"/>	
Name	ground	hover	fixed-point	ide flight	climb	special	vel flight	Centrifugal Force
Duration	0.5	1.5	1.0	0.5	0.5	0.5	2.5	600.0
Sine Load	5500.0	1648.0	5254.0	4671.0	1865.0	4681.0	896.0	Coefficient A
Basic Load	1543.0	-2037.0	-3545.0	-1572.0	-1748.0	-1257.0	-2875.0	3.5E-5
Rotation	8.5	8.5	10.0	6.5	7.5	15.0	10.0	Coefficient B
								-0.805

Figure 7. Tool from the helicopter sliding bearing simulation system based on the digital twin.

5. Discussion

The digitization and intelligence of mechanical products face many new challenges. Massive amounts of new data are collected during product design, manufacturing, and operation, but their isolation and fragmentation restrict further mining of their value. The integration of these data has become a key issue to be solved in the information era. With the development of digital twins, full-element reconstruction in virtual space is regarded as an important data fusion approach. This is theoretically reasonable, but the implementation of digital twin technology requires further study. In this context, a hierarchical modeling method for digital twin models of mechanical products was proposed in this study.

First, this method provides a feasible modeling framework and develops a full-element digital twin model for mechanical products. Compared with some partial models or frameworks for respective needs in the existing literature, such as computational virtual abstraction of complex manufacturing phenomena [28] and digital twin-oriented assembly model [35], this is a native product digital twin modeling method rather than an extension of the original simulation model. This method uses ontology to structure various types of data into computer-understandable descriptions. For example, in this case, various data including complex structural relationships between assemblies and parts, drawings, models, parameters, and other feature data with completely different formats were transformed into attribute instances and their relationships through ontology modeling. In order to elaborate on the modeling process in an appropriate space, some typical features were selected and connections between systems were simplified to links in the case study. In actual projects, more data types and interfaces are usually necessary, which requires more case-based research work in the future. This is also the main reason why the developed model is called a digital twin prototype. Overall, the developed model serves as a data center for the entire lifecycle of the product or is integrated into existing data management systems, integrating the previously isolated, fragmented, and scattered data on various platforms into a unified data architecture. By abstracting the essential content and relationships of data, a formally unified and computer-understandable model was developed, which provides feasibility for data fusion and further applications.

Second, the digital model developed in this method is hierarchical and more suitable for mechanical products. The product digital twin modeling methods in the existing literature, such as generic cyber-physical system architecture [30] and heterogeneous data model for product development and manufacturing [31], are for general products and do not utilize the structural characteristics of mechanical products. In addition, the general domain model based on ontology is typically flat and decentralized in these studies. However, for mechanical products, data calling requirements are often clustered in the structural dimension. For example, in the case of a sliding bearing finite element simulation, the required data mainly include various feature data related to the assembly and its parts. Cross-domain calls in the structural dimension rarely occur. For example, for the sliding bearing simulation of the rotor system, the data of the landing gear in another system are generally not used. In order to describe the model clearly and concisely, the main systems of the helicopter and related parts of the sliding bearing are modeled in this case. As not many components were involved, quantitatively comparing efficiency improvements is difficult, which needs to be further expanded in future research based on software engineering. All in all, the model developed by the proposed method utilizes the structural characteristics of mechanical products, and hierarchical digital mapping can better meet the application requirements.

Third, this method stores the developed ontology model in a graph database, which is simple, direct, and efficient. In the existing research, owing to the limitation of the development stage of information technology, scholars have been devoted to studying how to store ontology models based on text or relational databases for a long time, such as ManuService ontology [46] and the ontology-based CAD model retrieval method [47]. However, the ontology model is essentially a graph model, and the inherent structure of the two-dimensional table restricts the representation of complex semantics in the ontology and the efficient storage of large amounts of data. With the rapid development of new non-relational databases, using graph databases to directly store ontology models has become possible, but their application is still rare in the field of digital twins. This method proposes a means to map the developed ontology model to the graph database and to demonstrate it in this case. Although the underlying structure of the existing graph databases are similar, many differences still exist in specific operations, which should be noted by researchers who need to replicate and apply this method. In general, the results show that the graph database can support the storage and management of the product digital twin model using this method.

In addition, compared with the common large-scale or customized software in the field of product data management, the proposed method is lightweight, flexible, and easy to implement. Following these steps, engineers with programming capabilities can quickly and accurately develop the required product digital twin model in a graph database, thereby integrating the existing isolated, fragmented data. The integrated data can be efficiently connected and used in computer programs and can be further used for digital or intelligent applications such as monitoring, simulation, and forecasting. Additionally, this method can be iteratively optimized, and its scalability can meet the dynamic and evolutionary requirements of the digital twin model. The development of the product digital twin model is not a one-time process but a continuous expansion and optimization as the received data and usage requirements change. This method realizes looping and iteration from the framework level, and the selected ontology and graph database have good dynamic expansion performances. When the product digital twin model needs to integrate more data to serve new needs and to add corresponding nodes, relationships, and attributes according to this method, the digital twin model is updated to the new state of the physical entity.

From the management perspective, the data integration and iterative optimization supported by this method can also provide some inspiration for the management and sharing of product data. The ideal product digital twin model is unique and dynamic; in its entire life cycle, only one digital model continuously evolves and integrates all data,

rather than multiple isolated models of various stages assembled. The product digital twin model developed by this method, as a digital map integrating all elements, can be used as the only digital model in the entire lifecycle. From design and manufacturing to operation and maintenance, different participants only need to add new elements to the same digital twin model in the graph database for updating. In this context, the data flow between the original isolated models is no longer a problem because they have been aggregated into a digital model at this time, breaking the barriers between data islands. For participants at various stages, such as designers, processors, and maintainers, they no longer transfer data but cooperate to develop a complete product digital model.

6. Conclusions and Future Works

With the development of information and communication technology, a large amount of multi-source heterogeneous data with high information density and utilization value are generated, but isolated and fragmented data are difficult to use effectively. In this context, digital twins have been increasingly researched as they enable the user to rebuild all of the elements of a physical entity in a virtual space. However, the development of a digital twin model is still in its infancy, and most of the models currently developed are partial mappings oriented to specific needs and do not take advantage of the characteristics of mechanical products. In this study, a hierarchical integrated modeling method for digital twin models of mechanical products is proposed. By combining it with ontology technology, this method develops a native, full-element digital twin model for mechanical products. The model could serve as a data center for the entire lifecycle of the product or can be integrated with existing data management systems, integrating the previously isolated, fragmented, and scattered data on various platforms into a unified data architecture. In addition, the model utilizes the structural characteristics of mechanical products and is developed as a hierarchical digital mapping to better meet the application requirements. The proposed method has the potential to be used by more data technologies, thereby enabling smart manufacturing.

Although the proposed method can realize the development of a digital twin model at the theoretical level and prototype system, it still has some limitations, and the related enabling technologies need to be further studied in the future. For example, during the evolution of the digital twin model, direct modification or deletion in the ontology model may result in data loss and difficulty in traceability. To meet the high reliability and traceability requirements of mechanical data, a model evolution and version management mechanism must be studied. Another example is that the stakeholders of a mechanical product usually do not completely trust each other. Therefore, the development of a secure cooperative modeling mechanism for a single digital model remains to be explored. In addition, further research is required to explore how SysML or other modeling languages can be used to standardize the expression of product digital twins and, for data that cannot be stored directly in a graph database but temporarily imported by a link, a more efficient integrated interface to achieve better fusion can be developed.

Author Contributions: Conceptualization, M.Z. and L.T.; methodology, M.Z.; software, M.Z.; validation, L.T. and M.Z.; resources, L.T.; data curation, M.Z.; writing—original draft preparation, M.Z.; writing—review and editing, M.Z. and L.T.; visualization, M.Z.; funding acquisition, L.T. 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 51675299; the Beijing Municipal Natural Science Foundation, grant number 3182012; the National Key Research and Development Project of China, grant number 2018YFB1700604; and the National Key Research and Development Project of China, grant number 2020YFB1709103.

Data Availability Statement: Not applicable.

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

References

1. Vrchota, J.; Pech, M. Readiness of Enterprises in Czech Republic to Implement Industry 4.0: Index of Industry 4.0. *Appl. Sci.* **2019**, *9*, 5405. [CrossRef]
2. Tan, Y.; Yang, W.; Yoshida, K.; Takakuwa, S. Application of IoT-aided simulation to manufacturing systems in cyber-physical system. *Machines* **2019**, *7*, 2. [CrossRef]
3. Pech, M.; Vrchota, J.; Bednář, J. Predictive Maintenance and Intelligent Sensors in Smart Factory: Review. *Sensors* **2021**, *21*, 1470. [CrossRef]
4. Gupta, S.K. Towards Next Generation Pandemic Proof Factories. *Smart Sustain. Manuf. Syst.* **2020**, *4*, 20200058. [CrossRef]
5. West, S.; Stoll, O.; Meierhofer, J.; Züst, S. Digital Twin Providing New Opportunities for Value Co-Creation through Supporting Decision-Making. *Appl. Sci.* **2021**, *11*, 3750. [CrossRef]
6. Tao, F.; Cheng, J.; Qi, Q.; Zhang, M.; Zhang, H.; Sui, F. Digital Twin-Driven Product Design, Manufacturing and Service with Big Data. *Int. J. Adv. Manuf. Technol.* **2018**, *94*, 3563–3576. [CrossRef]
7. Wuest, T.; Weimer, D.; Irgens, C.; Thoben, K.D. Machine learning in manufacturing: Advantages, challenges, and applications. *Prod. Manuf. Res.* **2016**, *4*, 23–45. [CrossRef]
8. Modeling, Simulation, Information Technology & Processing Roadmap. Available online: https://www.nasa.gov/pdf/501321main_TA11-ID_rev4_NRC-wTASR.pdf (accessed on 16 March 2021).
9. Gartner Top 10 Strategic Technology Trends for 2016/2017/2018/2019. Available online: <https://www.gartner.com/smarterwithgartner/> (accessed on 16 March 2021).
10. Tao, F.; Qi, Q. Make More Digital Twins. *Nature* **2019**, *573*, 490–491. [CrossRef]
11. Leonova, A.E.; Karpov, V.I.; Chernyy, Y.Y.; Romanova, E.V. Transformation PLM-Systems into the Cyber-Physical Systems for the Information Provision for Enterprise Management. *Int. Conf. Cyber-Phys. Syst. Control* **2019**, *6*, 431–439. [CrossRef]
12. Mulla, F.M.; Kulkarni, V.N.; Gaitonde, V.N.; Kotturshettar, B.B. PLM as a tool for collaboration in aerospace industries-A review. *AIP Conf. Proc.* **2021**, *2316*, 020004. [CrossRef]
13. Githens, G. Product Lifecycle Management: Driving the Next Generation of Lean Thinking. *J. Prod. Innov. Manag.* **2007**, *24*, 278–280. [CrossRef]
14. Grieves, M.W. *Virtually Perfect: Driving Innovative and Lean Products through Product Lifecycle Management*; Space Coast Press: Cocoa Beach, FL, USA, 2011.
15. Romero, D.; Wuest, T.; Harik, R.; Thoben, K.D. Towards a Cyber-Physical PLM Environment: The Role of Digital Product Models, Intelligent Products, Digital Twins, Product Avatars and Digital Shadows. *IFAC-PapersOnline* **2020**, *53*, 10911–10916. [CrossRef]
16. Haag, S.; Anderl, R. Digital twin—Proof of concept. *Manuf. Lett.* **2018**, *15*, 64–66. [CrossRef]
17. What Is a Digital Twin. Available online: <https://www.ge.com/digital/blog/what-digital-twin> (accessed on 13 September 2021).
18. Tao, F.; Zhang, M.; Liu, Y.; Nee, A.Y. Digital Twin Driven Prognostics and Health Management for Complex Equipment. *Cirp Ann.-Manuf. Technol.* **2018**, *67*, 169–172. [CrossRef]
19. Thoben, K.D.; Wiesner, S.; Wuest, T. 'Industrie 4.0' and Smart Manufacturing—A Review of Research Issues and Application Examples. *Int. J. Autom. Technol.* **2017**, *11*, 4–16. [CrossRef]
20. Wellsandt, S.; Hribernik, K.; Thoben, K.D. Sources and characteristics of information about product use. *Procedia CIRP* **2015**, *36*, 242–247. [CrossRef]
21. West, S.; Meierhofer, J.; Stoll, O.; Schweiger, L. Value propositions enabled by digital twins in the context of servitization. In *Advanced Services for Sustainability and Growth*; Aston Business School, Aston University: Birmingham, UK, 2020; p. 152.
22. Zhuang, C.; Liu, J.; Xiong, H.; Ding, X.; Liu, S.; Weng, G. Connotation, Architecture and Trends of Product Digital Twin. *Comput. Integr. Manuf. Syst. CIMS* **2017**, *23*, 753–768. [CrossRef]
23. Wolny, S.; Mazak, A.; Carpella, C.; Geist, V.; Wimmer, M. Thirteen years of SysML: A systematic mapping study. *Softw. Syst. Modeling* **2020**, *19*, 111–169. [CrossRef]
24. Corallo, A.; Latino, M.E.; Lazoi, M.; Lettera, S.; Marra, M.; Verardi, S. Defining product lifecycle management: A journey across features, definitions, and concepts. *Int. Sch. Res. Not.* **2013**, *2013*, 170812. [CrossRef]
25. Schroeder, G.N.; Steinmetz, C.; Pereira, C.E.; Espindola, D.B. Digital Twin Data Modeling with Automationml and a Communication Methodology for Data Exchange. *IFAC-PapersOnline* **2016**, *49*, 12–17. [CrossRef]
26. Schluse, M.; Atorf, L.; Rossmann, J. Experimentable Digital Twins for Model-Based Systems Engineering and Simulation-Based Development. In Proceedings of the 2017 11th Annual IEEE International Systems Conference, Montreal, QC, Canada, 24–27 April 2017; pp. 1–8. [CrossRef]
27. Medini, K.; Romero, D.; Wuest, T. Developing a Multi-Agent System to Support Multi-Variant Production Ramp-Up Management. *Smart Sustain. Manuf. Syst.* **2021**, *1*, 129–147. [CrossRef]
28. Ullah, A.S. Modeling and Simulation of Complex Manufacturing Phenomena Using Sensor Signals from the Perspective of Industry 4.0. *Adv. Eng. Inform.* **2019**, *39*, 1–13. [CrossRef]
29. Näfors, D.; Berglund, J.; Gong, L.; Johansson, B.; Sandberg, T.; Birberg, J. Application of a Hybrid Digital Twin Concept for Factory Layout Planning. *Smart Sustain. Manuf. Syst.* **2020**, *4*, 20190033. [CrossRef]
30. Zheng, P.; Sivabalan, A.S. A Generic Tri-Model-Based Approach for Product-Level Digital Twin Development in a Smart Manufacturing Environment. *Robot. Comput.-Integr. Manuf.* **2020**, *64*, 101958. [CrossRef]

31. Bao, Q.; Zhao, G.; Yu, Y.; Dai, S. Product Information Units Modeling Oriented to Digital Twin. *IOP Conf. Ser. Earth Environ. Sci.* **2021**, *726*, 012014. [[CrossRef](#)]
32. Singh, S.; Shehab, E.; Higgins, N.; Fowler, K.; Reynolds, D.; Erkoyuncu, J.A.; Gadd, P. Data management for developing digital twin ontology model. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2020**, *235*, 2323–2337. [[CrossRef](#)]
33. Zheng, X.; Psarommatis, F.; Petralli, P.; Turrin, C.; Lu, J.; Kiritsis, D. A Quality-Oriented Digital Twin Modelling Method for Manufacturing Processes Based on A Multi-Agent Architecture. *Procedia Manuf.* **2020**, *51*, 309–315. [[CrossRef](#)]
34. Li, H.; Lu, J.; Zheng, X.; Wang, G.; Kiritsis, D. Supporting Digital Twin Integration Using Semantic Modeling and High-Level Architecture. *IFIP Int. Conf. Adv. Prod. Manag. Syst.* **2021**, *633*, 228–236. [[CrossRef](#)]
35. Fang, L.; Liu, Q.; Zhang, D. A Digital Twin-Oriented Lightweight Approach for 3D Assemblies. *Machines* **2021**, *9*, 231. [[CrossRef](#)]
36. Gruber, T.R. A Translation Approach to Portable Ontology Specifications. *Knowl. Acquis.* **1993**, *5*, 199–220. [[CrossRef](#)]
37. Gruber, T.R. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *Int. J. Hum. Comput. Stud.* **1995**, *43*, 907–928. [[CrossRef](#)]
38. Wu, K.; Huang, L.; Li, B.; Shi, J. Failure Management and Support System of Mechanical Parts Based on Failure Knowledge-Base. *J. Graph.* **2019**, *40*, 623–630. [[CrossRef](#)]
39. Zhao, J.C.; Huang, L.P.; Tian, L.; Wu, K.Q. Knowledge System for Helicopter Transmission Design Based on Ontology. In Proceedings of the 2017 International Conference on Management Engineering, Software Engineering and Service Sciences, Wuhan, China, 14–16 January 2017; pp. 321–325. [[CrossRef](#)]
40. Eck, O.; Schaefer, D. A Semantic File System for Integrated Product Data Management. *Adv. Eng. Inform.* **2011**, *25*, 177–184. [[CrossRef](#)]
41. Nagarajan, H.P.; Panicker, S.; Mokhtarian, H.; Remy-Lorit, T.; Coatanéa, E.; Prod'hon, R.; Jafarian, H.; Haapala, K.R.; Chakraborti, A. Graph-Based Metamodeling for Characterizing Cold Metal Transfer Process Performance. *Smart Sustain. Manuf. Syst.* **2019**, *3*, 169–189. [[CrossRef](#)]
42. Lu, Y.; Wang, H.; Xu, X. Manuservice Ontology: A Product Data Model for Service-Oriented Business Interactions in a Cloud Manufacturing Environment. *J. Intell. Manuf.* **2019**, *30*, 317–334. [[CrossRef](#)]
43. Erkoyuncu, J.A.; del Amo, I.F.; Ariansyah, D.; Bulka, D.; Roy, R. A design framework for adaptive digital twins. *CIRP Ann.* **2020**, *69*, 145–148. [[CrossRef](#)]
44. Hu, B.; Feng, Y.; Gao, Y.; Zheng, H.; Tan, J. A Digital Twin-Driven Improved Design Approach of Drawing Bench for Brazing Material. *Int. Des. Eng. Tech. Conf. Comput. Inf. Eng. Conf.* **2019**, *59186*, V02AT03A010. [[CrossRef](#)]
45. Luo, X.; Zhou, Y.; Liu, B.; Wang, Y. Research on Ontology-Based Knowledge Modeling of Design for Complex Product. In *Computational Intelligence and Intelligent Systems*; Springer: Singapore, 2015; pp. 665–671. [[CrossRef](#)]
46. Wu, Z.; Liu, H.; Goh, M. Knowledge Recommender System for Complex Product Development Using Ontology and Vector Space Model. *Concurr. Eng. Res. Appl.* **2019**, *27*, 347–360. [[CrossRef](#)]
47. Qin, F.; Gao, S.; Yang, X.; Li, M.; Bai, J. An Ontology-Based Semantic Retrieval Approach for Heterogeneous 3d Cad Models. *Adv. Eng. Inform.* **2016**, *30*, 751–768. [[CrossRef](#)]
48. Yu, C.; Zhang, F.P.; Butt, S.I.; Yan, Y.; Lv, W. Ontoimm: An Ontology for Product Intelligent Master Model. *Appl. Sci.* **2019**, *9*, 2553. [[CrossRef](#)]
49. Mandorli, F.; Borgo, S.; Wiejak, P. From Form Features to Semantic Features in Existing Mcad: An Ontological Approach. *Adv. Eng. Inform.* **2020**, *44*, 101088. [[CrossRef](#)]
50. Yang, P.; Lu, J.; Feng, L.; Wu, S.; Wang, G.; Kiritsis, D. A Knowledge Management Approach Supporting Model-Based Systems Engineering. In *Trends and Applications in Information Systems and Technologies*; Springer: Cham, Switzerland, 2021; pp. 581–590. [[CrossRef](#)]
51. Owlím. Available online: <https://www.w3.org/2001/sw/wiki/Owlim> (accessed on 16 March 2021).
52. Protégé. Available online: <https://protege.stanford.edu/> (accessed on 16 March 2021).
53. Huang, Q.; Qian, Y.; Yuan, Q.; Lu, J. Research on the Storage Model of OWL Ontologies Based on Graph Databases. *J. China Soc. Sci. Tech. Inf.* **2019**, *38*, 310–321. [[CrossRef](#)]
54. Elbattah, M.; Roushdy, M.; Aref, M.; Salem, A.B.M. Large-Scale Ontology Storage and Query Using Graph Database-Oriented Approach: The Case of Freebase. In Proceedings of the 2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS), Cairo, Egypt, 12–14 December 2015; pp. 39–43. [[CrossRef](#)]
55. Pham, P.; Nguyen, T.; Do, P. Computing Domain Ontology Knowledge Representation and Reasoning on Graph Database. In *Information Systems Design and Intelligent Applications*; Springer: Singapore, 2018; pp. 765–775. [[CrossRef](#)]
56. Lampoltshammer, T.J.; Wiegand, S. Improving the Computational Performance of Ontology-Based Classification Using Graph Databases. *Remote Sens.* **2015**, *7*, 9473–9491. [[CrossRef](#)]
57. Esteban, M.F.; Hughes, P.; El Rashidy, R.; Van Gulijk, C. Manifestation of Ontologies in Graph Databases for Big Data Risk Analysis. In Proceedings of the Annual European Safety and Reliability Conference, Trondheim, Norway, 17–21 June 2018; pp. 3189–3193. [[CrossRef](#)]
58. Gong, F.; Ma, Y.; Gong, W.; Li, X.; Li, C.; Yuan, X. Neo4j Graph Database Realizes Efficient Storage Performance of Oilfield Ontology. *PLoS ONE* **2018**, *13*, e0207595. [[CrossRef](#)] [[PubMed](#)]