Peer-to-Peer Networking and Applications (2022) 15:1638–1652
https://doi.org/10.1007/s12083-022-01311-w

A novel manufacturing service model transformation method based on product lifecycle

Tao Ding[1,2] · Guangrong Yan[1,2] · Zhenggan Zhou[1] · Yi Lei[1,2]

Received: 15 April 2021 / Accepted: 24 February 2022 / Published online: 17 March 2022

Abstract
With the increase of personalized customization and collaborative production requirements, more and more manufacturing enterprises virtualize and publish their resources and capabilities as cloud services for sharing. However, due to the lack of a general modelling method in the sharing process, data cannot be interpenetrated among different life cycle stages. Also, models built in a lifecycle stage cannot be transformed and propagated to other stages. To alleviate these drawbacks, in this paper, a novel service model transformation method based on product lifecycle is designed and developed to model and transform manufacturing services among different life cycle stages efficiently and accurately. Specifically, based on the discussion of the business model of life cycle service in cloud manufacturing environment, a novel service model transformation method which includes general and view service transformation is proposed and elaborated. Then, the life cycle service model is established mathematically, and eight transformation operators are summarized and their mathematical definitions are given in detail. Meanwhile, the transformation logic process and change propagation are studied. The proposed method is superior to previous methods in that: 1) the model established in this paper is a generic model which can run through different life cycle stages, including both general and personalized data; 2) the eight operator definitions cover most of the operation types in the model transformation process, which greatly improves the operability of the model automatic transformation; 3) the establishment of change propagation mechanism ensures the accuracy of model synchronization when data changes. The successful application in an instrument enterprise demonstrates the rationality and effectiveness of the proposed methodology.

Keywords Product lifecycle · Cloud manufacturing · Service model · Transformation operator · Logic Rules · Change propagation

1 Introduction

With the development of computer and information technology, consumers have put forward higher requirements for product quality and delivery time. At the same time, manufacturing products usually contain multi-level assembly structures and complex process routes, and the production process requires the collaboration of multiple enterprises. However, there are great differences in data definitions and semantic syntax between various corporations, which lead to low efficiency of production collaboration and inconsistent data among enterprises. Facing the challenges of rapid production response and efficient resource utilization, manufacturers have to spend much time on organizing and coordinating production resources distributed in different places. To perform large scale collaborative manufacturing, cloud manufacturing is proposed based on internet and aims to build a new networked production organization mode, which can provide users with flexible and on-demand shared manufacturing resources [1]. Driven by the virtualization and servitization of manufacturing resources, the speed of digital transformation of manufacturing enterprise has been significantly accelerated [2].

✉ Guangrong Yan
yangr@buaa.edu.cn

Tao Ding
oceanwaving@buaa.edu.cn

Zhenggan Zhou
zzhenggan@buaa.edu.cn

Yi Lei
leiy@buaa.edu.cn

[1] School of Mechanical Engineering and Automation, Beihang University, Beijing, China

[2] The National Engineering Laboratory of Intelligent, Collaborative Manufacturing Technology and Application, Beijing, China

Product lifecycle refers to the whole process of production from demand analysis, design process, manufacturing, operation to maintenance and scrap [3]. The purpose of PLM (product lifecycle management) is not only to realize efficient management of the information and knowledge but also to achieve data integration from all involved stages of product lifecycle, which helps to improve the profitability and competitiveness of enterprises [4]. Due to the large temporal and spatial distance in various stages of the whole life cycle, production data of services are usually stored in heterogeneous information systems, resulting in model non-interoperability and data inconsistency among stages [5]. In order to fully manage model data of whole life cycle, the closed-loop PLM, which regards the product lifecycle stages as an end-to-end whole, has been proposed and developed to maintain data integrity and traceability throughout the whole life cycle stages [6]. However, the main limitations of these approaches are that they interpenetrate data among different lifecycle stages mainly based on integration and lack of specific transformation methods. As the service data in different lifecycle stages has characteristics of modularization and componentization [7], this would lead to inaccurate and time-consuming interpenetration results.

In this paper, we focus on the analysis of data characteristics in different lifecycle stages. The proposed approach explored and extracted the general and individual data to establish a generic model and transformation method, so as to realize the interpenetration at model level. Furthermore, to effectively cope with model transformation at different lifecycle stages, we summarized and defined eight operators, based on which the model transformation can be executed more instructively. In addition, we also studied the model propagation mechanism when data changes.

In summary, the major contributions of this paper are threefold:

1. A generic service model and transformation method is established to realize data interpenetration at model level.
2. Eight transformation operators are summarized and mathematically defined, based on which the model transformation can be carried out more instructively.
3. The model propagation mechanism is proposed when data changes to solve the synchronization problem of life cycle model.

The remainder of this paper is organized as follows. Section 2 gives a brief review of the related research. In Sect. 3, the business model of life cycle manufacturing services in cloud manufacturing is analyzed, and the novel service model transformation method based on general and view service is elaborated. Section 4 presents the definition of the life cycle service model, studies the eight transformation operators, and summarizes the logic process and change propagation which the model transformation follows. Section 5 shows a case study and prototype system applied in an instrument enterprise, which illustrates the feasibility and effectiveness of the proposed methodology. Section 5.2 concludes the whole paper.

2 Related work

In the industrial field, the product lifecycle refers to all life cycle stages of the whole production and application process. In order to ensure the accuracy and consistency of data in the whole life cycle, manufacturing data in different life cycle stages or various enterprises need to be modeled and searched uniformly [8]. At the beginning of research, the attributes and relationships of manufacturing resources, including basic attributes, usage attributes, and state attributes, are main the consideration components for modelling, but these are mostly static attributes [9]. Subsequently, multi-granularity resources (including workflow, activity and resource) are gradually taken into account to model services over different granularities [10]. With the deepening of research, dynamic attributes of resources are gradually considered, and the real-time operation data are also fed back to the service model [11]. To consider the diversity and heterogeneity of resource data, several novel spectral clustering methods are proposed for data partitioning [12, 13], which reduces the complexity of resource modelling. To ensure data consistency, the ontology semantic method is introduced into modelling and matching [14], and some ontology languages, such as OWL, OWL-S, are also applied to describe the model [15, 16]. Moreover, in order to meet the requirements of personalized customization, service modelling by means of considering the relevant relationship between demands and services has emerged as one of the most primary research directions [17, 18]. In addition, some researchers present new modelling method of data and information expansion based on the source model, such as extending [19] or enhancing [20] 3D model data to expand information of other life cycle stages, thereby improving the universality of the source service model. However, these methods mainly modelling based on the data of a certain stage, and then expand or enhance data to other stages, which cannot achieve global interpenetration at model level.

On the basis of manufacturing service modelling, research on data integration of manufacturing service model is also developed simultaneously. To achieve the traceability of service data, the block chain technology [21], a product configuration knowledge sharing mechanism [22] and a customized sales strategy [23] are introduced and applied into modelling in different lifecycle stages. In order to accomplish the information feedback function of closed-loop PLM, an ontology-based reasoning platform is put forward to connect the after-sales stage and the design stage [24]. Furthermore, a centrally driven system workflow is established to achieve knowledge sharing among design, manufacturing and supplier departments [25]. In addition, some emerging technical methods and intelligent algorithms are constantly applied to data integration of models. The machine learning methods is adopted to realize the sharing of quality data among the manufacturing, inspection and after-sales stage [26]. Li et al. [27] propose a novel event-adaptive concept integration algorithm which learns the sematic correlation from the concept vocabulary and emphasizes on the most related concepts for the zero-shot event detection. In addition, the categorical theory [28] and DTM (dynamic time warping) method [29] is adopted to connect the product data between design and manufacturing stage, so that designers can obtain the manufacturing data to make the optimal product design decisions. Moreover, the rapid development of the Internet of Things can promote the interconnection of physical information with the support of embedded technology, which raises the concern over application to data integration [30]. The research of these integration methods has greatly promoted the data sharing in different lifecycle stages. Nevertheless, due to the lack of the definition of a universal life cycle model, the scope and integrity of data penetration are limited.

Because of the complexity and heterogeneity of manufacturing service data, not all types of data can be integrated, the research on model transformation is gradually carried out. Model transformation is the process of selecting appropriate model transformation method to obtain the target model according to the source model and the pre-defined mapping rules [31, 32]. Model transformation method includes some mapping rules, and the source model needs to maintain semantic consistency with the target model during the transformation process [33, 34].

The initial research is to transform by identifying the common model characteristics in different stages. Rondini et al. [35] put forward the service reference model and standardize the service model delivery process by identifying key attributes of models. Gradually, semantic identification method is applied to the model transformation. A hierarchical semantic mapping method is proposed to achieve data model transformation while still ensuring the semantic consistency [36]. In order to maintain semantic and topological relationship of the 3D model, a multi-scale semantic information transformation model is presented [37]. With the in-depth study of model transformation, the focus of research gradually falls on new technologies of model transformation and mapping methods. A bilateral model transformation method based on file and API is proposed to realize the transformation and sharing of models among different departments [38], and a generic neural network architecture suitable for heterogeneous model transformations is proposed to learn the manipulation operation [39]. To improve the efficiency of model transformation, a translation schema is developed to translate ATL transformations to Java [40], and Petri net is widely used as a process modelling method to transform process semantic information to model data [41]. Subsequently, more and more attention is paid to the quality of model transformation. A method for using proper patterns called MUPPIT is proposed to detect anti-patterns in the transformation [42], and an approach to symbolic execution of transformation to detect logical errors was also presented [43]. Moreover, Chu et al. [44] put forward a transformation method based on UML profile general extension mechanism, which restricts model transformation by establishing templates in extension files to ensure the high quality of model transformation. In addition, some researchers study from the perspective of model evolution and maintenance. A multi-department virtual alliance is constructed to analyze heterogeneous service models within the alliance [45], and the approach which leverages contract-based model testing techniques is proposed to assist engineers in model transformation evolution and repairing [46]. These studies have explored the transformation methods of models from different perspectives, but the effectiveness is not strong due to the lack of summary and refinement of specific transformation operations.

To summarize, the current manufacturing service modelling, integration and model transformation methods have contributed a lot to the resource virtualization and service encapsulation. However, the existing researches mainly focus on partial lifecycle stages, while studies on modelling and transformation methods considering the global life cycle have been rarely carried out. Following these approaches, this paper studies the manufacturing service modelling and transformation method based on the product lifecycle, comprehensively considers the difference and relationship of service models in different stages, and provides a generic modelling and transformation method for manufacturing services from product lifecycle perspective.

3 Service business model and transformation framework

3.1 Life cycle service business model

In cloud manufacturing environment, the life cycle service model refers to resources and capabilities encapsulated and published from all stages of product lifecycle. The process of manufacturing service model sharing environment is shown in Fig. 1. Manufacturing enterprises describe and model their resources and capabilities in life cycle stages, then publish them to the cloud platform as services, which

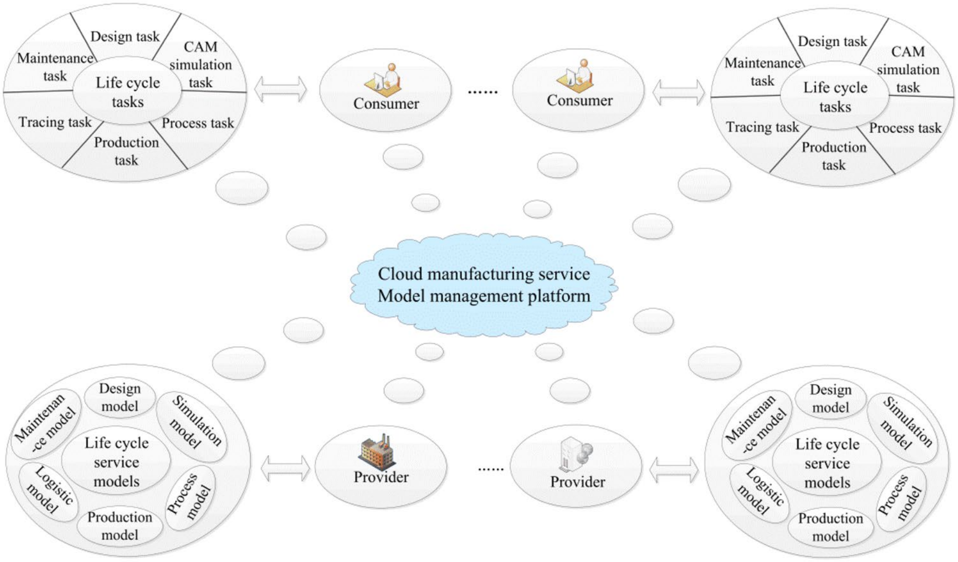


Fig. 1 Manufacturing service model sharing environment

are available for consumers to search and obtain [47]. In this process, accurate modelling and efficient transformation of manufacturing services are the key issues of resource sharing and collaborative manufacturing.

The life cycle service model needs to be built according to stage-specific requirements. In this study, we take the four most representative stages of design, process, manufacturing and maintenance in product lifecycle as an example, and the business model of life cycle manufacturing service is shown in Fig. 2. The manufacturing service data in various life cycle stages are both different and related. Although the input data, participants and output data of each stage are different, the output of previous stage will become the input of later stage, and together constitute the whole life cycle model from design, production to operation and maintenance stage.

On one hand, each stage of product lifecycle has its own manufacturing service data. For instance, in the design stage, there are requirements analysis, structural design, simulation analysis and other design data, while the manufacturing data (such as material blending, machining production and quality inspection, etc.) presents in the manufacturing stage. On the other hand, the service data in different life cycle stages are also interrelated and extended. Considering manufacturing equipment for example, the equipment data is managed as a type resource in the design stage due to the design stage primarily cares about the type of equipment, but in process or manufacturing

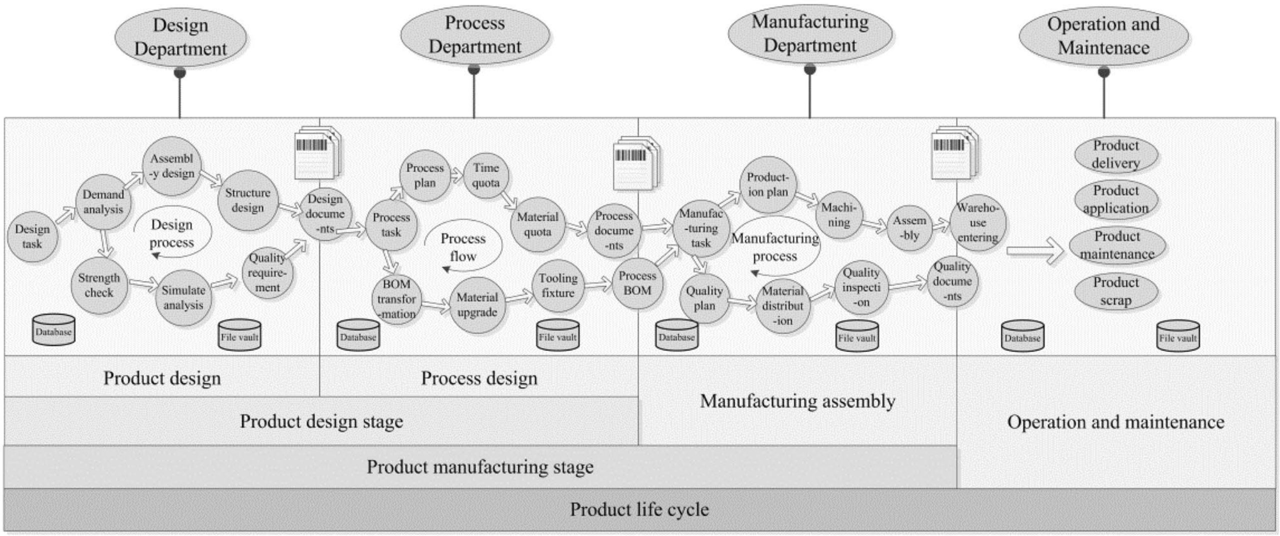


Fig. 2 Business model of product lifecycle service

stage, the equipment data includes both types and individual information because the latter two stages concern about the specific data of each equipment. Therefore, the equipment type data has inheritance relationship among design, process and manufacturing stage, while the individual information is the personalized attribute only belongs to manufacturing stage.

3.2 Transformation framework of service model

Selecting four most representative stages of product lifecycle (i.e. design, process, manufacturing and maintenance) for example, life cycle service model includes four substage models: design service, process service, manufacturing service and maintenance service model. According to the relationship between model services and life cycle stages, the process of transformation includes two aspects: general and view service transformation, as shown in Fig. 3. The former is to update and upgrade general services, and the latter is primarily the addition and deletion operation of view services.

The general service transformation must ensure the data integrity because the general service is the resource that exists in multiple life cycle stages. We assumed that the source stage model LS_s needs to transform to target stage model LS_T. The general service data section of the source view model is CS and the general service data section of the transformed model is CT. The two general service data sets need to meet this mathematical relationship: $CS \subseteq CT$, it means that the general service data in the source stage should be a subset of the data in the target view model. In other words, during the life cycle model transformation, the target general service model will be an enhanced version model of the source general service. For the view service transformation, the characteristics of life cycle stage should be taken into account since view services are the personalized service data presents in certain or several stages. Therefore, it is important to maintain the data consistency and semantic accuracy during the view service transformation.

Considering data management granularity, the life cycle service model data can be divided into three types: class,

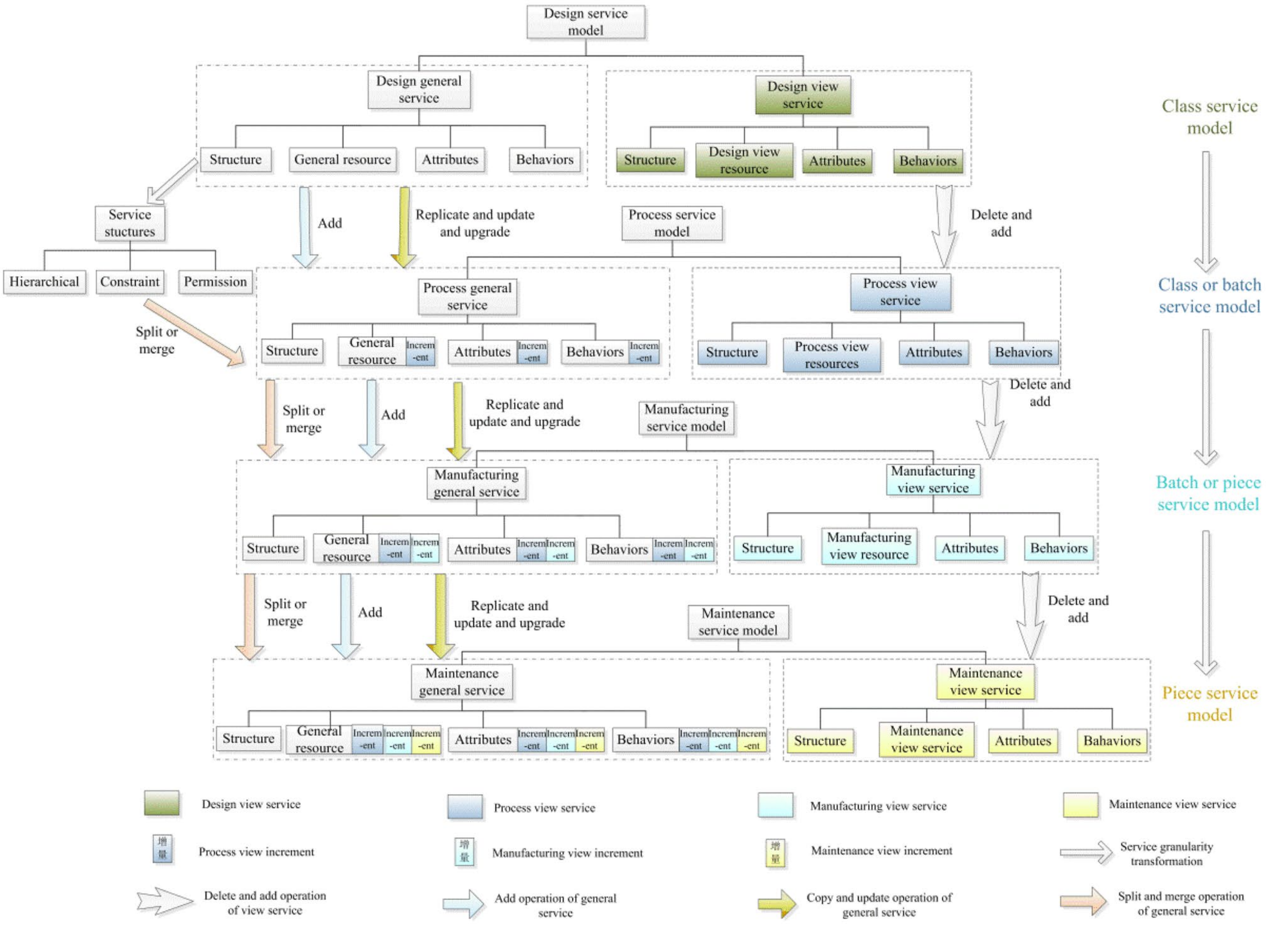


Fig. 3 Schematic diagram of life cycle service model transformation

batch and piece data. During the transformation process, service model data is converted from class type in design stage to batch type in process stage, and then to batch or piece type in manufacturing and maintenance stage. The granularity of model data is managed by class type in design stage, and is organized by batch type in process stage. Then when the model data evolve to production stage, the model data needs to be associated with specific orders, materials and manufacturers, and the data management granularity becomes piece type. On summary, the service management granularity of service data has evolved from class type in design stage to batch data in process stage then to piece type in production stage. The relationship among class type, batch type and piece type can be expressed as: 1 : M : N, where N satisfies $\mathrm{N} = \sum_{i=1}^{m} M_i$, and M_i represents the specific number of each batch.

4 Model definition and transformation operators

4.1 Definitions of life cycle service model

The manufacturing life cycle service model is defined with the introduction of discrete mathematics knowledge in this section. This is a general model that penetrates all lifecycle stages and provides a model basis for subsequent research on transformation operations. Considering the manufacturing service is composed of resources, attributes, behaviors and structures, the modelling of life cycle service is carried out from these four aspects.

Definition 1 (Manufacturing services S) The manufacturing service is the combination of manufacturing resource R, attributes A and behaviors B. Manufacturing resource refers to the individual resource which completes a task independently. The attributes of manufacturing resource can be divided into static attributes and dynamic attributes, represented by A_s and A_d respectively. Resource behaviors include two categories: class behavior and domain behavior. Class behavior is the ability of the resource itself, represented by B_{class}. Domain behavior is the ability of business domain or life cycle stage, represented by B_{Domain}. In general, manufacturing resource is the basis, attributes are basic information description of the resource, and behaviors are capabilities and methods of the resource. Moreover, manufacturing resources include general and view resources. The formalized descriptions of manufacturing service are defined in Eq. (1):

$$S = \langle R, A, B \rangle = \langle r_i, a_i, b_i \rangle$$

$$S = \langle R, A, B \rangle = \left\{ \langle r_{gi} \cup r_{vj}, a_{gi} \cup a_{vj}, b_{gi} \cup b_{vj} \rangle \right\}$$

$$s.t.\ r_i \in R,\ a_i \in \mathrm{A}, b_i \in B,\ \mathrm{i,j} \in \{1, 2, 3 \cdots n\} \tag{1}$$

Definition 2 (Service structures SC) The relationship between services is defined as service structure. According to the type, service structure can be divided into hierarchical structure (SHC), constraint structure (SCC) and permission structure (SRC). Hierarchical structure refers to the parent–child relationship between the manufacturing service and other services, such as assembly structure between materials, assembly number, etc. The hierarchical structure can be expressed as follows:

$$SHC = \langle s_i, s_j, z_{ij} \rangle \tag{2}$$

where s_i is the parent node, s_j is the child node, and z_{ij} is the hierarchical relationship between the two nodes. Furthermore, because of the relationship between manufacturing service and other services, the service will be constrained by other related services. The constrained relationship between services is called constraint structure, such as PBOM is constrained by EBOM, machining equipment is subject to type of process equipment and so on. Constraint structure includes active constraint and passive constraint. This service change causes other services to change, which is called active constraint. On the contrary, the change of this service is caused by other services change, this is called passive constraint. The constraint structure is described as follows:

$$S_s = \left\{ s_i | \exists \langle s_i, s, scc_i \rangle \text{ or } \langle s, s_i, scc_i \rangle \right\} \tag{3}$$

From the data security point of view, the operation of manufacturing services needs permission control. The permission relationship of manufacturing services is called permission structure, which consists of role permission and data permission. Role permission is the access control of different roles to operations, and data permission refers to the access level division of data. The permission structure of service S is represented in Eq. (4).

$$SRC_s = \left\{ src = \langle s_i, s_j, src_{ij} \rangle | s_i = sors_j = s \right\} \tag{4}$$

Definition 3 (Life cycle service model LS) Based on the above definitions, the life cycle service model is defined as the combination of manufacturing service and service structure, as expressed in Eq. (5). Moreover, the life cycle service model includes general and view service model.

$$LS = \langle S_g, S_v, \mathrm{SC}_g, \mathrm{SC}_v \rangle = \left\{ \langle s_{gi}, sc_{gi} \rangle \text{ and } \langle s_{vj}, sc_{vj} \rangle \right\}$$

$$s.t. s_{gi} \in S, s_{vj} \in S, \mathrm{i,j} \in \{1, 2, 3 \cdots n\} \tag{5}$$

Taking the life cycle model of part p for example, firstly establish the service model (including resource, attributes and behaviors) as follows:

$$S_p = \{s = \langle r, a_r, b_r \rangle | r = \mathrm{p}\}$$

$$\begin{aligned} S_p &= \{s = \langle r, a_r, b_r \rangle | r = \mathrm{p}\} \\ &= \{\langle r_g \cup r_v, a_s \cup a_d, b_g \cup b_d \rangle | r = \mathrm{p}\} \end{aligned} \tag{6}$$

Considering the relationship between part p and other materials, the service structure model can be expressed as:

$$SC_p = \{sc = \langle shc_{ij}, scc_{ij}, src_{ij} \rangle\}$$

$$s.t.\ i = porj = p\ and\ \mathrm{i,j} \in \{1, 2, 3 \cdots n\} \tag{7}$$

Finally, the life cycle service model of part p (including service model and structure model) is created as follows:

$$LS_p = \langle S_p, SC_p \rangle \tag{8}$$

In summary, the relationship between the components of life cycle service model is shown in Fig. 4. The manufacturing service model includes resource, attributes and behaviors. The service structure describes the relationship between the service and other related services of product lifecycle. Manufacturing service model and structure model together constitute the life cycle service model.

4.2 Operators for service model transformation

In order to elaborate the process of life cycle service model transformation accurately, eight basic operators are summarized, namely adding, deleting, updating, replication, citing, upgrading, splitting and merging. Furthermore, mathematical symbols are introduced and eight operators are defined mathematically, i.e., addingⒶ, deletionⒹ, updatingⓊ, replicationⓇ, citingⒸ, upgradingⒼ, splittingⓈ, mergingⓂ. These operators have the same priority and are lower than parenthesis.

1. Adding

The adding operator is to increase resource, attributes, behaviors or structures based on the original model. During the transformation process, the target model increases resources or services with their own stage characteristics, then establish structural relationships with the existing services, as shown in Fig. 5a. Assuming that the original service model is L and the new added service is s_l, the adding operation can be defined as:

$$L = \langle S, SC \rangle = \langle \{s_i, s_j\}, \{sc_{ij}\} \rangle$$

$$\begin{aligned} L \textcircled{A} \mathrm{s}_l &= L + \{s_l\} + \mathrm{sc}_{li} + sc_{lj} \\ &= \langle \{s_i, s_l, s_j\}, \{sc_{li}, sc_{lj}, sc_{ij}\} \rangle \end{aligned}$$

$$s.t.\ \mathrm{i,j} \in \{1, 2, 3 \cdots n\}\ , i \neq j, l \in N^+ \tag{9}$$

2. Deletion

In the transformation process of life cycle service model, some resources, attributes, behaviors or structures with stage characteristics should be deleted, so as to reduce data redundancy of service model, as shown in Fig. 5b. We assumed that the original model is L, and the service needs to be removed is s_l, the deletion operation is expressed in Eq. (10).

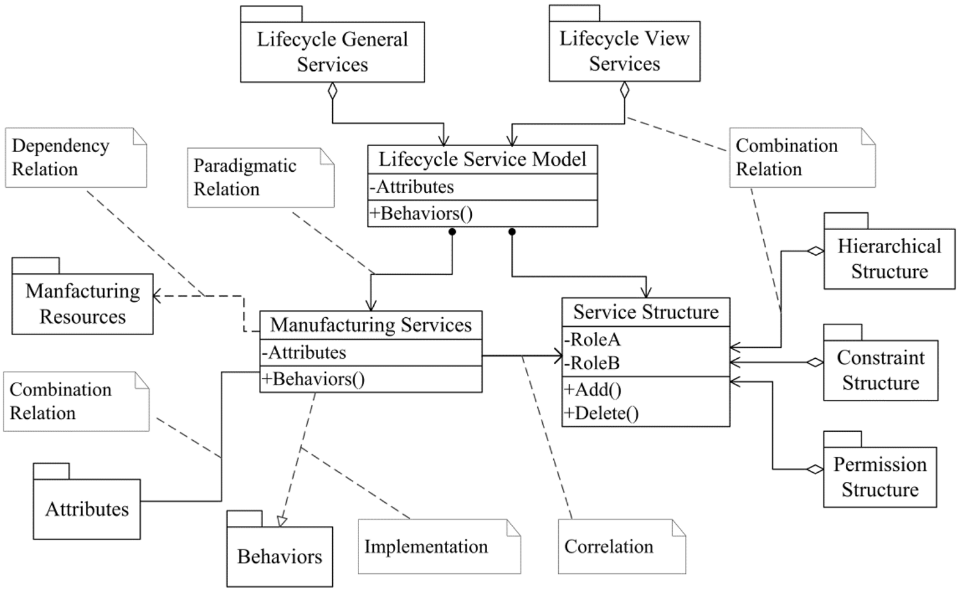


Fig. 4 Relationship of life cycle model components

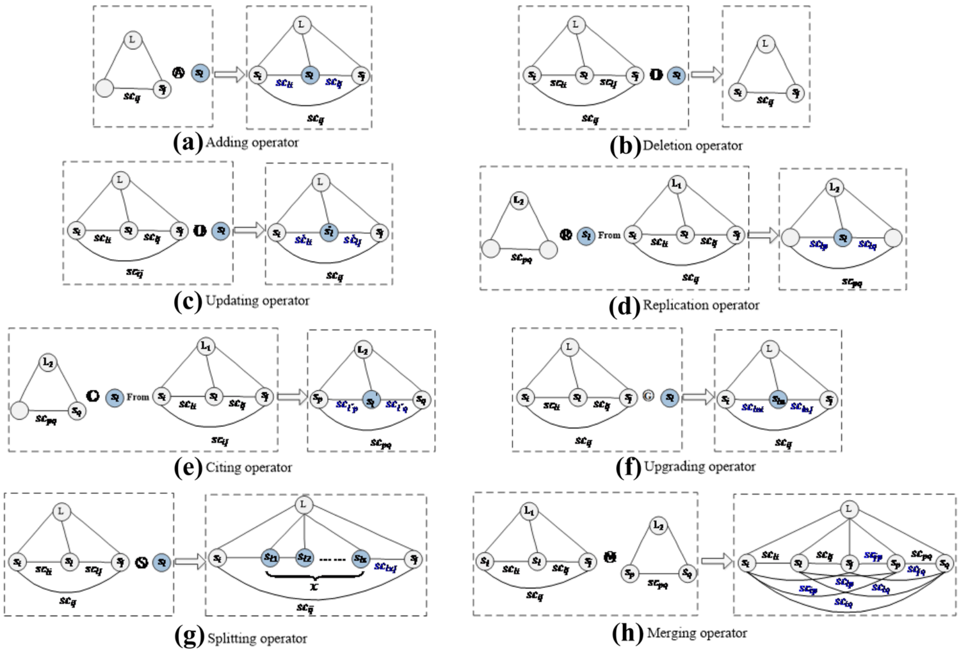


Fig. 5 Basic operations of model transformation

$$L = \langle S, \mathrm{SC} \rangle = \langle \{s_i, s_l, s_j\}, \{sc_{li}, sc_{lj}, sc_{ij}\} \rangle$$

$$L\textcircled{D}\mathrm{s}_l = L - \{s_l\} - sc_{li} - sc_{lj} = \langle \{s_i, s_j\}, \{sc_{ij}\} \rangle \quad s.t.\ \mathrm{i,j} \in \{1, 2, 3 \cdots n\}\ and\ i \neq j \tag{10}$$

3. Updating

During the model transformation process, the value of service model often needs to be updated to meet the requirements of the target stage. The updating operator will not only modify the service itself, but also affect the structure relationship, as shown in Fig. 5c. The original model is assumed as L, and the service needs to be updated is supposed to be $\mathrm{s_l}$. The updating operator is described as follows:

$$L = \langle S, \mathrm{SC} \rangle = \langle \{s_i, s_l, s_j\}, \{sc_{li}, sc_{lj}, sc_{ij}\} \rangle$$

$$\begin{aligned} L\textcircled{A}\mathrm{s}_l &= L - \{s_l\} - sc_{li} - sc_{lj} + \{\dot{s}_l\} + s\dot{c}_{li} + s\dot{c}_{lj} \\ &= \langle \{s_i, \dot{s}_l, s_j\}, \{sc_{li}, s\dot{c}_{lj}, s\dot{c}_{ij}\} \rangle \\ &s.t.\ \mathrm{i,j} \in \{1, 2, 3 \cdots n\}\ and\ i \neq j \end{aligned} \tag{11}$$

where $\dot{\mathrm{s}}_\mathrm{l}$ and $\mathrm{s\dot{c}_{lj}}$ represent the updated services and the updated structural relationships respectively.

4. Replication

Due to some services exist in multiple life cycle stages, the target service model can obtain most of the data by replicating from the original model. Replication is one of the most efficient transformation operators. As shown in Fig. 5d, the transformation data of replication operator includes services and structures. Assuming two life cycle service models are $\mathrm{L_1}$ and $\mathrm{L_2}$, and the service $\mathrm{s_l}$ in $\mathrm{L_1}$ need to be replicated to $\mathrm{L_2}$. The replication operator is represented as:

$$\mathrm{L}_1 = \langle S, SC \rangle = \langle \{s_i, s_l, s_j\}, \{sc_{li}, sc_{lj}, sc_{ij}\} \rangle \quad s.t.\ i,j \in \{1, 2, 3 \cdots n\}\ and\ i \neq j$$

$$\mathrm{L}_2 = \langle S, SC \rangle = \langle \{s_p, s_q\}, \{sc_{qp}\} \rangle \quad s.t.\ p, q \in \{1, 2, 3 \cdots m\}\ and\ p \neq q$$

$$\begin{aligned} \mathrm{L}_2\textcircled{R}\mathrm{s}_l\ \mathrm{from}\ \mathrm{L}_1 &= L_2 + \{s_l\} + sc_{lp} + sc_{lq} \\ &= \langle \{s_p, s_l, s_q\}, \{sc_{pq}, sc_{lp}, sc_{lq}\} \rangle \end{aligned} \tag{12}$$

5. Citing

Citing operator is used for the situation when the service model at one stage only need query the service model at another stage without modification. There have great differences between citing operator and copying operator. The copying operation is to create a new service object, while the citing operation is only to establish a reference relationship with the original object. Most importantly, the reference service model can only be queried but not modified. As shown in Fig. 5e, citing operation needs to create relationship between reference service and original service, and establish structural relationship. It is assumed that two life cycle service models are $\mathrm{L_1}$ and $\mathrm{L_2}$, model $\mathrm{L_2}$ needs refer to the service $\mathrm{s_l}$ in model $\mathrm{L_1}$. The formalized descriptions of citing operator can be described as follows:

$$\mathrm{L}_1 = \langle S, \mathrm{SC}\rangle = \langle \left\{s_i, s_l, s_j\right\}, \left\{sc_{li}, sc_{lj}, sc_{ij}\right\}\rangle \\ s.t.\ i,j \in \{1,2,3\cdots n\}\ and\ i \neq j$$

$$L_2 = \langle S, \mathrm{SC}\rangle = \langle \left\{s_p, s_q\right\}, \left\{sc_{qp}\right\}\rangle \\ s.t.\ p,q \in \{1,2,3\cdots m\}\ and\ p \neq q$$

$$\begin{aligned}\mathrm{L}_2 \text{Ⓒ} \mathrm{s}_l \text{ from } \mathrm{L}_1 &= L_2 + \left\{s_l^{'}\right\} + sc_{l'p} + sc_{l'q} \\ &= \langle \left\{s_p, s_l^{'}, s_q\right\}, \left\{sc_{pq}, sc_{l'p}, sc_{l'q}\right\}\rangle\end{aligned} \tag{13}$$

where $s_l^{'}$ is the citing model point to s_l, and $sc_{l'p}$, $sc_{l'q}$ represent the structural relationship between $s_l^{'}$ and the original services (s_p, s_q) in L_2.

6. Upgrading

When service model changes during transformation, the upgrading operator will be used to ensure that the change process can be fully recorded and traced. Upgrading operations create a new service model based on the original service and increase the version number of the model, as shown in Fig. 5f. The service model of old version is converted to background only for query. Suppose that the service needs to be upgraded in life cycle model L is s_l, the upgrading operator is described in Eq. (14).

$$L = \langle S, SC\rangle = \langle \left\{s_i, s_l, s_j\right\}, \left\{sc_{li}, sc_{lj}, sc_{ij}\right\}\rangle \\ s.t.\ i,j \in \{1,2,3\cdots n\}\ and\ i \neq j$$

$$\begin{aligned}L \text{Ⓖ} \mathrm{s}_l &= L - \left\{s_l\right\} - sc_{li} - sc_{lj} + \left\{s_{ln}\right\} + sc_{l\mathrm{n}i} + sc_{lnj} \\ &= \langle \left\{s_i, s_{ln}, s_j\right\}, \left\{sc_{l\mathrm{n}i}, sc_{lnj}, sc_{ij}\right\}\rangle\end{aligned} \tag{14}$$

In the formula, s_{ln} refers to the upgraded model based on s_l. The upgrading operation satisfies the expression as:

$$s_{ln} = \mathrm{s}_\mathrm{l} + \Delta \mathrm{s}_\mathrm{l} + UpdateVersionNo \tag{15}$$

where Δs_l represents the increment of transformation, and *UpdateVersionNo* describes the upgraded version number.

7. Splitting

Due to different granularity of service model management in various life cycle stages, splitting operator will be used in the conversion process when class data is transferred to batch or piece data. The process of splitting operation includes model service splitting and structural relationship splitting, as shown in Fig. 5g. Assuming the number of service s_l contained in the life cycle service model L is $x(x > 1)$, the splitting operation for implementing piece traceability is described as follows:

$$L = \langle S, SC\rangle = \langle \left\{s_i, s_l, s_j\right\}, \left\{sc_{li}, sc_{lj}, sc_{ij}\right\}\rangle \\ s.t.\ i,j \in \{1,2,3\cdots n\}\ and\ i \neq j$$

$$L \text{Ⓢ} \mathrm{s}_l = L +^{S_l} /_x$$

$$\langle \left\{s_i, \left(s_{l1}, s_{l2} \cdots s_{lx}\right), s_j\right\}, \left\{\begin{array}{c}\left(sc_{l1i}, sc_{l2i} \cdots sc_{lxi}\right), \\ \left(sc_{l1j}, sc_{l2j} \cdots sc_{lxj}\right), \\ sc_{ij}\end{array}\right\}\rangle \tag{16}$$

where $s_{l1}, s_{l2} \cdots s_{lx}$ represent new service models after splitting. Similarly, the corresponding structural relationships are split following the same rules.

8. Merging

Contrary to the splitting operator, the merge operator usually happens in the transformation from batch or piece data to class data. The main purpose of merging operator is to reduce data redundancy and improve data management efficiency. Merging operator involves service model merging and structural relationship merging, as shown in Fig. 5h. Suppose two life cycle service models, L_1 and L_2, need to be merged, and the merging operation is described in Eq. (17).

$$\mathrm{L}_1 = \langle S_{\mathrm{L}1}, SC_{L1}\rangle = \langle \left\{s_i, s_l, s_j\right\}, \left\{sc_{li}, sc_{lj}, sc_{ij}\right\}\rangle$$

$$s.t.\ i,j \in \{1,2,3\cdots n\}\ and\ i \neq j$$

$$\mathrm{L}_2 = \langle S_{L2}, SC_{L2}\rangle = \langle \left\{s_p, s_q\right\}, \left\{sc_{qp}\right\}\rangle$$

$$s.t.\ p,q \in \{1,2,3\cdots m\}\ and\ p \neq q$$

$$L_1 \text{Ⓜ} \mathrm{L}_2 =$$

$$\begin{aligned}&S_{\mathrm{L}1} + S_{L2} + \left\{sc_{pi}, sc_{pl}, sc_{pj}\right\} + \left\{sc_{qi}, sc_{ql}, sc_{qj}\right\} \\ &= \langle \begin{array}{c}\left\{s_i, s_l, s_j, s_p, s_q\right\}, \\ \left\{sc_{li}, sc_{lj}, sc_{ij}, sc_{pq}, sc_{pi}, sc_{pl}, sc_{pj}, sc_{qi}, sc_{ql}, sc_{qj}\right\}\end{array}\rangle\end{aligned} \tag{17}$$

4.3 Transformation process and change propagation

The transformation process of life cycle service models in different stages should follow the principle from the whole to the part. Depending on the analysis of service transformation, the overall flow chart and rule of service model are summarized into three steps (as shown in Fig. 6). Firstly the general and view model of life cycle services are distinguished and transformed, then the service structure (including hierarchical structure, constraint structure and permission structure) is converted, and finally the service model (including resource, attribute and behavior) is transformed.

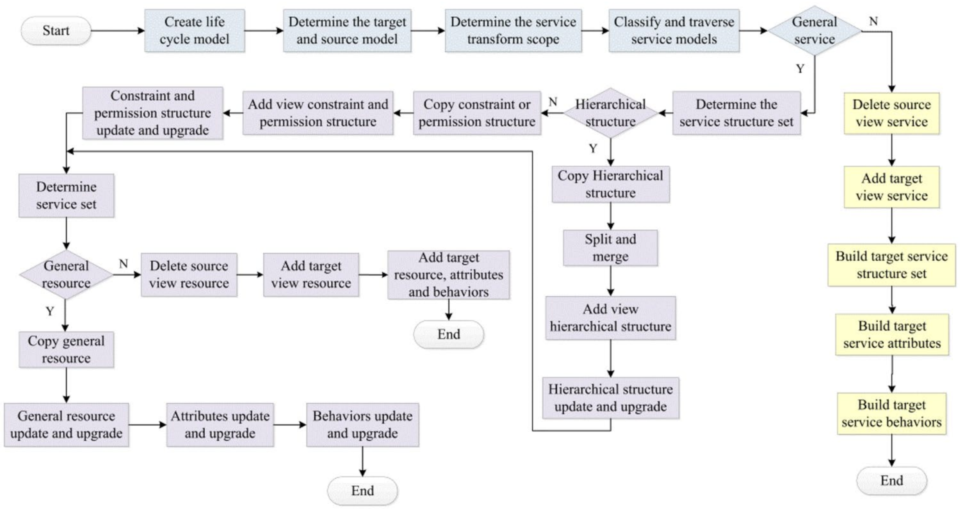

Start
Create life cycle model
Determine the target and source model
Determine the service transform scope
Classify and traverse service models
General service
N
Y
Delete source view service
Add target view service
Build target service structure set
Build target service attributes
Build target service behaviors
End
Determine the service structure set
Hierarchical structure
N
Y
Copy constraint or permission structure
Add view constraint and permission structure
Constraint and permission structure update and upgrade
Copy Hierarchical structure
Split and merge
Add view hierarchical structure
Hierarchical structure update and upgrade
Determine service set
General resource
N
Y
Delete source view resource
Add target view resource
Add target resource, attributes and behaviors
End
Copy general resource
General resource update and upgrade
Attributes update and upgrade
Behaviors update and upgrade
End

Fig. 6 The transformation logic process of life cycle service model

As shown in Fig. 6, the common transformation rules of life cycle service model are divided into three parts, which are displayed with different color blocks. The blue blocks are the beginning part of transformation, which mainly contains life cycle model creation, the determination of transformation scope and direction and the distinguishing of general and view services. The purple blocks are the general service transformation part, which are divided into two aspects: service structure conversion and service model conversion. The service structure conversion is responsible for the transformation of hierarchy structure, constraint structure and permission structure, and the service model conversion is mainly in charge of transformation resources, attributes and behaviors. The yellow blocks are the transformation part of view service models, which primarily consist of deleting the original view service model and adding the target view service model.

In addition, due to the dynamic and complexity characteristics of the production process, the data of service model will inevitably change. In order to describe the transformation process of service model accurately, it is necessary to study and elaborate the change propagation mechanism of service model.

When the data of service model changes, we firstly need to find the change closure, which is a collection of all service models affected by the changed model. The determination of the change closure is mainly depending on service structure relationship, i.e. hierarchical, constraint and permission structure. The transformation process of service models in change closure follow the conversion method proposed in this paper. New life cycle service models will generated after the change propagation transformation. The change propagation mechanism is described in Fig. 7.

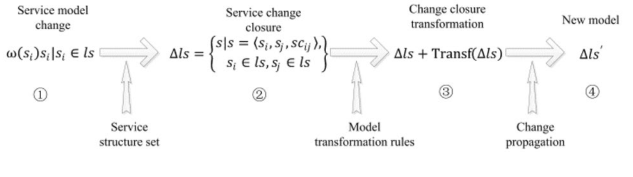

Service model change
$\omega(s_i)s_i|s_i \in ls$
①
Service structure set
Service change closure
$\Delta ls = \left\{\begin{matrix} s|s=\langle s_i, s_j, sc_{ij}\rangle, \\ s_i \in ls, s_j \in ls \end{matrix}\right\}$
②
Model transformation rules
Change closure transformation
$\Delta ls + \mathrm{Transf}(\Delta ls)$
③
Change propagation
New model
$\Delta ls'$
④

Fig. 7 Change propagation mechanism of service model

In Fig. 7, s_i refers to the changed service in the life cycle service model ls, $\omega\left(s_i\right)$ represents the change function of service model, Δls describes the service change closure based on the analysis of service structure, $Transf(\Delta ls)$ defines the transformation function of service model, and ls' is the new life cycle service models after change propagation transformation.

The change propagation mainly consists of service model transformation and service structure transformation. Taking the change propagation of service model from process stage to manufacturing stage as an example, the change propagation rules are described in Table 1 by using ATL language.

Table 1 Change propagation rules of service model

Module Change $\Delta \mathrm{LS}_\mathrm{p}$ 2 $\Delta \mathrm{LS}_\mathrm{m}$
Create out : $\Delta \mathrm{LS}_\mathrm{m}$ from IN: $\Delta \mathrm{LS}_\mathrm{p}$
Uses strings;
Rule Change Propagation
{
From p: $\Delta \mathrm{LS}_\mathrm{p}$ to m: $\Delta \mathrm{LS}_\mathrm{m}$ (
$\Delta S_m < \text{Ⓖ}\left(\text{Ⓤ}\left(\text{Ⓡ}\Delta S_{pc}\right)\right) - \left(\text{Ⓓ}\Delta S_{pv}\right) + \left(\text{Ⓐ}\Delta S_{mv}\right)$
$\Delta SC_m < \text{Ⓖ}\left(\text{Ⓤ}\left(\text{Ⓡ}\Delta SC_p\right)\right) + \text{Ⓖ}\left(\text{Ⓢ}\left(\text{Ⓡ}\Delta SC_p\right)\right) +$
$\text{Ⓖ}\left(\text{Ⓜ}\left(\text{Ⓡ}\Delta SC_p\right)\right) + \left(\text{Ⓐ}\Delta SC_m\right)$
)
m: LS_m (
$\mathrm{LS}' < \mathrm{LS}_\mathrm{m} + \Delta \mathrm{LS}_\mathrm{m} = \mathrm{LS}_\mathrm{m} + \Delta \mathrm{S}_\mathrm{m} + \Delta \mathrm{SC}_\mathrm{m}$
)
}

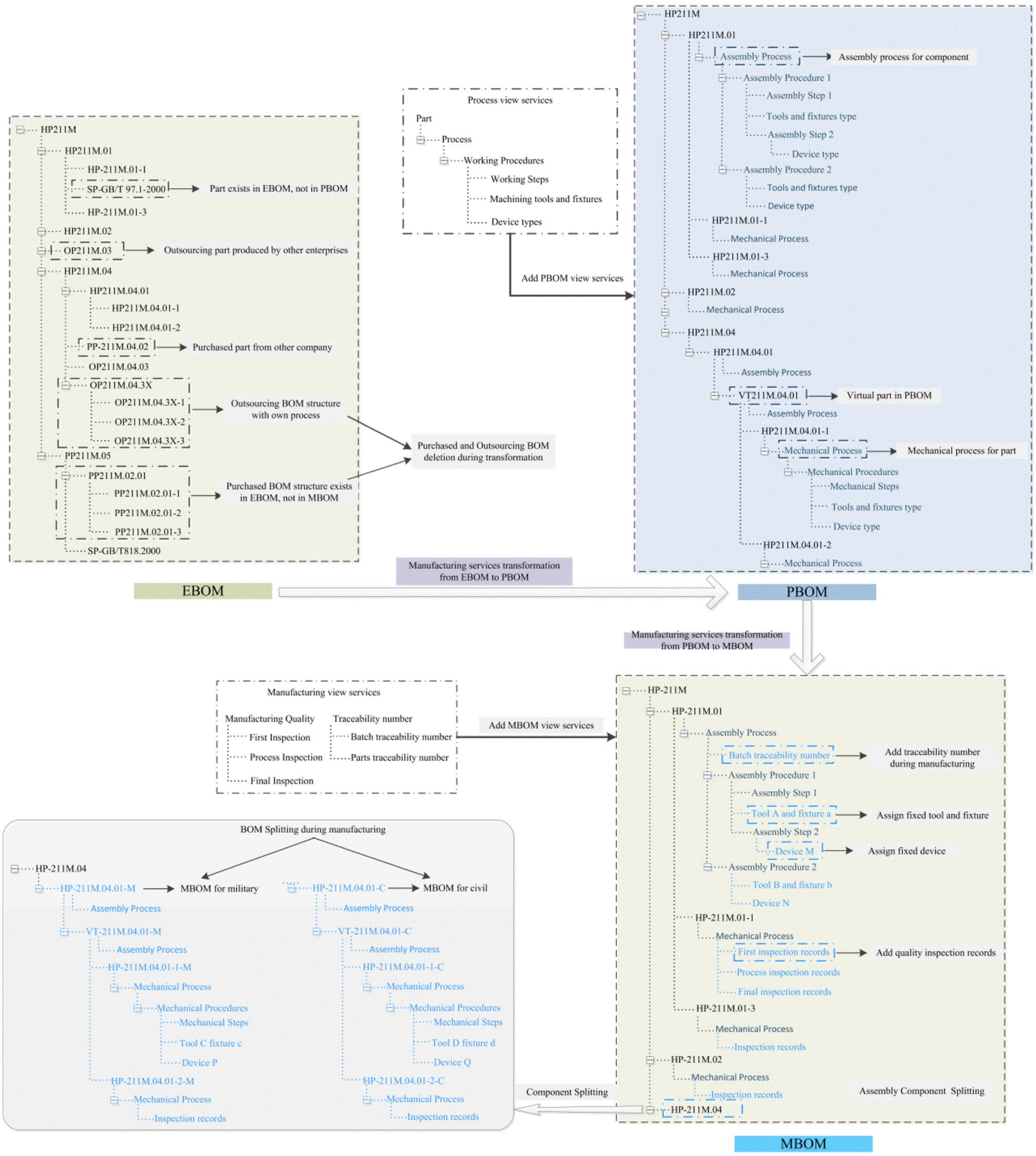


Fig. 8 BOM service model transformation process

5 Case study

The service model transformation methodology proposed in this study has been successfully applied in an instrument enterprise, and a software system is also developed to realize the data conversion.

5.1 A product BOM service model transformation

BOM is the important data of product life cycle service model, so the transformation process of the product BOM data is very representative. In order to explain the process of service model transformation clearly, this paper selects

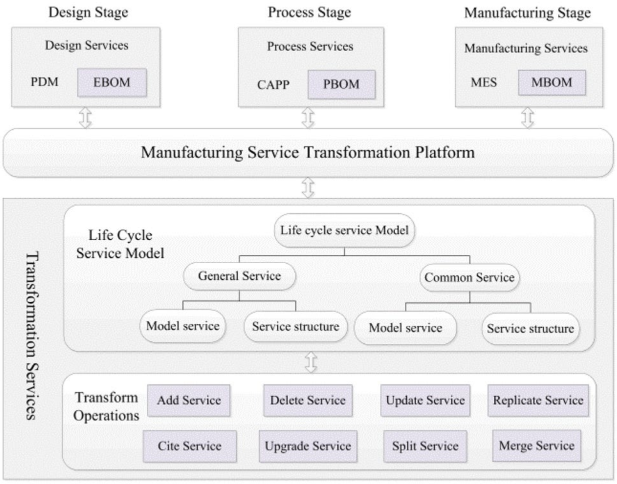


Fig. 9 Transform platform framework

the BOM model transformation process of an instrument product as a verification example. Considering the confidentiality of enterprise data, materials and structure information of the BOM are fictionalized. The involved materials can be classified into four types: standard part (SP), homemade part (HP), outsourcing part (OP) and purchased part (PP).

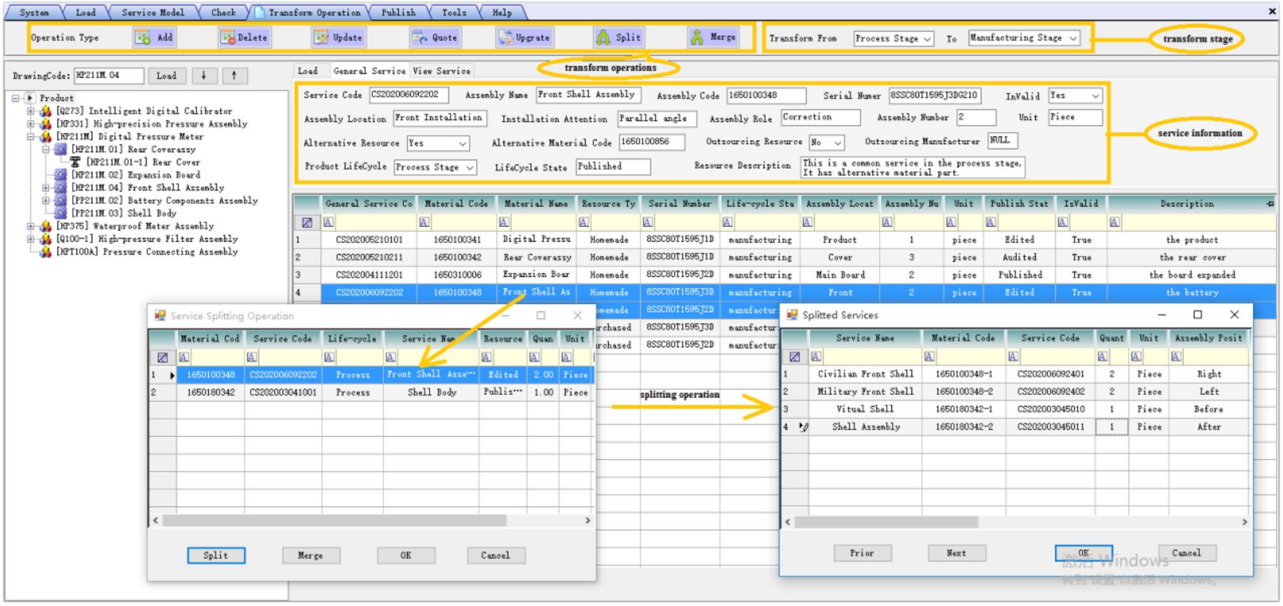
According to the transformation method proposed in this study, the life cycle transformation process of the BOM service model is shown in Fig. 8. The shadow parts in the figure which is circled by dash line represent the general service model transformation, the operations of which include adding, updating, citing, splitting, merging and upgrading. Outside the shadow, surrounded by dash dot line is the view service model transformation, the main operations of which mainly include deleting and adding. The detailed transformation flow and operation information are identified and explained in the diagram. The service model management and integration throughout the life cycle are clearly described by the transformation process.

5.2 System implementation

In order to achieve the integration and transformation of service model in different life cycle stages of the instrument enterprise, a transformation platform (TPlatform) architecture is designed, as shown in Fig. 9. The system includes eight transformation operator services (adding, deleting, updating, replication, citing, upgrading, splitting and merging), which are used to ensure the efficiency and accuracy of the conversion. The TPlatform provides a platform system for multiple life cycle BOM models transformation, which includes loading, service model, check, transform operation, publish and other functions. Because the material in manufacturing stage needs to be distinguished between military and civilian, two materials in the process stage are split into four materials in manufacturing stage by invoking the splitting operator (as shown in Fig. 10).

5.3 Results discussion

Overall, we can see from the case study that the BOM service model can be transformed from one life cycle stage to another by using the method proposed in this



Transform From: Process Stage To: Manufacturing Stage

DrawingCode: HP211M.04

- Product
  - [Q273] Intelligent Digital Calibrator
  - [HP331] High-precision Pressure Assembly
  - [HP211M] Digital Pressure Meter
    - [HP211M.01] Rear Coverassy
      - [HP211M.01-1] Rear Cover
    - [HP211M.02] Expansion Board
    - [HP211M.04] Front Shell Assembly
    - [PP211M.02] Battery Components Assembly
    - [PP211M.03] Shell Body
  - [HP375] Waterproof Meter Assembly
  - [Q100-1] High-pressure Filter Assembly
  - [HPT100A] Pressure Connecting Assembly

Service Code: CS202006092202 Assembly Name: Front Shell Assembly Assembly Code: 1650100348 Serial Numer: 8SSC80T1595J3DG210 InValid: Yes

Assembly Location: Front Installation Installation Attention: Parallel angle Assembly Role: Correction Assembly Number: 2 Unit: Piece

Alternative Resource: Yes Alternative Material Code: 1650100856 Outsourcing Resource: No Outsourcing Manufacturer: NULL

Product LifeCycle: Process Stage LifeCycle State: Published Resource Description: This is a common service in the process stage, It has alternative material part.

	General Service Co	Material Code	Material Name	Resource Ty	Serial Number	Life-cycle Sta	Assembly Locat	Assembly Nu	Unit	Publish Stat	IsValid	Description
1	CS202005210101	1650100341	Digital Pressu	Homemade	8SSC80T1595J1D	manufacturing	Product	1	piece	Edited	True	the product
2	CS202005210211	1650100342	Rear Coverassy	Homemade	8SSC80T1595J1D	manufacturing	Cover	3	piece	Audited	True	the rear cover
3	CS202004111201	1650310006	Expansion Boar	Homemade	8SSC80T1595J2D	manufacturing	Main Board	2	piece	Published	True	the board expanded
4	CS202006092202	1650100348	Front Shell As	Homemade	8SSC80T1595J3D	manufacturing	Front	2	piece	Edited	True	the battery
				menade	8SSC80T1595J2D	manufactur						
				rchased	8SSC80T1595J3D	manufactur						
				rchased	8SSC80T1595J2D	manufactur						

Service Splitting Operation

	Material Cod	Service Code	Life-cycle	Service Nam	Resource	Quan	Unit
1	1650100348	CS202006092202	Process	Front Shell Asse···	Edited	2.00	Piece
2	1650180342	CS202003041001	Process	Shell Body	Publis···	1.00	Piece

Splitted Services

	Service Name	Material Code	Service Code	Quant	Unit	Assembly Posit
1	Civilian Front Shell	1650100348-1	CS202006092401	2	Piece	Right
2	Military Front Shell	1650100348-2	CS202006092402	2	Piece	Left
3	Vitual Shell	1650180342-1	CS202003045010	1	Piece	Before
4	Shell Assembly	1650180342-2	CS202003045011	1	Piece	After

Fig. 10 Splitting operations in TPlatform

paper. In the transformation instance, the conversation process includes general and view model transformation. The operation objects include material, attribute, relationship and relationship attribute. The complexity of attribute and relationship are significantly higher than that of attributes. Due to the complexity of BOM data, the transformation process contains almost all types of operators proposed in this paper. From the perspective of running time, updating is the shortest because it only needs attribute operation, followed by the operators of adding, deleting and citing which related to material or relationship operation, and the most complicated operators are replication, upgrading, splitting and merging which involves both material and attribute operation. Since the transformation of a service model often related to multiple objects, attributes and operators, the running time is proportional to the number of objects, attributes and operators involved.

6 Conclusion and future work

This paper has presented a novel transformation method to address the problem of data interpenetration and efficient model conversion in different stages of product lifecycle. Instead of using data integration or enhancing a stage model, we deeply analyze the data evolution relationship of product lifecycle and build a general model to achieve data interpenetration in different lifecycle stages from the model level. Since the service model in different lifecycle stages has characteristics of modularization and componentization, we propose a novel model transformation method which contains general and view service transformation. Moreover, by summarizing and mathematically defining eight transformation operators, the proposed method can provide more explicit instructions for the model conversion. Furthermore, in order to solve the propagation problem of models in different stages when data changes, we study and establish the change propagation mechanism to ensure the timely data synchronization. The successful application in the transformation of BOM service models in an instrument enterprise verifies the rationality and effectiveness of the proposed methodology.

In the future, we plan to introduce clustering to classify resource before modelling and explore machine learning algorithms to improve the transformation method.

Funding This research was supported by the National Science and Technology Major Project of China under Grant No. 2018ZX04001006.

Declarations

Conflicts of interest The authors declare that they have no conflict of interest.

References

1. Zhang L, Luo YL, Tao F, Li BH, Ren L, Zhang XS, Hua G, Cheng Y, Hu AR, Liu YK (2014) Cloud manufacturing: a new manufacturing paradigm. Enter Inf Syst-UK 8(2):167–186
2. Borangiu T, Trentesaux D, Thomas A, Leitão P, Barata J (2019) Digital transformation of manufacturing through cloud services and resource virtualization. Comput Ind 108:150–162
3. Li JR, Tao F, Cheng Y, Zhao LJ (2015) Big Data in product lifecycle management. Int J Adv Manuf Technol 81:667–684
4. Vezzetti E, Violante MG, Marcolin F (2014) A benchmarking framework for product lifecycle management (PLM) maturity models. Int J Adv Manuf Technol 71:899–918
5. Hedberg T, Feeney AB, Helu M, Camelio JA (2017) Toward a lifecycle information framework and technology in manufacturing. J Comput Inf Sci Eng 17:021010
6. Shin JH, Kiritsis D, Xirouchakis P (2015) Design modification supporting method based on product usage data in closed-loop PLM. Int J Comput Integ M 28(6):551–568
7. Moghaddam M, Nof SY (2018) Collaborative service-component integration in cloud manufacturing. Int J Prod Res 56(1–2):677–691
8. Luo Y, Zhang L, Tao F et al (2013) A modelling and description method of multidimensional information for manufacturing capability in cloud manufacturing system. Int J Adv Manuf Technol 69(5–8):961–975
9. Liu N, Li X (2012) A resource virtualization mechanism for cloud manufacturing systems. International IFIP Working Conference on Enterprise Interoperability, Springer, Berlin Heidelberg 122:46–59
10. Liu N, Li XP, Shen WM (2014) Multi-granularity resource virtualization and sharing strategies in cloud manufacturing. J Netw Comput Appl 46:72–82
11. Xu W, Yu J, Zhou Z et al (2015) Dynamic modelling of manufacturing equipment capability using condition information in cloud manufacturing. J Manuf Sci E-T ASME 137(4):040907
12. Li ZH, Nie FP, Chang XJ et al (2018) Rank-Constrained spectral clustering with flexible embedding. IEEE Transactions on neural networks and learning systems 29(12):6073–6082
13. Li ZH, Nie FP, Chang XJ et al (2018) Dynamic affinity graph construction for spectral clustering using multiple features. IEEE Transactions on neural networks and learning systems 29(12):6323–6332
14. Talhi A, Fortineau V, Huet JC et al (2019) Ontology for cloud manufacturing based product lifecycle management. J Intell Manuf 30:2171–2192
15. Zhao YY, Liu Q, Xu WJ, Wu XX et al (2017) Dynamic and unified modelling of sustainable manufacturing capability for industrial robots in cloud manufacturing. Int J Adv Manuf Technol 93(5–8):2753–2771
16. Guo H, Shu M (2018) Research on the cloud service description model for cloud service composition in cloud manufacturing system. J Phys Conf Ser 052015:1–7
17. Yu CY, Mou SD, Ji YJ et al (2018) a delayed product differentiation model for cloud manufacturing. Comput Ind Eng 117:60–70
18. Duan YY, Han K (2017) Self-organization evolution model of cloud manufacturing service collaborative network. AMME 2017:210–213

19. Namchul D (2017) Integration of design and manufacturing data to support personal manufacturing based on 3D printing services. Int J Adv Manuf Technol 90:3761–3773
20. Camba JD, Contero M, Company P, Pérez D (2017) On the integration of model-based feature information in product lifecycle management systems. Int J Inform Manage 37(6):611–621
21. Liu XL, Wang WM, Guo HY, Barenji AV, Li Z, Huang GQ (2019) Industrial blockchain based framework for product lifecycle management in industry 4.0. Robot Comput Integr Manuf 63(1):1–16
22. Myrodia A, Randrup T, Hvam L (2019) Configuration lifecycle management maturity model. Comput Ind 106:30–47
23. Lam HY, Tsang YP, Wu CH, Tang Valerie (2020) Data analytics and the P2P cloud: an integrated model for strategy formulation based on customer behavior. Peer Peer Netw Appl (4)
24. Ali MM , Doumbouya MB , Louge T et al (2020) Ontology-based approach to extract product's design features from online customers' reviews. Comput Ind 116:103175
25. Souri ME, Gao J, Simmonds C (2019) Integrating manufacturing knowledge with design process to improve quality in the aerospace industry. Procedia CIRP 84:374–379
26. Ko T, Lee JH, Cho H, Cho S, Lee W, Lee M (2017) Machine learning-based anomaly detection via integration of manufacturing, inspection and after-sales service data. Ind Manag Data Syst 117(5):927–945
27. Li ZH, Yao LN, Chang XJ et al (2019) Zero-shot event detection via event-adaptive concept relevance mining. Pattern Recogn 88:595–603
28. Wisnesky R, Breiner S, Jones A, et al (2016) Using category theory to facilitate multiple manufacturing service database integration. J Comput Inf Sci Eng 16(2):021011
29. Helu M, Joseph A, Hedberg T (2018) A standards-based approach for linking as-planned to as-fabricated product data. CIRP Ann - Manuf Techn 67:487–490
30. Mekki K, Derigent W, Rondeau E et al (2017) Data lifecycle management in smart building using wireless sensors networks. IFAC-PapersOnLine 50–1:12944–12949
31. Li DC, Wen IH, Chen WC (2016) A novel data transformation model for small data-set learning. Int J Prod Res 54(24):7453–7463
32. Calvar TL, Chhel F, Jouault F et al (2021) Coupling solvers with model transformations to generate explorable model sets. Softw Syst Model 20:1633–1652
33. Xiang F, Huang YY, Zhang Z, Jiang GZ, Zuo Y (2019) Research on ECBOM modelling and energy consumption evaluation based on BOM multi-view transformation. J Amb Intel Hum Comp 10:953–967
34. Rutle A, Lovino L, König H et al (2020) A query-retyping approach to model transformation co-evolution. Softw Syst Model 19:1107–1138
35. Rondini A, Pezzotta G, Cavalieri S et al (2018) Standardizing delivery processes to support service transformation: A case of a multinational manufacturing firm. Comput Ind 100:115–128
36. Costa G, Sicilia A (2020) Alternatives for facilitating automatic transformation of BIM data using sematic query languages. Automat Constr 120:103384
37. Zhu LL (2019) Research on building semantic model transformation based on integration of BIM and GIS. Central China Normal University 5:35–39 (in Chinese)
38. Platenius-Mohr M, Malakuti S, Grüner S, Schmitt J, Goldschmidt T (2020) File- and API-based interoperability of digital twins by model transformation: An IIoT case study using asset administration shell. Future Gener Comp Sy 113:94–105
39. Burgueño L, Cabot J, Li S et al (2021) A generic LSTM neural network architecture to infer heterogeneous model transformations. Softw Syst Model: https://doi.org/10.1007/s10270-021-00893-y
40. Höppner S, Kehrer T, Tichy M (2021) Constrasting dedicated model transformation languages versus general purpose languages: a historical perspective on ATL versus Java based on complexity and size. Softw Syst Model: https://doi.org/10.1007/s10270-021-00937-3
41. Jesús B, Lucía DV, Pedro A, Ana N (2019) A safety analysis of roundabouts and turbo roundabouts based on Petri nets. Traffic Inj Prev 20(4):400–405
42. Panahandeh M, Hamdaqa M, Zamani B et al (2021) MUPPIT: a method for using proper patterns in model transformations. Softw Syst Model 20:1491–1523
43. Azizi B, Zamani B, Kolahdouz-Rahimi S (2020) SEET: Symbolic execution of ETL transformations. J Syst Softw 168:110675
44. Chu CY, Ren XS, Sun AC (2018) Platform independent model transformation based on sysml. Comput Appl Softw 35(12):7–11 (in Chinese)
45. Qi LQ, Wang CC, Wang CD (2020) Key Factors of the transformation of port's equipment manufacturing and producer services based on virtual alliance. J Coastal Res103:654–657
46. Rodriguez-Echeverria R, Macias F, Rutle A et al (2021) Suggesting model transformation repairs for rule-based languages using a contract-based testing approach. Softw Syst Model: https://doi.org/10.1007/s10270-021-00891-0
47. Xu B, Qi J, Hu XX, Leung KS et al (2018) Self-adaptive bat algorithm for large scale cloud manufacturing service composition. Peer Peer Netw Appl 11:1115–1128

Tao Ding has received his Ph.D. degree in School of Mechanical Engineering and Automation at Beihang University (BUAA), Beijing in the year 2019. He is currently working as postdoctoral researcher in Mechanical Engineering, Beihang University. His research interests include collaborative manufacturing, blockchain, cloud manufacturing and distributed systems.

Guangrong Yan is currently an associate professor in School of Mechanical Engineering and Automation at Beihang University (BUAA). He is a member of the National Professional Standardization Technical Committee in China. He has been engaged in research on digital manufacturing, CAD/CAM and PLM.

Zhenggan Zhou is a professor in School of Mechanical Engineering and Automation at Beihang University (BUAA). He is also the vice president of NDT (Nondestructive Testing) Branch of Chinese Society of Mechanical Engineering. His research interests include ultrasonic nondestructive testing and computer control technology.

Yi Lei is a professor in School of Mechanical Engineering and Automation at Beihang University (BUAA). He was awarded the second prize of the National Science and Technology Process of China. He is a member of National Technical Committee on Information Technology of Standardization in China. His research interests include cloud manufacturing, CAD/CAM and collaborative manufacturing.